



Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.



Ontology-based computer aid for the automation of HAZOP studies

Johannes I. Single^{*}, Jürgen Schmidt, Jens Denecke

CSE Center of Safety Excellence (CSE Institut), Joseph-von-Fraunhofer-Str. 9, Pfingsttal, 76327, Germany

ARTICLE INFO

Keywords:

Computer-aided HAZOP
Ontologies and reasoning
Ontology-based hazard identification
Reasoning strategy
Next-generation HAZOP

ABSTRACT

Hazard and Operability (HAZOP) studies are conducted to identify and assess potential hazards which originate from processes, equipment, and process plants. These studies are human-centered processes that are time and labor-intensive. Also, extensive expertise and experience in the field of process safety engineering are required. There have been several attempts by different research groups to (semi-)automate HAZOP studies in the past. Within this research, a knowledge-based framework for the automatic generation of HAZOP worksheets was developed. Compared to other approaches, the focus is on representing semantic relationships between HAZOP relevant concepts under consideration of the degree of abstraction. In the course of this, expert knowledge from the process and plant safety (PPS) domain is embedded within the ontological model. Based on that, a reasoning algorithm based on semantic reasoners is developed to identify hazards and operability issues in a HAZOP similar manner. An advantage of the proposed method is that by modeling causal relationships between HAZOP concepts, automatically generated but meaningless scenarios can be avoided. The results of the enhanced causation model are high quality extended HAZOP worksheets. The developed methodology is applied within a case study that involves a hexane storage tank. The quality and quantity of the automatically generated results agree with the original worksheets. Thus the ontology-based reasoning algorithm is well-suited to identify hazardous scenarios and operability issues. Node-based analyses involving multiple process units can also be carried out by a slight adjustment of the method. The presented method can help to support HAZOP study participants and non-experts in conducting HAZOP studies.

1. Introduction

The Hazard and Operability (HAZOP) method is a recognized and widely used scenario-based hazard evaluation procedure. Within HAZOP studies, processes, process plants, and equipment are systematically examined to identify and assess potential hazards and operability issues. Thus, the HAZOP method contributes to prevent accidents and increase process and plant safety. The method was developed in the 1960s and was first publicly presented in 1973 (Kletz, 1997).

HAZOP studies are carried out regularly within the life cycle of a process plant, for instance, during planning, commissioning, and revision. The methodology is human-centered and intended as a moderated brainstorming technique conducted by an interdisciplinary team of specialists. Depending on the scope and type of HAZOP, the team is composed of various specialists, such as study leader, operator, process engineer, process control engineer, and safety engineer.

Usually, HAZOP studies are conducted in the following steps: (0) collect the prerequisite process-safety information, (1) dividing the

process plant into nodes based on a piping and instrumentation diagram (P&ID) or process flow diagram (PFD), (2) examine the design intention of each node, (3) apply the process deviations to each node, (4) discussion of potential causes and consequences of the process deviation, (5) definition of safeguards and recommendations. Depending on the application of the method, the likelihood of occurrence and severity of the scenarios can be determined and the risk assessed.

The hazard identification is usually based on the assessments of experts. Thus, it depends on their professional experience, education, training, but also company policies, regulatory guidelines and soft facts, e.g., group dynamics, safety culture, and sentiment. These circumstances make it a time and labor-intensive process. Some authors criticize methodological aspects of the HAZOP method (Baybutt, 2015), for instance:

- that equal scenarios could have multiple deviations, which in turn can lead to repetitions that frustrate team members and reduces their alertness,

^{*} Corresponding author.

E-mail address: johannes.single@cse-institut.de (J.I. Single).

- that the focus is on a set of deviations instead of the initial events of scenarios,
- that operability scenarios outnumber hazard scenarios significantly and that these are often not considered because they would significantly increase the time required.

Leveson and Stephanopoulos (2014) point out that the accident causation models, which are the basis of the classic methods of hazard identification (e.g., HAZOP), are not sufficient to identify all hazards in complex systems. The System Theoretic Process Analysis (STPA) method was developed to overcome methodological limitations (Leveson and Thomas, 2018).

Another methodical aspect of the HAZOP method is the way it is conducted. Due to the current COVID-19 situation, HAZOP studies are partly held via web conferencing. This may result in reduced alertness of individual participants because the team is not in the same room and has shared insight into P&IDs and other process-safety information.

In order to reduce the time required, there are efforts to automate aspects of the HAZOP method. The automation of HAZOP studies and the provision of a computer-aid for HAZOP studies is a research topic for more than 30 years. Several researchers and research groups investigated the status quo regarding the automation of HAZOP studies and the general improvement of the method (Dunjó et al., 2010; Cameron et al., 2017; Single et al., 2019). The key results of this study are briefly described in the following.

Principally, computer systems for the computer-aid of HAZOP studies are concerned with the main tasks (1) creating and using a representation of the process plant and equipment, (2) storing relevant expert knowledge, (3) automatically infer conclusions about plausible scenarios, and identifying safeguards (Single et al., 2019). There are numerous technologies available to perform these tasks. Typically, the representation of the process plant is based on graph-theoretical concepts, such as signed-directed graphs (SDG). There are various knowledge representation formalisms, e.g., rule-based, frames, ontologies, to represent HAZOP relevant knowledge. Moreover, all previous approaches have an individual and technology-dependent strategy to identify scenarios that depends on the knowledge representation formalism.

Single et al. (2019) divided existing approaches regarding the automation of HAZOPs into three generations (cf. Fig. 1). The first approaches (Generation I) were based on so-called expert systems, which were mainly rule-based. In these approaches, general and specific knowledge is represented using IF-THEN rules, e.g., “IF flammable substance THEN avoid a source of ignition”. Generation II (cf. Fig. 1) approaches integrated models regarding the behavior of process variables, process plant, or the intention of the equipment with rule-based elements. Approaches of generation III (cf. Fig. 1) combined the usage of a model with case-based reasoning (CBR) (Kolodner, 1993) to

improve their reasoning capability.

Furthermore, a distinction between knowledge-based and data-driven methods can be made. Knowledge-based systems use semantic models to represent domain-specific relationships and system behavior. Based on semantic models, conclusions are drawn using reasoners or inference engines. Data-driven methods use raw or pre-processed data to train models to draw conclusions and predict system behavior. Historical data, process data, simulated data, or even records or documents can be used as an information basis. Within this research, the focus is on knowledge-based systems and approaches.

1.1. Representation of knowledge

Conclusions are drawn within HAZOP studies based on the experience and knowledge of the participants. Therefore, the modeling of expert knowledge and its representation is a crucial stage in automating HAZOP studies. Within the scope of this research, concepts from the knowledge domains substances, processes, process plants and units, equipment, hazards and malfunctions, causes, consequences, and safeguards are considered.

There are various knowledge representation formalisms, such as *rules*, *semantic nets*, *frames*, and *ontologies* (Li et al., 2018). Rules can represent logical implications and conditional actions (Davis et al., 1993). Semantic nets make use of nodes to represent objects and edges (connections) to represent the relationships between these objects. Lehmann (1992) described a semantic net as a “graph of the structure of meaning”. The frames formalism can be used to represent stereotypical situations (Minsky, 1974). Davis et al. state that frames are useful for the definition of terms, objects, and for describing taxonomic relationships and relationships between objects (Davis et al., 1993). Ontologies can also be used to represent objects and the semantic and hierarchical relationships between them. Depending on the used ontology language, restriction in the form of axioms can be added to specify the meaning of objects.

Ontologies have already been proposed within the scope of HAZOP studies by various research groups (Kuraoka and Batres, 2003; Daramola et al., 2011; Single et al., 2019; Single et al., 2020). Zhao et al. (2009) used ontologies in their HAZOP expert system. Other research groups used ontologies within decision support systems, for the support of FMEA studies, or the representation of knowledge extracted from chemical accident databases (Baumeister and Striffler, 2015; Ebrahimpour et al., 2010; Single et al., 2020).

Semantic nets, frames, and ontologies can be assigned to the group of knowledge graphs. Furthermore, frames are the more expressive successor of semantic nets. Frames and ontologies have similar capabilities, and differences can only be identified by looking at formal frame languages and ontology languages. Also, there are ontology languages that use rule-based elements (Horrocks et al., 2004). Therefore, ontology and frame languages are briefly discussed in the following.

1.2. Ontology languages in brief

The development of knowledge representation languages goes back to the 70s. One of the first logic-based frame languages was KL-ONE (Brachman and Schmolze, 1985). Another acknowledged ontology language is the *Resource Description Framework* (RDF) language that was developed in the late 90s. It is intended to describe machine-understandable web resources (Lassila and Swick, 1997). It can be used to explain concepts and their relations and forms the basis for other ontology languages. RDF models are formalized using the Extensible Markup Language (XML). RDF Schema (RDFS) is an extension of RDF (Hitzler et al., 2008). DAML + OIL is based on RDF and was additionally provided with formal aspects of description logic (DL) (Kalibatiene and Vasilecas, 2011). It is the predecessor of the OWL language. The *Web Ontology Language* (OWL) can unambiguously describe concepts, their relations and restrict concepts using axioms.

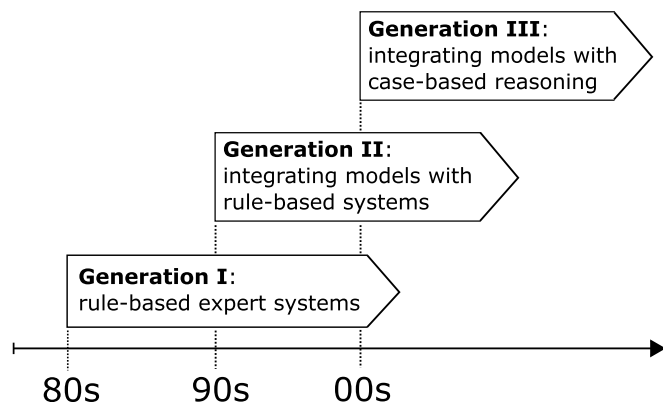


Fig. 1. Different generations of automation approaches (Single et al., 2019).

There are three sublanguages of OWL with different levels of expressiveness: OWL Lite, OWL DL, and OWL Full. To overcome the limitations of OWL regarding the ability to represent general rules, the *Semantic Web Rule Language* (SWRL) was developed. SWRL is an extension of OWL with horn-like rules in the *Rule Markup Language* (RuleML) (Horrocks et al., 2004). OWL 2 was developed to overcome drawbacks of OWL 1, such as limited expressivity regarding properties, and extended the set of built-in datatypes (Grau et al., 2008).

1.3. Objectives of this research

This paper aims to present an ontology-based method to generate HAZOP worksheets automatically. In the course of this, the importance of the ontological model and its semantic concepts is presented. Furthermore, a strategy shall be developed and described to infer logical conclusions from the proposed ontology and a process plant representation. In the process, extended concepts such as causes (primary and secondary), chain of consequences, and safeguards are identified. After the ontology-based method is presented, it is applied within a case study to a hexane storage tank. The case study serves to evaluate the automatically generated results. Compared to previous automation approaches, the automatically generated results shall be more comprehensible, while the modeling effort for representing the process units should be kept to a minimum. This shall be achieved by considering the following aspects:

- a complete causation model should provide transparency regarding the automatically identified scenarios,
- a reasoning algorithm should emulate the HAZOP process,
- meaningless HAZOP results shall be avoided within the automatic reasoning process,
- safeguards should be reliably identified,
- HAZOP worksheets should be created directly.

2. Methodology

A computer system that utilizes a knowledge base to draw conclusions and infer facts is called a knowledge-based system. The first step in developing a knowledge-based system is the conceptualization of relevant knowledge. The conceptualization process is shown in the upper part of Fig. 2 (conceptualization process). It includes the design of a structure and the modeling and formalization of knowledge. Furthermore, the process plant, equipment, and substances must be adequately represented. The results of the conceptualization and modeling process are ontology-based knowledge representation and an object-oriented process unit model library (cf. Fig. 2, upper part).

The conclusions drawn based on the ontologies depend on the inference strategy used to evaluate the ontology (cf. Fig. 2, lower part). The starting point is the selection of the relevant process units, processes, and the involved substances. This is done based on an object-oriented process unit model library. After selecting the required input data, an inference algorithm infers causes, consequences, safeguards, and related concepts based on the (process) deviations, process unit, and substance information (cf. Fig. 2).

2.1. The ontological model of HAZOP concepts

The design of a knowledge model of the relevant concepts in the form of an ontology is the first step within the conceptualization phase (cf. Fig. 2, upper part). This means that the concepts and the relationships between them must be identified and modeled carefully. In this context, modeling refers to set-theoretic and semantic modeling. Conclusions can be drawn from this semantic model regarding relevant scenarios.

According to Baybutt (2015), “There are other elements of scenarios that may be important and often are not recorded in HAZOP study worksheets”. The issue of a complete description of causal relationships within scenarios is considered within this research. Human experts can relate situations based on their professional experience to draw conclusions. Computer systems need well-defined models to infer facts and draw conclusions. For this reason, an extended HAZOP model is used in

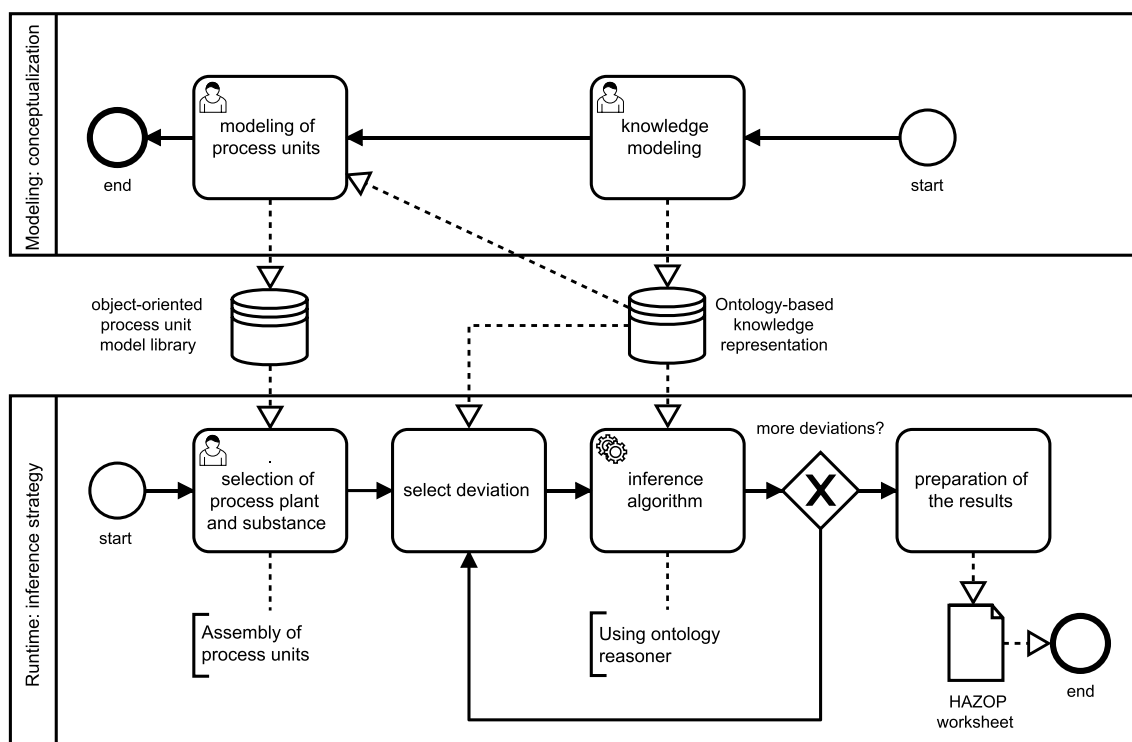


Fig. 2. Conceptualization process and application of the evaluation logic.

this work to illustrate causal relationships.

Therefore, the core concepts **deviations**, **causes**, **super causes**, **effects**, **consequences**, and **safeguards** are distinguished. Within this paper, object and data properties (relations between concepts) are denoted in an underlined font in lower camel case (e.g., hasIntendedFunction). Furthermore, specific ontology classes are denoted with double quotes in upper camel case notation (e.g., "LossOfPrimaryContainment").

An ontological model always represents a simplification of reality. It is assumed that causes and effects may be associated with deviations. Effects can lead to consequences, while consequences can have subsequent consequences. Causes can have higher-level causes in the form of super causes. Furthermore, safeguards are implemented to mitigate consequences and effects and prevent causes. These core concepts are shown in Fig. 3 and described in the following:

- **deviations** are composed of guidewords and (process) parameters and describe deviations from the design intent (part of a typical HAZOP worksheet),

- **causes** are a typical feature of HAZOP worksheets, and they describe the causes of the process deviations under consideration (part of a typical HAZOP worksheet),
- **super causes** are higher-level (primary) causes of causes; e.g., the cause *wrong rotating speed* of a pump could have the super cause *malfunction speed control*,
- **effects** arise from process deviations, and within the proposed model representation, for instance, this includes *rupture*, *overflowing* or *abnormal evaporation*,
- **consequences** potentially result from one or more effects; for instance, the effect *rupture* could have the consequence *loss of primary containment*. Also, consequences can have subsequent consequences which form a causal chain, e.g., the *formation of ex-atmosphere* leads to an *explosion* (part of a typical HAZOP worksheet),
- **safeguards** describe preventive, mitigative, primary and secondary safeguards, operational measures, and alarms (part of a typical HAZOP worksheet).

These fundamental relationships between the core concepts are not sufficient for the automatic identification of scenarios. Therefore,

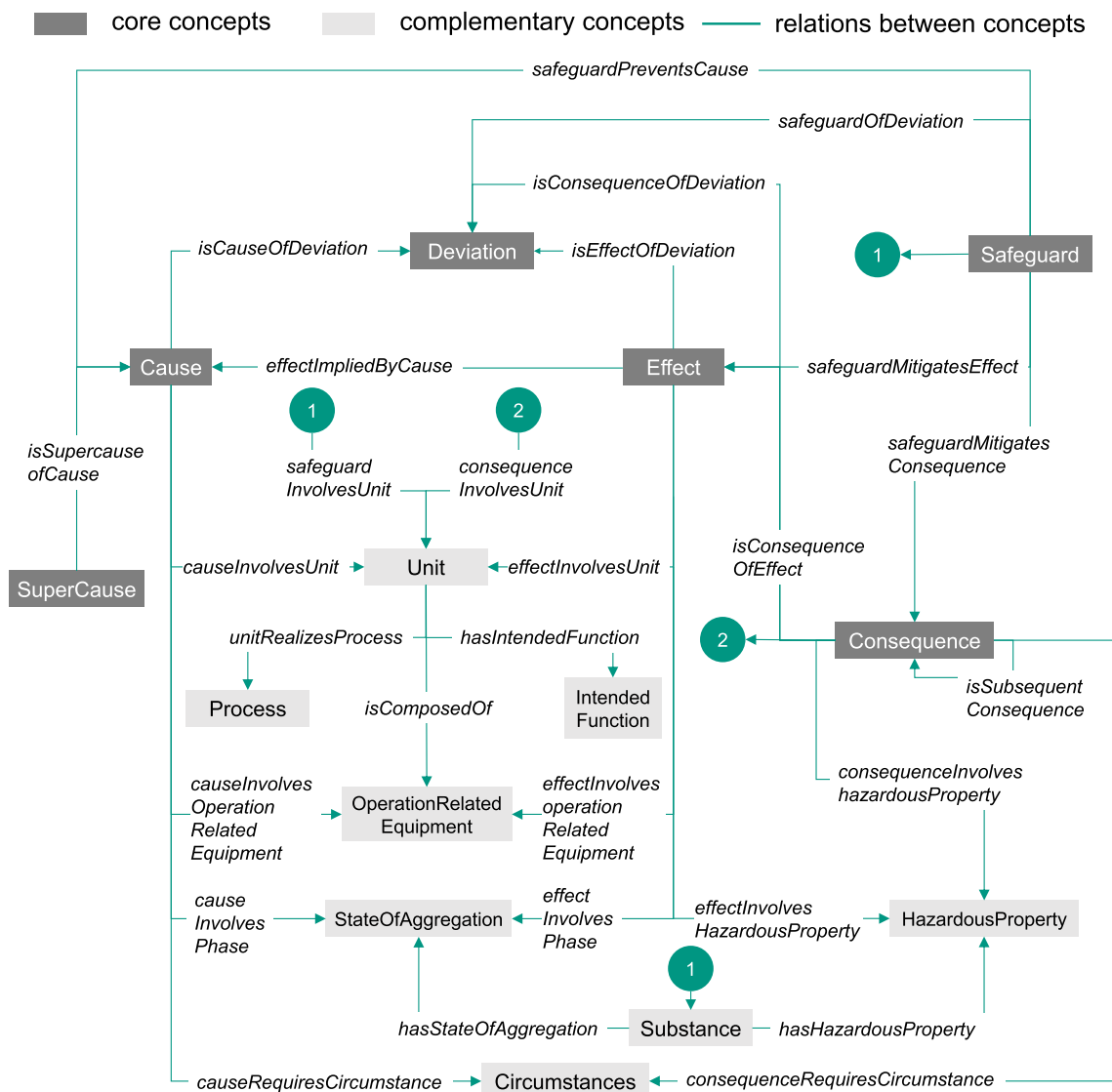


Fig. 3. Complete ontological model of relevant concepts.

complementary concepts and relationships are introduced to complete the ontological model, which include:

- **substance** involves properties of the substance, such as state of aggregation or hazardous attributes (e.g., flammability),
- **process unit** describes units (e.g., atmospheric storage tank) and operation related equipment (e.g., circulation pump, drain valve),
- **process** describes the interaction between substances and units,
- **circumstances** additional requirements that describe conditions such as ignition sources or other environmental conditions.

These considerations result in an ontological model that is shown in Fig. 3. The core concepts are directly connected to the complementary concepts, such as process, intended function, and substance. Without taking the process unit, substances, or other circumstances into account, no reliable conclusions can be drawn about hazardous scenarios or operability issues.

Potential causes and effects can be modeled using the concepts deviation, unit, substance, and additional circumstances. Plausible causes and effects can only be identified based on adequate representations of the involved process units. In the case of oversimplification, specific causes and effects cannot be identified. The “OperationRelatedEquipment” concept is used for this purpose (cf. Section 3.1).

Within the presented model super causes (or primary causes) are used to describe causes in more detail. For instance, the cause “ExternalLeakage” could have the super cause “DefectiveSeal”.

Furthermore, consequences follow effects and depend on the substance properties, process unit, and additional circumstances. For instance, the hazardous properties of substances significantly influence potential consequences. Also, the release of a flammable gas could lead to the formation of an explosive atmosphere, while the release of an inert gas could pose the danger of suffocation. Also, the additional circumstance „IgnitionSource“ is required to identify the consequence “Explosion”. Proposals for safeguards can be modeled based on causes, effects, and consequences, and complementary concepts.

“Safeguard” concepts are also connected to the “Unit” and “Substance” concept since they strongly depend on the process unit, involved equipment, and substance properties.

2.2. Logical foundation

The designed ontology was formalized using the Web Ontology Language (OWL) that was recommended by the World Wide Web Consortium (W3C) in 2004 (Hitzler et al., 2010). More specifically, the OWL 2 DL sublanguage was chosen because of its expressiveness and efficient reasoning. Within this research, the OWL 2 DL ontology is mainly based on classes (concepts), object and data properties (roles) to relate these classes. Additionally, axioms are used to restrict classes to specify their meaning.

Annotations are another component of the OWL ontology language. Annotation properties can be used to represent additional information, such as labels, descriptions, or further resources. They can be added to classes, instances, or properties. Within this work, they were used to provide explanations regarding the ontological model, explain concepts in detail, and provide sources.

The developed knowledge-models are based on formal logic. This

can be illustrated with the help of the description logic (DL) syntax. Classes within OWL are called concepts in DL, while object/data properties are called roles.

In Table 1, OWL constructors are compared to the DL and Manchester OWL syntax for illustrative purposes (Harmelen et al., 2007; Horridge et al., 2006). Only concepts necessary for clarification are presented.

The first example shows an intersection (\sqcap) between the classes A and B, which means the intersection contains all individuals that occur in class A and B (cf. Table 1).

In the second example, a class restriction is shown. Here, r stands for an object property while C represents a class. Furthermore, it is an existential restriction (\exists), which describes classes of individuals that have at least one specific property (e.g., `isEffectOfDeviation`) with individuals of a specific class (e.g., “LowTemperature”) (Horridge, 2011).

The third example shows the usage of data types. It consists of an existential restriction (\exists), the data property v , and the datatype T . Hence, class restrictions can be formulated as literals, e.g., flashpoint in Kelvin. OWL 2 DL has built-in data types, such as strings or floats (Hitzler et al., 2012).

In the fourth example, the equivalent class (\equiv) axiom is shown. This means the classes D and E contain exactly the same individuals (cf. Bechhofer et al., 2004), e.g., “Fracture” and “Break”.

First, an example super cause shall be described. The cause “ClosedOutletValve” could have the generic super cause “OperatorError”. The object property `isSuperCauseOfCause` is used to relate causes and super causes (cf. ontological model in Fig. 3). The corresponding formal definition of this simple knowledge model can be expressed using the DL syntax as demonstrated below:

$$\begin{aligned} \text{OperatorError} &\equiv \text{SuperCause} \sqcap \\ &\exists \text{isSupercauseOfCause.ClosedOutletValve.} \end{aligned}$$

Furthermore, effects can be derived directly from causes. For instance, the cause ‘contamination by water’ can imply the effect ‘drain line fracture’ (e.g., under the condition of a cold temperature). Using description logic (DL), the “DrainLineFracture” concept can be expressed as:

$$\begin{aligned} \text{DrainlineFracture} &\equiv \text{Effect} \sqcap \\ &\exists \text{effectImpliedByCause.ContaminationByWater} \sqcap \\ &\exists \text{isEffectOfDeviation.LowTemperature} \sqcap \\ &\exists \text{effectInvolvesPhase.Liquid} \sqcap \\ &\exists \text{effectInvolvesOperationRelatedEquipment.DrainValve.} \end{aligned}$$

Effects can lead to consequences that can be expressed as causal chains that consist of primary, secondary, and tertiary consequences, e.g., ‘loss of primary containment’ \rightarrow fire.

The hazardous attributes of substances and additional circumstances, e.g., ignition source, have a direct influence on the inferred consequences. For instance, using the object properties described in Fig. 3, the consequence “Fire” can be expressed as:

$$\begin{aligned} \text{Fire} &\equiv \text{Consequence} \sqcap \\ &\exists \text{isSubsequentConsequence.LossOfPrimaryContainment} \sqcap \\ &\exists \text{consequenceInvolvesHazardousAttribute.Flammable} \sqcap \\ &\exists \text{consequenceRequiresCircumstance.IgnitionSource.} \end{aligned}$$

Safeguards can either be derived based on potential effects and

Table 1
Relationship between OWL constructors, DL and Manchester OWL syntax.

	OWL constructor	DL syntax	Manchester OWL syntax	Example
1	intersectionOf	$A \sqcap B$	A and B	PHA \sqcap HAZOP
2	someValuesFrom	$\exists r.C$	r some C	$\exists \text{isEffectOfDeviation.LowTemperature}$
3	someValuesFrom	$\exists v.T$	v some T	$\exists \text{substanceHasFlashpointInKelvin.} \leq 328.15$
4	equivalentClass	$D \equiv E$	D EquivalentTo E	Fracture \equiv Break

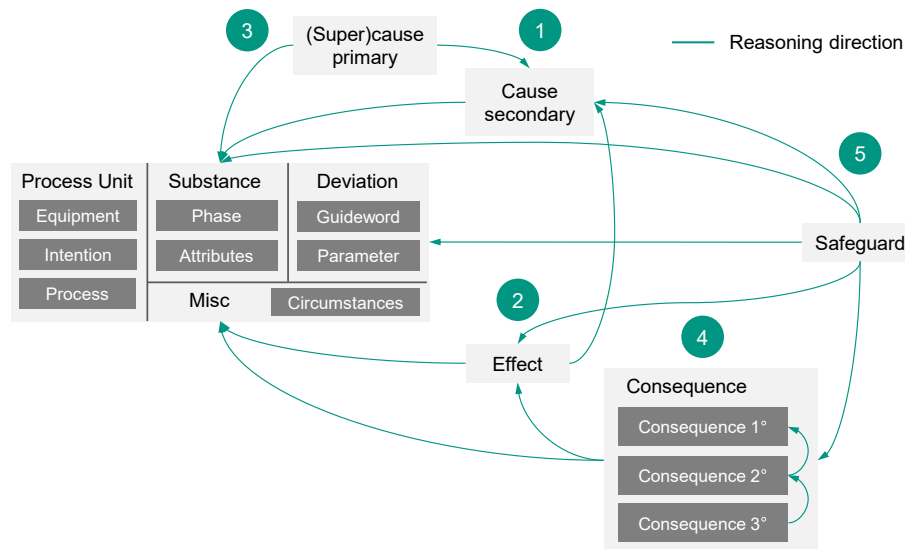


Fig. 4. Overall inference strategy.

consequences (mitigative safeguards) or directly on causes or deviations (preventive safeguards). For instance, a “PressureVacuumReliefValve” is a “Safeguard” that prevents the “CollapseOfEnclosure” of a “StorageTankUnit” which can be expressed as:

$PressureVacuumReliefValve \equiv Safeguard \sqcap$
 $\exists safeguardInvolvesUnit.StorageTankUnit \sqcap$
 $\exists safeguardMitigatesEffect.CollapseOfEnclosure.$

In addition to the use of object properties, data properties can also be used within the OWL 2 ontology language. These allow, among other things, to use numerical values within ontologies.

For example, some regulations (e.g., German Technical Regulations for Hazardous Substances, TRGS 509) define limits for the flashpoint of liquid substances above which minimum separation distances between storage tanks must be kept. This regulation can be represented in the form of a safeguard as follows:

$ConsiderDistanceBetweenTanks \equiv Safeguard \sqcap$
 $\exists safeguardInvolvesUnit.StorageTankUnit \sqcap$
 $\exists substanceHasFlashpointInKelvin. \leq 328.15.$

The Python module Owlready2 was used to create the ontology in an object-oriented manner programmatically (Lamy, 2017). When formalizing ontologies, it must be ensured that the intended meaning of the concepts matches the logical meaning that is embedded within the ontology.

2.3. Inference algorithm

The inferences that can be drawn from the ontologies depend on the evaluation strategy (compare Fig. 2, lower part). The scope of this paper is to demonstrate an equipment-based HAZOP analysis. Therefore, the node under consideration consists of a single process unit, including the directly involved equipment, e.g., circulation or transfer pump, cooling jacket, or power supply. In case a node consists of several process units, the inference algorithm must be executed for each unit, and additionally, interactions between the units must be considered. However, the evaluation of the ontologies always follows the same scheme.

The ontology is evaluated with different inputs and objectives within an inference algorithm to generate equipment-based HAZOP worksheets automatically. The call of a reasoner is an integral part of the inference algorithm, which is called multiple times. Reasoners are software packages that are used to infer logical consequences from ontologies. Thus, reasoners assume classification tasks, such as the computation of subsumption relations (e.g., concept A is a subset of concept B). According to Wang et al., a “classifier tries to build a model that satisfies all the axioms in the ontology” (Wang et al., 2006). Also, reasoners check the consistency of an ontology. Within this research, the Hermit (Glimm et al., 2014) reasoner was used, which is implemented in the Owlready2 Python module.

The developed inference algorithm is schematically shown in Fig. 4. The reasoning mechanism and the entire computer code is programmed using Python. The inference algorithm is a crucial part of the inference strategy that emulates the HAZOP process. The order of the inference steps determines which concepts are identified first (cf. numbers in Fig. 4). First potential causes, then effects, then super causes, then consequences, and then safeguards are identified. The reasoning direction is indicated with the lines and arrows.

Based on the deviations, substance information, and process units, causes and effects are inferred. Since effects can also be dependent on causes, first the causes and then the effects are identified (cf. Fig. 4). In this context, substance properties such as the state of aggregation and hazardous attributes are relevant. The equipment, the intended function of the node (equipment) under consideration, and the process are also essential. Furthermore, additional circumstances, such as ignition source or environmental conditions (e.g., confinement), are considered.

After identifying potential causes, super causes (primary causes) are inferred based on them and the process unit and associated equipment (cf. Fig. 4). Super causes are used to describe higher-level causes and thus the background of the causes in more detail.

Based on the identified effects, consequences are inferred. Individual consequences can result in subsequent consequences. Thus chains of consequences can occur. When identifying consequences, the properties of the substances involved are often of particular importance.

In the last step, preventive measures based on the deviations and

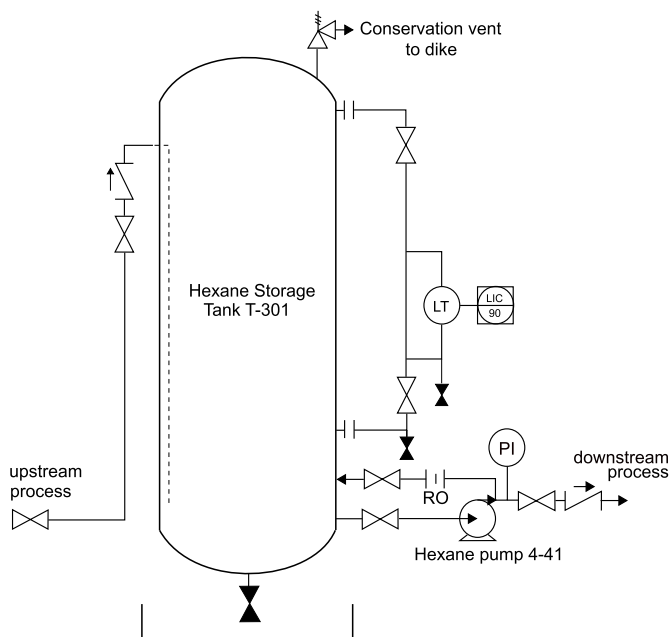


Fig. 5. P&ID of the hexane tank, adapted from (CCPS, 2001).

causes and mitigative safeguards based on the effects and consequences are inferred. Again, equipment and substance information is considered.

3. Results and discussion

The proposed method is applied within a case study to evaluate its capabilities. Therefore, the quality and the quantity of the generated results are compared to the original HAZOP worksheets. Afterward, the advantages of the method and its transferability to other process units and plants are discussed, and an outlook on future research work is provided.

3.1. Case study: hexane tank

The case-study involves a hexane storage tank (CCPS, 2001). A schematic excerpt P&ID of the system is shown in Fig. 5. It involves a storage tank (T-301) that stores liquid hexane, which is typically used as a solvent or reaction medium. Due to the vapor pressure of hexane, the storage tank is pressurized. The tanks' liquid level is controlled by a control loop, including a pressure indicator controller, and the tank is half full in standard operation. The makeup pump (4-41) is supplying a downstream process that is not further specified. Since the tank is considered in isolation, the upstream process is not specified further.

3.2. Representation of the process unit and substance

For the reasoning algorithm to identify hazardous scenarios and operability problems based on the proposed ontology, a representation of the storage tank is also required. Only essential parts of the hexane tank are represented, to keep the modeling effort for representing the process unit as simple as possible. The hexane tank is represented using the following concepts for modeling the process unit:

- Process unit: *storage tank (T-301)*,
- Operation related equipment: *drain line and valve; inlet and outlet valve; level indicator controller; recirculation/transfer pump (4-41)*,
- Intended function: *storage of liquid hexane.*

The conservation vent valve and the dike are intentionally not considered as part of the storage tank in the safety assessment. They should be recognized as a result of the automated HAZOP.

Furthermore, substance attributes, such as the intended state of aggregation (liquid) and the hazardous properties of hexane (e.g., flammable), are also considered.

Also, the assumptions made are taken into account in the form of circumstances:

- ignition sources: are always assumed as given, unless they can be excluded,
- presence of ambient air: is assumed because the tank is located outside,
- human interaction: it is assumed that an operator can interact with the process,
- introduction of impurities: is assumed to be possible, e.g., during filling.

Table 2
Illustrative model representation of the process unit, substance, and further circumstances.

	Universal concepts	Specific concepts	Property
Process unit	Unit	StorageTankUnit	
	IntendedFunction	Storage	hasIntendedFunction
	OperationRelatedEquipment	DrainValve	isComposedOf
		InletValve	
OutletValve			
Substance	LevelIndicatorController		
	RecirculationPump		
	StateOfAggregation	Liquid	hasStateOfAggregation
	HazardousProperty	Flammable	hasHazardousProperty
	Flashpoint	251.15 K	hasFlashpointInKelvin
Assumptions	Circumstances	IgnitionSource LocatedOutside IntroductionOfWater IntroductionOfImpurities SufficientOxygenAvailable InvolvesOperator	

Table 3
HAZOP table based on (CCPS, 2001).

Pos.	Deviation	Causes	Identified	Consequence	Identified	Safeguard/Recommendations	Identified			
0	High level	Flow from tank truck not discontinued before tank capacity has been reached	Yes	High pressure	Indirect	Level indication with high-level alarm (audible in the control room)	Yes			
		Inventory control error – truck arrives before needed	Yes				Other			
1	Low level	Inventory control error – truck arrives too late	Yes	No safety consequences – potential process interruption if not refilled before the downstream feed tank is empty	Yes	–	Other			
		Low flow or no flow – line from the tank truck to hexane storage tank T-301 through hexane unloading pump	Yes							
2	High temperature	No credible causes identified	Other	–	Other	–	Other			
3	Low temperature	Low ambient temperature while there is water contamination in the tank	Indirect	Possible freezing of accumulated water in the heel of the tank or the tank’s drain line or instrument lines, resulting in fracture of the drain line and loss of containment	Indirect	–	Other			
4	High pressure	High level	Other	Release of hexane through the relief valve into the tank’s dike	Yes	–	Other			
				Fire hazard affecting a large area if not contained by the dike	Yes					
				Loss of containment (if the overpressure cause exceeds the tank pressure rating)	Yes					
5	Low pressure	Tank blocked in before cooldown, following steam-out	Yes	Equipment damage resulting in a collapse of the tank under a vacuum	Yes	Standard procedures for steam-out of vessels	Other			
6	High concentration of contaminants (Other than composition)	Water not completely drained following a steam-out or washout	Yes	Possible freezing of accumulated water in the tank during a period of low ambient temperature	Yes	–	Other			
		High concentration of contaminants – line from the tank truck to hexane storage tank T-301 through hexane unloading pump	Yes							
7	Loss of containment (Elsewhere flow)	Corrosion	Partially	Release of hexane	Yes	Operation/maintenance response as required, including isolation if needed	No			
		Erosion								
		External fire					Capability to manually isolate the tank	No		
		External impact								
		Gasket, packing or seal failure					Periodic non-destructive inspection per API recommended practice and ASME code	Yes		
		Improper maintenance								
		Instrument or instrument line failure					Fire hazard affecting a large area particularly if the capacity of the dike is exceeded	Yes	Relief valve that discharges to the tank’s dike	Yes
		Material defect								
		Sample station valve leaking							Dike sized for 1.5 times the capacity of the tank	Yes
		Vent or drain valve leaking								
Low temperature			Emergency response procedures	No						
High pressure (if the overpressure cause exceeds the equipment pressure rating)										

Table 4
Automatically created extended HAZOP worksheet for the deviations ‘high level’ and ‘low level’.

Pos.	Unit	Id.	Substance	Deviation	Super cause	Cause	Effect	Consequence	Safeguard
1	StorageTankUnit	T-301	Hexane	HighLevel	–	ExcessiveInflow	Overfilling	LossOfPrimaryContainment, Fire	CollectingBasin, OverfillProtection
2	StorageTankUnit	T-301	Hexane	HighLevel	–	ExcessiveInflow	Overfilling	LossOfPrimaryContainment, FormationOfExAtmosphere, Explosion	CollectingBasin, DefineExProtectionZone, OverfillProtection
3	StorageTankUnit	T-301	Hexane	HighLevel	OperationalError, ValveFailure	ClosedOutletValve	Overfilling	LossOfPrimaryContainment, Fire	CollectingBasin, OverfillProtection
4	StorageTankUnit	T-301	Hexane	HighLevel	OperationalError, ValveFailure	ClosedOutletValve	Overfilling	LossOfPrimaryContainment, FormationOfExAtmosphere, Explosion	CollectingBasin, DefineExProtectionZone, OverfillProtection
5	StorageTankUnit	T-301	Hexane	HighLevel	–	MalfunctionFlow-Controller	Overfilling	LossOfPrimaryContainment, Fire	CollectingBasin, OverfillProtection
6	StorageTankUnit	T-301	Hexane	HighLevel	–	MalfunctionFlow-Controller	Overfilling	LossOfPrimaryContainment, FormationOfExAtmosphere, Explosion	CollectingBasin, DefineExProtectionZone, OverfillProtection
7	StorageTankUnit	T-301	Hexane	LowLevel	DepositionOfImpurities	BlockedInflowLine	PumpRunningDry	PumpBreakdown, ProductionDowntime	DryRunProtection, LevelControllerLowAlarm
8	StorageTankUnit	T-301	Hexane	LowLevel	HumanError, OperationalError, ValveFailure	ClosedInletValve	PumpRunningDry	PumpBreakdown, ProductionDowntime	DryRunProtection, LevelControllerLowAlarm
9	StorageTankUnit	T-301	Hexane	LowLevel	HumanError, OperationalError, ValveFailure	ClosedInletValve	EmptyingOf-Container	ProductionDowntime	LevelControllerLowAlarm
10	StorageTankUnit	T-301	Hexane	LowLevel	–	MalfunctionFlow-Controller	EmptyingOf-Container	ProductionDowntime	LevelControllerLowAlarm
11	StorageTankUnit	T-301	Hexane	LowLevel	OperationalError	LossOfInflow	PumpRunningDry	PumpBreakdown, ProductionDowntime	DryRunProtection, LevelControllerLowAlarm
12	StorageTankUnit	T-301	Hexane	LowLevel	OperationalError	LossOfInflow	EmptyingOfContainer	ProductionDowntime	LevelControllerLowAlarm

The overall model representation of the hexane tank, including substance-specific information and additional assumptions, are shown in Table 2. Within the developed object-oriented process unit model library (cf. Fig. 2, upper part), several process unit representations are available. These details regarding the substance, process unit, involved operation related equipment and additional circumstances are modeled as ontology concepts that serve as an input for the inference algorithm (cf. Fig. 4).

3.3. Original HAZOP results

Within the original HAZOP study, the following deviations were considered: 'high level', 'low level', 'high temperature', 'low temperature', 'high pressure', 'low pressure', 'high concentration of contaminants', and 'loss of containment'. For practical use, the deviations are directly composed of guidewords and parameters.

The results of the original HAZOP worksheets are shown in Table 3. The direct comparison of the automatically generated with the original results is made in the column *Identified*. In case the automatically created results match the original results, it is indicated with a "Yes" in the *Identified* column. In the case of a similar conclusion within a different scenario, it is indicated with an "Indirect". In case another conclusion is drawn, it is indicated with an "Other".

For instance, in Table 3, the safeguard for the 'low pressure' deviation is 'standard procedures for steam out of vessel', while the proposed method identified "CollectingBasin", "VacuumProofDesign", "Pressur-eVacuumReliefValve" (cf. Figure A-2, scenario 29).

Also, there can be additional scenarios that have been found. For instance, multiple 'high temperature' deviation scenarios have been identified, while no scenarios have been listed in the original worksheets (cf. Table 3).

Accordingly, Table 3 provides a first overview regarding the scenarios, which were recognized in the comparison to the original HAZOP.

3.4. Results of the proposed methodology

The application of the described methodology leads to automatically created HAZOP worksheets that are shown in Table 4 and in the appendix in Table A-1, Table A-2, Table A-3. Before results are returned, they are structured, formatted, and redundant results are removed, which is done automatically. These worksheets are generated by the computer code that was implemented within Python. Worksheets are returned as Hypertext Markup Language (HTML) files, which can be further processed.

Overall, 40 potential scenarios with eight different deviations were identified automatically. The selected deviations are the same as in the original HAZOP example (CCPS, 2001). Within the proposed approach, different names were used for the deviations. Instead of the deviation 'high concentration of contaminants', the deviation 'other than composition' was used. Also, instead of 'loss of containment', the deviation 'elsewhere flow' was used.

Based on the inference strategy, one cause or effect per scenario is identified, while several super causes and a chain of consequences are possible. In addition to the typical HAZOP worksheet columns, the considered substances, process unit, super causes, and effect are shown. Each row represents a scenario. In conventional HAZOP worksheets, no distinction is made between causes and super causes and effects and consequences. These are recorded in the same column. Within the automatically generated HAZOP worksheets, details such as super-causes, substances, and equipment are explicitly listed for each scenario to enable plausibility checks.

Within the generated HAZOP worksheets, there are no references to other scenarios. In each series, the scenario is described in full, including the process unit under consideration and the substance. Compared to conventional HAZOP worksheets, there are some duplications, especially regarding the consequences. On the other hand, this allows the comprehensibility of the automatically generated results. Hence, each scenario (each row) is different but may share identical columns.

Generally, causes can lead to different effects, which in turn can lead to other consequences. For instance, the following cause has different effects:

- Cause: "LossOfInflow" → effect: "PumpRunningDry" (cf. Table 4, Pos. 11),
- Cause: "LossOfInflow" → effect: "EmptyingOfContainer" (cf. Table 4, Pos. 12).

Thus, parallel effects are inferred and are represented within the HAZOP worksheet. Also, an effect can have multiple different causes that are listed separately, for instance:

- Effect: "Overfilling" → Cause: "ExcessiveInflow" (cf. Table 4, Pos. 1–2),
- Effect: "Overfilling" → Cause: "ClosedOutletValve" (cf. Table 4, Pos. 3–4),
- Effect: "Overfilling" → Cause: "MalfunctionFlowController" (cf. Table 4, Pos. 5–6).

Furthermore, the same effects can lead to different chains of consequences. Different chains of consequences form different scenarios with different safeguards and are listed in separate rows. For instance, the effect "Rupture" could lead to the consequence chains:

- "LossOfContainment, DirectIgnition, Fire" (cf. Table 4, Pos. 1) or,
- "LossOfContainment, FormationOfExAtmosphere, Explosion" (cf. Table 4, Pos. 2).

There are also scenarios where no super cause or effect or no other consequence has been identified. Since super causes are optional and describe the background of causes in more detail, it cannot be concluded that the results are incomplete due to missing super causes. If other parts of the HAZOP worksheet are missing, and for example, only a cause without an effect has been identified, this must be discussed by the HAZOP team.

In general, identified scenarios serve as a basis for discussion within HAZOP studies. The human expert team has the task of reviewing the scenarios for plausibility, likelihood of occurrence, and severity. It seems to make more sense that human experts discard results rather than a computer system not providing them.

3.5. Comparison and discussion of the HAZOP worksheets

Within a conventional HAZOP, potential causes and consequences are usually identified based on the deviations and information concerning the node. For instance, within the original HAZOP worksheet in Table 2 (Pos. 5) based on the deviation 'low pressure', the cause "tank blocked in before cooldown" and the consequence "equipment damage resulting in a collapse" are identified. Based on these results, appropriate safeguards can be selected. Within the developed extended worksheet, there are multiple causes of the deviation 'low pressure' (cf. Table A-2, scenario 29–34). For example, the cause "ObstructedVentPath" with the super causes "FaultyInstallation" and "HumanError" have been

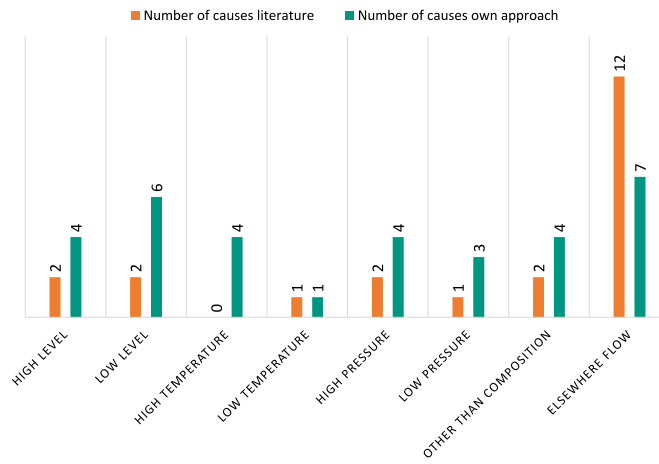


Fig. 6. Comparison of the identified causes.

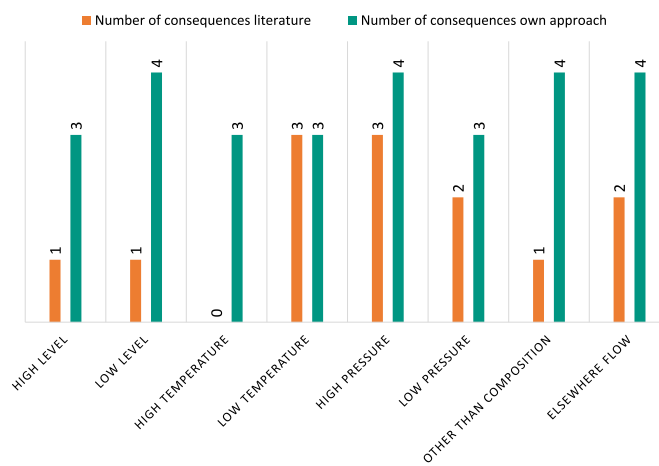


Fig. 7. Comparison of the identified potential consequences.

automatically identified. Furthermore, the effect “CollapseOfEnclosure” with two different consequence chains, “LossOfContainment” and “Fire” or “Explosion”, have been identified. Based on these findings, multiple safeguards were proposed, e.g., “CollectingBasin”, “VacuumProofDesign”, “DefineExProtectionZone”, “PressureVacuumReliefValve” (cf. Table A-2, scenario 29–30).

From a qualitative point of view, a large part of the causes and consequences were identified (cf. Table 2). The direct comparison of HAZOP results requires the interpretation of scenarios. Different chains of causes or consequences are shown separately in a new row (cf. Table 4, Table A-1, Table A-2, Table A-3). Different deviations may lead to similar scenarios. Some scenarios show similarities or were interpreted differently with similar conclusions. For instance, in the original HAZOP, the consequence of the ‘high level’ deviation is ‘high pressure’. Within the automatically generated worksheets, the consequence is “Rupture”. In the original worksheet, this consequence is again listed under the ‘high pressure’ deviation. References to other scenarios have been avoided, and all scenarios have been described in full to improve readability.

Furthermore, the consequence “freezing and fracture of the drain line” was recognized within this work with the deviation ‘other than composition’ (cf. Table A-1, scenario 15–16). This can be explained by the fact that without water contamination, it would not happen at all. In the original HAZOP, it was recognized with the deviation ‘low temperature’. This consequence results from the two deviations ‘low temperature’ and ‘other than composition’. (Duhon, 2017) speaks in this context of ‘guideword overlap’.

These examples show that different terms can be used while similar conclusions can be drawn. This is also a typical issue within conventional HAZOP studies because experts use different vocabulary, which also depends on company-specific guidelines. Within the automatically created worksheets, causes/super causes and effects/consequences are listed separately. In the original HAZOP worksheet, all causes and consequences are listed together. For instance, in the original worksheet, a cause of the deviation ‘loss of containment’ is corrosion. Within the proposed approach, “Corrosion” is a super cause while the corresponding cause is “ExternalLeakage”.

Safeguards depend heavily on the hazard potential, risk assessment, industry, and company-specific guidelines. Some of the listed safeguards can also correspond to general recommendations that are not tied to a specific scenario. In this case study, the generic safeguards were identified:

- “FlameArrester”,
- “PeriodicalExamination”,
- “ConsiderMinimumTankDistance”,
- “IgnitionSourceAssessment”.

Compared to the original HAZOP, more scenarios were recorded within the proposed method. On the other hand, the original HAZOP was intended to demonstrate different ideas and methods. It is questionable to what extent the completeness of the original HAZOP was the claim of the authors (CCPS, 2001).

Nevertheless, this HAZOP example is well suited to compare the quality of the results. The number of identified causes and consequences are shown in Figs. 6 and 7. To determine the number of concepts identified, chains of causes and consequences are counted. For instance, the scenario with the super cause “ExternalFire” and the cause “ThermalExpansion” would count as one in Fig. 6. The scenario with the effect “Rupture” and the consequence “LossOfPrimaryContainment, FormationOfExAtmosphere, Explosion” would count as two consequences in Fig. 7. This means that intermediate events in the consequence chain,

such as “FormationOfExAtmosphere” are not counted separately.

More causes (own: 33, original: 22) and consequences (own: 28, original: 13) have been identified using the proposed method, especially regarding the ‘high temperature’ deviation. In the original worksheet, more causes regarding the ‘elsewhere flow’ deviation have been identified. In both HAZOP approaches, many scenarios consider a loss of containment. This can lead to a fire or even an explosion due to the flammability of hexane. Within the original HAZOP worksheet, the scenario of an explosion was not considered. Therefore, a different number of consequences can be explained.

The focus of the original HAZOP was not on the identification of safeguards or recommendations. The selection of safeguards depends strongly on the identified causes and consequences. Since the causes and consequences differ, the number of identified safeguards is not directly compared within this paper.

A quantitative evaluation allows metrics to be calculated to measure the extent or focus on specific deviations. For example, the mean value of the identified causes (own approach) is 4.125 per deviation, while the mean value of the consequences is 3.5 per deviation. Concerning Figs. 6 and 7, it can be concluded that the number of causes per deviation varies more than the number of consequences. This information can be used, for example, to improve the ontological model. Concerning HAZOP automation systems, HAZOP metrics can track the quantitative effect on the HAZOP results of changes to the ontological model.

Nevertheless, the comparison of the number of identified scenarios does not allow any statement about the completeness or the quality of the results. This can only be done based on qualitative aspects through the analysis and interpretation of scenarios by human experts. Based on these expert assessments, concepts in the ontology can be extended to identify other scenarios in the future. Within this case study, the presented knowledge-based system was able to generate qualitatively equivalent results compared to the original HAZOP.

3.6. Methodical advantages

A separate consideration and listing of causes, super causes, effects, and consequences improve the transparency of causal relationships of the automatically generated results and helps to identify and resolve inconsistencies. The division of the concepts contributes to a more adaptive causation model. An advantage of the proposed method is that by considering causal relationships between HAZOP concepts within the ontological model, meaningless scenarios can be avoided.

The point raised by Leveson and Stephanopoulos (2014) that within classical accident causation models, it is assumed that cause and effect are directly related and that this is not always true has been considered within this method. As shown in Fig. 4 there is a connection between cause and effect, but it is optional. Accordingly, effects can be identified independently of causes. Furthermore, consequences are not exclusively dependent on effects but can be derived based on process deviations. Thus the causation model goes beyond previous approaches.

Since the automatically generated results are based on an ontological model, a plausibility check by human experts is still required. Nevertheless, the generated worksheets can be used as a basis for discussion in HAZOP studies or for the preparation of participants. Furthermore, results based on ontological models can contribute to the standardization of the HAZOP method and its results.

A further requirement is identifying reasonable safeguards, especially in comparison to other automation approaches from the past. Within the proposed approach, a distinction was made between general safeguards, e.g., “RegularInspections” and scenario-specific safeguards, e.g., “VacuumProofDesign”. Therefore, this requirement could be fulfilled. Safeguards are usually selected based on company-specific

regulations or other technical rules or standards. Thus, the automatically identified safeguards are proposals that still require expert evaluation.

A further goal of the method is to limit the modeling effort for representing the process and the process units as much as possible. The degree of abstraction was defined so that only relevant equipment specific components are represented, which are needed to identify typical industrial HAZOP scenarios. The total modeling effort for simple systems such as the storage tank is minimal (cf. Section 3.2). Furthermore, once modeled process units can be reused. In the course of this, mainly the hazardous material properties and the state of aggregation of the substances involved were considered. Furthermore, general circumstances can be defined for individual process units, such as “LocatedOutside”, “IgnitionSource”, or “InvolvesHumanInteraction”.

The mentioned issue that scenarios can have multiple process deviations, which in turn leads to repetition and frustration of team members, can be excluded by computer systems. On the other hand, the HAZOP team has a monitoring function that critically evaluates automatically generated HAZOP worksheets.

3.7. Transferability of the method

Ontology-based models are necessary to apply the presented method. These models describe the semantic relations between process deviation, cause, effect, consequence, and safeguard. Because many concepts can be transferred, these models are not necessarily bound to specific process units but are generic. Regarding the hexane case study, this means, for example, that both pump specific and storage tank specific details have been modeled, which can be reused.

Each scenario to be identified must first be represented within the ontological model. Scenarios must also be abstracted and simplified so that they can be represented within the ontological model. Currently, the developed ontology structure consists of 448 class definitions. Of these, 53 classes refer to different causes, 56 to super causes, 46 to effects, 30 to consequences, and 39 to safeguards. The other classes refer to process units, equipment, substances, and other boundary conditions. The ontology structure is continuously enhanced.

Specific class definitions can be extended to be applicable to further process units. Compared to the representations from Section 2.2, class definitions can be significantly more complicated. New definitions can only be implemented by taking the other classes and the ontological structure into account. Otherwise, existing concepts could be dissolved, or the wrong conclusions could be drawn and the wrong scenarios identified accordingly. Thus, it is unnecessary to create new ontologies for each process unit or node, but process units can be assembled using the modeled equipment. From different process units, nodes and entire process plants can be assembled.

The basis of the proposed method is a representation of the process units and equipment. In the context of this work, an object-oriented process unit model library (cf. Fig. 2) was developed, which is also based on ontology classes. The library is designed so that new process units can be easily created. A further aspect is considering the upstream and downstream propagation of process deviations and effects through plant components and entire nodes. This is the subject of current research.

3.8. Future work

Future research could aim to integrate the developed object-oriented process unit model library with concepts from the ISO 15926 (Batres et al., 2005). Also, the integration of other ontologies, e.g., OntoCAPE, could be pursued in the future (Morbach et al., 2009).

In the context of the extension of the ontological model, security

aspects could be included in the future, so safety and security aspects can be considered together.

Furthermore, a distinction could be made between safeguards and recommendations.

Another idea for future research approaches would be to use intelligent P&IDs (Kang et al., 2019). This would reduce the effort to model the process plant using the process unit model library. Thus, the process unit and equipment information could be directly extracted from intelligent P&IDs and processed. On the other hand, the modeling effort for simple systems such as the storage tank is minimal and cannot be compared with the effort required to read and process information from smart P&IDs.

4. Conclusions

In this research approach, a knowledge-based methodology is developed to generate HAZOP worksheets automatically. For the representation of knowledge from the HAZOP domain, ontologies are used. Within the knowledge modeling process, it is particularly important to define relationships between knowledge concepts unambiguously. The completeness of the results is directly dependent on the quality and completeness of the ontology concepts. In addition to knowledge modeling, a method was developed to represent process units. In this context, the selected degree of abstraction plays a decisive role. If process units are modeled very roughly, wrong conclusions may be drawn. In case of a too detailed process unit representation, the process unit modeling process becomes too complicated and time consuming. After the design, conceptualization, and formalization of a suitable ontology, an inference strategy was designed and implemented to draw conclusions about the ontology and the process unit representation. Based on that, HAZOP worksheets have been generated automatically. Conventional HAZOP studies form socio-technical systems where the results depend on the skills of the HAZOP team. In computer systems, the results depend directly on the representation of knowledge, representation of the process plant, and the inference algorithms. While computer systems can reduce human errors within HAZOP studies, they are shifted to the modeling process of ontologies.

The described methodology is applied within a case study to a hexane storage tank as an equipment-based HAZOP analysis. Afterward, the automatically generated results are evaluated and compared to the original HAZOP results. The presented results show that the developed ontology and the ontology-based reasoning algorithm are well-suited to generate equipment-specific HAZOP worksheets. More research is needed regarding the application of the method within node-based HAZOP studies, regarding the extraction of information on process units from P&IDs, and the propagation of hazards throughout the process plant. At this moment, there is no automatic risk assessment, and human experts must carry it out. Furthermore, the identified safeguards are proposals that always require expert judgment. Finally, this research approach contributes to demonstrating the capabilities of knowledge-based systems for HAZOP studies.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Table A-1
Automatically created extended HAZOP worksheet for the deviations ‘other than composition’ and ‘elsewhere flow’

Pos.	Unit	Id.	Substance	Deviation	Super cause	Cause	Effect	Consequence	Safeguard
13	StorageTankUnit	T-301	Hexane	OtherThan-Composition	–	Contaminants	PoorProductQuality	NoSafetyConsequence	PeriodicalSampleTaking
14	StorageTankUnit	T-301	Hexane	OtherThan-Composition	HumanError	ConfusionOf-Substances	–	–	PeriodicalSampleTaking
15	StorageTankUnit	T-301	Hexane	OtherThan-Composition	CondensationAir-Humidity, IntroductionOfWater	Contamination ByWater	DrainlineFracture	LossOfPrimaryContainment, Fire	WaterDetectionSystem, CollectingBasin, PeriodicalSampleTaking
16	StorageTankUnit	T-301	Hexane	OtherThan-Composition	CondensationAir-Humidity, IntroductionOfWater	Contamination ByWater	DrainlineFracture	LossOfPrimaryContainment, FormationOfExAtmosphere, Explosion	WaterDetectionSystem, CollectingBasin, DefineExProtectionZone
17	StorageTankUnit	T-301	Hexane	Elsewhere-Flow	ImproperInstallation, OperationalError	MechanicalSeal-FailureOfPump	PoolFormation	PumpBreakdown, ProductionDowntime	CollectingBasin
18	StorageTankUnit	T-301	Hexane	Elsewhere-Flow	ImproperInstallation, OperationalError	MechanicalSeal-FailureOfPump	PoolFormation	LossOfPrimaryContainment, Fire	CollectingBasin
19	StorageTankUnit	T-301	Hexane	Elsewhere-Flow	ImproperInstallation, OperationalError	MechanicalSeal-FailureOfPump	PoolFormation	LossOfPrimaryContainment, FormationOfExAtmosphere, Explosion	CollectingBasin, DefineExProtectionZone, ExplosionProofMotor
20	StorageTankUnit	T-301	Hexane	Elsewhere-Flow	Corrosion, Erosion, LeakyConnections, MaterialDefect, MechanicalDamage	ExternalLeakage	PoolFormation	PumpBreakdown, ProductionDowntime	CollectingBasin
21	StorageTankUnit	T-301	Hexane	Elsewhere-Flow	Corrosion, Erosion, LeakyConnections, MaterialDefect, MechanicalDamage	ExternalLeakage	PoolFormation	LossOfPrimaryContainment, Fire	CollectingBasin
22	StorageTankUnit	T-301	Hexane	Elsewhere-Flow	Corrosion, Erosion, LeakyConnections, MaterialDefect, MechanicalDamage	ExternalLeakage	PoolFormation	LossOfPrimaryContainment, FormationOfExAtmosphere, Explosion	CollectingBasin, DefineExProtectionZone, ExplosionProofMotor

Table A-2
Automatically created extended HAZOP worksheet for the deviations ‘high pressure’ and ‘low pressure’

Pos.	Unit	Id.	Substance	Deviation	Super cause	Cause	Effect	Consequence	Safeguard
23	StorageTank-Unit	T-301	Hexane	High-Pressure	DepositionOf-Impurities	BlockedOutflow-Line	Crack	LossOfPrimaryContainment, Fire	CollectingBasin, PressureReliefValve
24	StorageTank-Unit	T-301	Hexane	High-Pressure	DepositionOf-Impurities	BlockedOutflow-Line	Crack	LossOfPrimaryContainment, FormationOfExAtmosphere, Explosion	CollectingBasin, DefineExProtectionZone, PressureReliefValve
25	StorageTank-Unit	T-301	Hexane	High-Pressure	DepositionOf-Impurities	BlockedOutflow-Line	Rupture	LossOfPrimaryContainment, Fire	CollectingBasin, PressureReliefValve, PressureVacuumReliefValve
26	StorageTank-Unit	T-301	Hexane	High-Pressure	DepositionOf-Impurities	BlockedOutflow-Line	Rupture	LossOfPrimaryContainment, FormationOfExAtmosphere, Explosion	CollectingBasin, PressureReliefValve, DefineExProtectionZone, PressureVacuumReliefValve
27	StorageTank-Unit	T-301	Hexane	High-Pressure	AbnormalHotIntake, ExternalFire, SolarRadiation	ThermalExpansion	Rupture	LossOfPrimaryContainment, Fire	CollectingBasin, PressureReliefValve, PressureVacuumReliefValve
28	StorageTank-Unit	T-301	Hexane	High-Pressure	AbnormalHotIntake, ExternalFire, SolarRadiation	ThermalExpansion	Rupture	LossOfPrimaryContainment, FormationOfExAtmosphere, Explosion	CollectingBasin, PressureReliefValve, DefineExProtectionZone, PressureVacuumReliefValve
29	StorageTank-Unit	T-301	Hexane	Low-Pressure	–	Ambient-Temperature-Change	CollapseOf-Enclosure	LossOfPrimaryContainment, Fire	CollectingBasin, PressureVacuumReliefValve, VacuumProofDesign
30	StorageTank-Unit	T-301	Hexane	Low-Pressure	–	Ambient-Temperature-Change	CollapseOf-Enclosure	LossOfPrimaryContainment, FormationOfExAtmosphere, Explosion	CollectingBasin, PressureVacuumReliefValve, VacuumProofDesign, DefineExProtectionZone
31	StorageTank-Unit	T-301	Hexane	Low-Pressure	FaultyInstallation, HumanError	Obstructed-VentPath	CollapseOf-Enclosure	LossOfPrimaryContainment, Fire	CollectingBasin, PressureVacuumReliefValve, VacuumProofDesign
32	StorageTank-Unit	T-301	Hexane	Low-Pressure	FaultyInstallation, HumanError	Obstructed-VentPath	CollapseOf-Enclosure	LossOfPrimaryContainment, FormationOfExAtmosphere, Explosion	CollectingBasin, DefineExProtectionZone, PressureVacuumReliefValve, VacuumProofDesign
33	StorageTank-Unit	T-301	Hexane	Low-Pressure	Malfunction-TransferPump	ExcessiveFluid-Withdrawal	CollapseOf-Enclosure	LossOfPrimaryContainment, Fire	CollectingBasin, PressureVacuumReliefValve, VacuumProofDesign
34	StorageTank-Unit	T-301	Hexane	Low-Pressure	Malfunction-TransferPump	ExcessiveFluid-Withdrawal	CollapseOf-Enclosure	LossOfPrimaryContainment, FormationOfExAtmosphere, Explosion	CollectingBasin, VacuumProofDesign, DefineExProtectionZone, PressureVacuumReliefValve

Table A-3

Automatically created extended HAZOP worksheet for the deviations 'high temperature' and 'low temperature'

Pos.	Unit	Id.	Substance	Deviation	Super cause	Cause	Effect	Consequence	Safeguard
35	StorageTank-Unit	T-301	Hexane	High-Temperature	Mechanical Damage	HeatInputBy-Recirculation-Pump	Abnormal Evaporation	ExceedingAllowable-Pressure	PressureReliefValve
36	StorageTank-Unit	T-301	Hexane	High-Temperature	Mechanical Damage	HeatInputBy-Recirculation-Pump	Exceeding FlashPoint	–	IgnitionSource-Assessment
37	StorageTank-Unit	T-301	Hexane	High-Temperature	AbnormalHotIntake, ExternalFire, SolarRadiation	AbnormalHeat-Input	Abnormal Evaporation	ExceedingAllowable-Pressure	PressureReliefValve
38	StorageTank-Unit	T-301	Hexane	High-Temperature	AbnormalHotIntake, ExternalFire, SolarRadiation	AbnormalHeat-Input	Exceeding FlashPoint	–	IgnitionSource-Assessment
39	StorageTank-Unit	T-301	Hexane	Low-Temperature	–	Ambient-Temperature-Change	BrittleFracture	LossOfPrimaryContainment, Fire	CollectingBasin
40	StorageTank-Unit	T-301	Hexane	Low-Temperature	–	Ambient-Temperature-Change	BrittleFracture	LossOfPrimaryContainment, FormationOfExAtmosphere, Explosion	CollectingBasin, DefineExProtectionZone

References

- Batres, R., West, M., Leal, D., Price, D., Masaki, K., Shimada, Y., Fuchino, T., Naka, Y., 2005. An upper ontology based on ISO 15926. *Comput. Aid. Chem. Eng.* 20, 1543–1548. [https://doi.org/10.1016/S1570-7946\(05\)80099-9](https://doi.org/10.1016/S1570-7946(05)80099-9).
- Baumeister, J., Striffler, A., 2015. Knowledge-driven systems for episodic decision support. *Knowl. Base Syst.* 88, 45–56. <https://doi.org/10.1016/j.knsys.2015.08.008>.
- Baybutt, P., 2015. A critique of the Hazard and Operability (HAZOP) study. *J. Loss Prev. Process. Ind.* 33, 52–58. <https://doi.org/10.1016/j.jlp.2014.11.010>.
- Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A., 2004. OWL Web Ontology Language Reference [WWW Document]. <https://www.w3.org/TR/owl-ref/#equivalentClass-def>. (Accessed 19 September 2020).
- Brachman, R.J., Schmolze, J.G., 1985. An overview of the KL-ONE knowledge representation system. *Cognit. Sci.* 9, 171–216. [https://doi.org/10.1016/S0364-0213\(85\)80014-8](https://doi.org/10.1016/S0364-0213(85)80014-8).
- Cameron, I., Mannan, S., Németh, E., Park, S., Pasman, H., Rogers, W., Seligmann, B., 2017. Process hazard analysis, hazard identification and scenario definition: are the conventional tools sufficient, or should and can we do much better? *Process Saf. Environ. Protect.* 110, 53–70. <https://doi.org/10.1016/j.psep.2017.01.025>.
- CCPS, 2001. *Layer of Protection Analysis: Simplified Process Risk Assessment, first ed.* Wiley-AIChE.
- Daramola, O., Stålhane, T., Sindre, G., Omoronyia, I., 2011. Enabling hazard identification from requirements and reuse-oriented HAZOP analysis. In: 2011 4th Int. Work. Manag. Requir. Knowledge, MaRK'11 - Part 19th IEEE Int. Requir. Eng. Conf. RE'11 3–11. <https://doi.org/10.1109/MARK.2011.6046555>.
- Davis, R., Shrobe, H., Szolovits, P., 1993. What is a knowledge Representation? *AI Mag.* 14, 17–33. <https://doi.org/10.1609/aimag.v14i1.1029>.
- Duhon, H., 2017. HAZOPS Should Be Fun - the Stream-Based HAZOP, first ed. *Gibson Applied Engineering and Technology, Inc (GATE)*, Houston, TX.
- Dunjó, J., Fthenakis, V., Vilchez, J.A., Arnaldos, J., 2010. Hazard and operability (HAZOP) analysis. A literature review. *J. Hazard Mater.* 173, 19–32. <https://doi.org/10.1016/j.jhazmat.2009.08.076>.
- Ebrahimipour, V., Rezaie, K., Shokravi, S., 2010. An ontology approach to support FMEA studies. *Expert Syst. Appl.* 37, 671–677. <https://doi.org/10.1016/j.eswa.2009.06.033>.
- Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z., 2014. Hermit: an OWL 2 reasoner. *J. Autom. Reas.* 53, 245–269. <https://doi.org/10.1007/s10817-014-9305-1>.
- Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U., 2008. OWL 2: the next step for OWL. *J. Web Semant.* 6, 309–322. <https://doi.org/10.1016/j.websem.2008.05.001>.
- Harmelen, F., Lifschitz, V., Porter, B., 2007. *The Handbook of Knowledge Representation*, vol. 1034. Elsevier Sci., San Diego, USA.
- Hitzler, P., Kroetzsch, M., Parsia, B., Patel-Schneider, P., F., Rudolph, S., 2012. OWL 2 Web Ontology Language Primer [WWW Document], second ed.. <https://www.w3.org/TR/owl2-primer/#Datatypes>. (Accessed 20 September 2020).
- Hitzler, P., Krötzsch, M., Rudolph, S., 2010. *Foundations of Semantic Web Technologies*. CRC Press.
- Hitzler, P., Krötzsch, M., Rudolph, S., Sure, Y., 2008. *Semantic Web*, first ed. Springer.
- Horridge, M., 2011. *A Practical Guide to Building OWL Ontologies Using Protégé 4 and CO-ODE Tools*. The University Of Manchester, Edition 1.3. http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf. (Accessed 25 October 2020).
- Horridge, M., Drummond, N., Goodwin, J., Rector, A., Stevens, R., Wang, H.H., 2006. The manchester OWL syntax. In: Grau, B.C., Hitzler, P., Shankey, C., Wallace, Evan (Eds.), *CEUR Workshop Proceedings*. Athens, Georgia (USA).
- Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M., 2004. SWRL: a semantic web Rule Language combining OWL and RuleML [WWW Document]. <https://www.w3.org/Submission/SWRL/>. (Accessed 4 November 2020).
- Kalibatiene, D., Vasilecas, O., 2011. Survey on ontology languages. <https://doi.org/10.1007/978-3-642-24511-4>.
- Kang, S.O., Lee, E.B., Baek, H.K., 2019. A digitization and conversion tool for imaged drawings to intelligent piping and instrumentation diagrams (P&ID). *Energies* 12. <https://doi.org/10.3390/en12132593>.
- Kletz, T.A., 1997. HAZOP - past and future. *Reliab. Eng. Syst. Saf.* 55, 263–266. [https://doi.org/10.1016/S0951-8320\(96\)00100-7](https://doi.org/10.1016/S0951-8320(96)00100-7).
- Kolodner, J., 1993. *Case-Based Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Kuraoka, K., Batres, R., 2003. An Ontological Approach to Represent HAZOP Information. *Process Systems Engineering Laboratory*, Tokyo Institute of Technology. <http://ise.me.tut.ac.jp/members/rbp/pubs/ontological-approach-to-represent-hazop.pdf>, 2003. (Accessed 25 October 2020).
- Lamy, J.B., 2017. Owlready: ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. *Artif. Intell. Med.* 80, 11–28. <https://doi.org/10.1016/j.artmed.2017.07.002>.
- Lassila, O., Swick, R.R., 1997. Resource description framework (RDF) model and syntax [WWW Document]. <https://www.w3.org/TR/WD-rdf-syntax-971002/>. (Accessed 25 October 2020).
- Lehmann, F., 1992. Semantic networks. *Comput. Math. Appl.* 23, 1–50. [https://doi.org/10.1016/0898-1221\(92\)90135-5](https://doi.org/10.1016/0898-1221(92)90135-5).
- Leveson, N.G., Stephanopoulos, G., 2014. A system-theoretic, control-inspired view and approach to process safety. *AIChE J.* 60, 2–14. <https://doi.org/10.1002/aic.14278>.
- Leveson, N.G., Thomas, J.P., 2018. STPA Handbook. Self publishing. [https://psas.scripts.mit.edu/home/get_file.php?name=STPA handbook.pdf](https://psas.scripts.mit.edu/home/get_file.php?name=STPA%20handbook.pdf), 2018. (Accessed 25 October 2020).
- Li, X., Zhang, S., Huang, R., Huang, B., Xu, C., Zhang, Y., 2018. A survey of knowledge representation methods and applications in machining process planning. *Int. J. Adv. Manuf. Technol.* 98, 3041–3059. <https://doi.org/10.1007/s00170-018-2433-8>.
- Minsky, M., 1974. A Framework for Representing Knowledge. *Massachusetts Institute of Technology, USA*. <https://dspace.mit.edu/bitstream/handle/1721.1/6089/AIM-306.pdf?sequence=2>, 1974. (Accessed 25 October 2020).
- Morbach, J., Wiesner, A., Marquardt, W., 2009. OntoCAPE-A (re)useable ontology for computer-aided process engineering. *Comput. Chem. Eng.* 33, 1546–1556. <https://doi.org/10.1016/j.compchemeng.2009.01.019>.
- Single, J.I., Schmidt, J., Denecke, J., 2020. Knowledge acquisition from chemical accident databases using an ontology-based method and natural language processing. *Saf. Sci.* 129, 104747. <https://doi.org/10.1016/j.ssci.2020.104747>.
- Single, J.I., Schmidt, J., Denecke, J., 2019. Computer-aided hazop studies: Knowledge representation and algorithmic hazard identification. *WIT Trans. Built Environ.* 189, 55–66. <https://doi.org/10.2495/SAFE190061>.
- Single, J.I., Schmidt, J., Denecke, J., 2019. State of research on the automation of HAZOP studies. *J. Loss Prev. Process. Ind.* 62, 103952. <https://doi.org/10.1016/j.jlp.2019.103952>.

Single, J.I., Schmidt, J., Denecke, J., 2020. Ontology-Based Support for Hazard and Operability Studies. *Int. J. Saf. Sec. Eng.* 10 (No. 3), 311–319. <https://doi.org/10.18280/ijssse.100302>.

Wang, H.H., Noy, N., Rector, A., Musen, M., Redmond, T., Rubin, D., Tu, S., Tudorache, T., Drummond, N., Horridge, M., 2006. Frames and OWL side by side. In: 9th International Protégé Conference; Presentation Abstracts. -.

Zhao, J., Cui, L., Zhao, L., Qiu, T., Chen, B., 2009. Learning HAZOP expert system by case-based reasoning and ontology. *Comput. Chem. Eng.* 33, 371–378. <https://doi.org/10.1016/j.compchemeng.2008.10.006>.