COMPUTER-AIDED HAZOP STUDIES: KNOWLEDGE REPRESENTATION AND ALGORITHMIC HAZARD IDENTIFICATION

JOHANNES SINGLE, JÜRGEN SCHMIDT & JENS DENECKE
CSE Center of Safety Excellence (CSE-Institut), Germany

ABSTRACT

Safety assessments are conducted to identify and assess the risks that arise from processes, process plants or technical systems in general. This includes the identification of potential hazards posed by plants. One recognized and generally accepted method for this is the hazard and operability (HAZOP) method. It is a human-centered process that is time- and labor-intensive. In the presented research approach, the structure of a computer-aided HAZOP system is described. The identification of hazards and malfunctions within technical systems is knowledge-intensive. Within this research approach, it transpired that the semantically correct and detailed modeling of deviation cause and effect relationships in the form of ontologies are of particular importance to draw correct conclusions. Thus, the guiding principles of a knowledge representation framework are described from a process safety perspective, and serve as a basis for the automatic identification of hazards. An integral understanding of the process, process plant and involved substances requires extensive knowledge. The way in which this knowledge is used and the search for hazards is conducted has an influence on the completeness of the results. Within this approach, the hazard/malfunction identification is conducted on different layers of abstraction to improve the efficiency of the search algorithm. The proposed methodology is applied within a case study to a technical system that consists of a compressor, vessel and valve. The first results demonstrate that the proposed method is well-suited to understand and identify the context of hazards and malfunctions. Thus, a system for computer-aided HAZOP studies can be used to assist HAZOP conductors in performing hazard analysis while increasing the speed of safety assessments and serving as a decision support system.

Keywords: computer-aided HAZOP study, hazard identification, ontology-based reasoning, decision support system, automatic hazard analysis.

1 INTRODUCTION

Process hazard analysis (PHA) methods are used to identify and evaluate hazardous events that can occur in process plants. Within the chemical process industry, the HAZOP study is an established technique to review the process plant and identify potential hazardous events and operability malfunctions (see Kletz [1]). In HAZOP studies, the process plant is systematically and critically analyzed. It is thereby divided into sections called nodes, which are easier to analyze compared to the entire process plant. This is usually done based on the plant's Piping and Instrumentation Diagrams (P&ID). Afterwards, process deviations are applied to the nodes, while possible causes, consequences and safeguards are identified and discussed. Usually, HAZOP studies are conducted by a team of experts from different fields, such as plant operation, process control and safety engineering. The HAZOP study is a human-centered, guided brainstorming technique. Thus, it is a time- and labor-intensive process that is not always carried out for the entire process plant, but for particularly critical sections. The results of the study depend on the knowledge, personal experience, moderation, communication, discussion culture and level of training of the HAZOP study practitioners. Furthermore, the capabilities of the practitioners can be limited by repetitive tasks, large amounts of data or stressful conditions. Hence, human experts can be supported by using computer-aided HAZOP (CAH) systems, which represent computer programs that serve as

WIT Transactions on The Built Environment, Vol 189, © 2019 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/SAFE190061

Decision Support Systems. Such systems can be used to remove repetitive tasks and analyze large sets of data in HAZOP studies. Also, plant operators can be supported in the early stages of the HAZOP process, in which a safety professional may not necessarily be present. The usability of CAH systems depends on the quality of their hazard identification abilities and the completeness and reliability of the results.

1.1 Brief state of research

The automation of elements of HAZOP studies has been a topic of research for more than 30 years. The first systems were mainly built on logic trees, rules and rule networks, within expert system shells (see [2]–[4]). Later research approaches used qualitative models for reasoning purposes in combination with rules (see [5]–[8]). Eizenberg et al. [9] used quantitative model-based reasoning approaches based on process simulations. Some approaches aimed to integrate computer-aided design (CAD) software with automatic hazard identification. More recent developments used sophisticated graph-based methods (see [10]–[12]). Zhao et al. [13] used case-based reasoning to draw conclusions based on previous situations.

1.2 Technical basis

A computer program for computer-aided HAZOP studies is essentially concerned with three tasks: (1) representation of the process plant, (2) knowledge representation of generic and process-specific knowledge, and (3) hazard or malfunction identification. The available technologies with which to achieve these tasks are briefly described below.

The technical system or process plant must be digitally represented in order to be analyzed. Thus, information regarding the arrangement of plant units and interconnections is needed. This can be generated using a graphical editor, although this can be time-consuming for complex systems. Another possibility is the direct extraction of plant information from a P&ID or Process Flow Diagram (PFD). This information can also be extracted using process simulation programs. This automatic form of extraction has the advantage of reducing the time that is required to model the process plant. In order to reuse process plant information, a shared data model is crucial to reduce the data translation effort. For instance, ISO 18629-1:2004 or DIN EN 62424:2017-12 describe exemplary data models (see [14], [15]).

The term “knowledge” represents the understanding of information, facts or skills. Knowledge must be made available to human experts and also to machines. Hence, a knowledge representation formalism is needed, which has different requirements:

- Various types of knowledge, such as facts and procedures, must be recorded;
- The semantic context of knowledge fragments must be clearly illustrated;
- Knowledge structures must be easily extensible; and
- Knowledge queries and inference must be efficient and complete.

There are various formalisms for knowledge representation, such as rules, semantic-nets in [16], frames in [17] and ontologies in [18]. Rules are formulated to model relationships. The semantic context of strict rules is difficult to map, so there must be a rule for every single relationship, while the context cannot be illustrated. In semantic-nets, knowledge is stored using graphs, where nodes represent objects while the edges (connections) represent the relationships between the objects. Frames were derived from semantic-nets and can be used to represent stereotypical situations, but frames lack formal semantics (see van Harmelen et al. [19]), which means that there is no explicit notation to illustrate the meaning of the knowledge. This can lead to ambiguity. Furthermore, there are ontologies, which are formal definitions of types, properties and relationships of knowledge fragments. Due to their formal semantics, ontologies can be used to unambiguously represent knowledge. The usage of ontologies for HAZOP studies has already been proposed by Batres [20] and Daramola et al. [21]. Other research groups have used ontologies for decision support in [22] and process supervision in [23].

The identification of hazards is conducted based on knowledge models and a digital representation of the process plant. Thus, facts and relationships must be extracted from these knowledge models, and can be achieved using various reasoning techniques. Here, "reasoning" describes the process of drawing conclusions and making inferences. There are various reasoning technique classes that can be used for hazard identification, such as rule-based, case-based, and model-based reasoning (see [24]–[26]). Rule-based reasoners attempt to match rules from a new case with rules from a knowledge base. In case-based reasoning, a previous situation that is similar to the current one is used to solve the current problem. Model-based reasoners use models to draw inferences. These models represent the behavior or structure of a system, and can be quantitative (based on equations) or qualitative (based on a behavioral description). The type of knowledge representation has a direct effect on the reasoning technique and vice versa. For instance, case-based reasoning cannot be directly applied to a knowledge representation that is well-suited to rule-based reasoning.

2 OBJECTIVES

The aim of this research project is to identify possible applications for computer-aided HAZOP systems and develop a comprehensive decision support system. The automatic hazard identification within the decision support system is based on complex knowledge models in the form of ontologies. In addition, different methods for the identification of hazards must be identified and tested, while methods that use artificial intelligence are analyzed.

In this paper, the focus is on the representation of knowledge using knowledge models and algorithmic hazard identification. There are sufficient technologies and data model structures to reuse digital process plant information for hazard identification purposes. Our own approach includes multilayer automatic hazard identification, which is based on an appropriate knowledge representation from a process safety perspective. Hence, the representation of knowledge is considered in particular detail. The following aims are addressed:

- Propose required knowledge domains from a process safety perspective;
- Present a knowledge structure that is suited for computer-aided HAZOP systems;
- Determine and establish different abstraction layers for the identification of hazards;
- Present a conceptual framework for a hazard identification algorithm;
- Demonstrate the methodology within a case study of a technical system; and
- Present the limitations and areas in which further research is needed.

3 METHODOLOGY

In the following section, the knowledge models (3.1), hazard identification on different layers (3.2), ontology implementation (3.3), guiding principles behind the ontologies (3.4) and hazard identification algorithm (3.5) are described.

3.1 Knowledge representation through knowledge models

In order to supply a computer-aided HAZOP system with knowledge, the knowledge representation must contain knowledge from various domains, such as

- Information regarding substances, e.g., hazardous properties or chemical reactivity;
- (Chemical) processes and process-related procedures;
- Process engineering knowledge to support a context-aware structure;
- HAZOP-specific knowledge, e.g., guidewords and the effect of deviations on processes;
- Knowledge regarding process units, equipment, subsystems and environment;
- Typical and atypical failure modes of plant components;
- Potential causes and consequences of failure modes;
- Safeguards to prevent hazardous events to mitigate failures or malfunctions; and
- Information about former events and incidents.

The required knowledge needs to be structured and formulated explicitly, while the semantic contexts must be well-defined. This can be achieved using ontologies for knowledge representation. Within this work, the knowledge representation framework is divided into *conceptual* and *application-oriented* ontologies. Ontologies can be used to make definitions coherent, organize knowledge clearly and unambiguously describe concepts. Hence, they are well-suited to describe relations between concepts and can be formalized using formal semantics. Furthermore, they are machine manipulable, which means that they are machine readable and writable. Currently, the designed ontologies cover the following domains:

- Process engineering principles, e.g., processes and relationships between process-variables;
- HAZOP, e.g., guidewords, deviations, and the propagation of deviations;
- Cause and consequence, e.g., deviation–cause–consequence relationships;
- Material, e.g., hazardous properties and chemical interactions; and
- Plant, e.g., the hierarchical structure of units, equipment or plant components.

3.2 Hazard identification on different layers of abstraction

Within this work, the identification of hazards is based on the knowledge models and a representation of the process plant. With regard to hazard identification, the integral consideration of the plant transpired to be too complex for an algorithm to analyze. This was due to the fact that an integral understanding of the process, the process plant and the involved substances and the interactions between substances requires extensive knowledge. The way in which this knowledge is used for hazard identification is important in order to draw correct conclusions. Thus, the introduction of layers of abstraction makes a search algorithm more efficient, since the consideration of additional information restricts the search range. The identification of hazardous events and operational malfunctions is conducted on four layers in this work: (1) Substance Layer, (2) Specific Unit Layer, (3) Abstract Object Layer and (4) Fault/Hazard Propagation Layer. These layers are illustrated in Fig. 1.

The hazard and malfunction identification at different levels of abstraction brings advantages, such as different degrees of detail, the avoidance of redundancies, the propagation of hazards and malfunctions, and the propagation of cause–effect relationships through the process plant. Furthermore, not all layers are needed to produce a rough estimation of hazards. The layers are described in greater detail below:

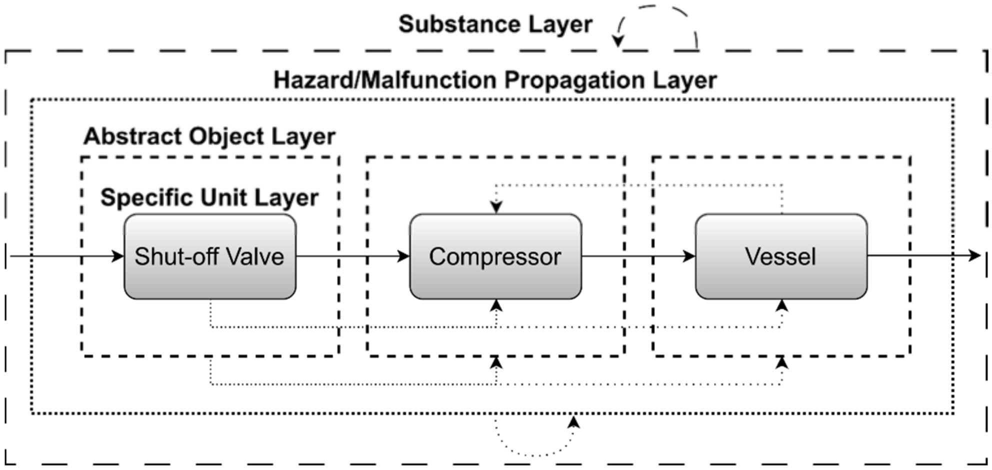


Figure 1: Schematic representation of the different layers of abstraction.

1. Substance Layer: the hazardous properties of substances, e.g., flammable;
2. Specific Unit Layer: equipment-specific hazards and malfunctions, e.g., faulty compressor speed;
3. Abstract Object Layer: direct periphery of equipment (e.g., oil-cooler), operating condition (e.g., start-up operation), type of equipment (e.g., machine), component of equipment (e.g., bearings or seals), and function (e.g., conveying of gases); and
4. Fault/Hazard Propagation Layer: interactions between equipment and the propagation of cause and effect, e.g., more flow + confinement = more pressure.

3.3 Ontology implementation

There are various ontological languages, such as Resource Description Framework (RDF) language and the Web Ontology Language (OWL). The RDF language is used to describe structured information within simple ontologies. OWL represents a compromise between expressiveness and efficient reasoning and is the standard modeling language for ontologies as recommended by the World Wide Web Consortium (W3C) in 2004 (see [27]). Furthermore, an improved version of the language, OWL2, has been produced. The basic elements of the OWL language are *classes* (concepts), *properties* (relations) and *individuals* (instances), while class relationships and restrictions are used to increase the semantic clarity.

The OWL2 ontologies in this work were designed and built using Owlready2 (see Lamy [28]), which is a module for ontology-oriented programming in Python. The HermiT reasoner (see Glimm et al. [29]) was included in Owlready2 and used for inferring logical consequences from the asserted facts and axioms within the ontologies. In this case, ontology-based reasoning was based on the inference rules that were specified by the ontology language, i. e., OWL2.

3.4 Guiding principles behind the designed ontologies

In this work, guiding principles for the modeling of deviation–cause–consequence relationships are developed and used to model relationships in a semantically correct and consistent way. These guiding principles are explained in detail below (see eqns (1)–(5)).

Deviations from the target state of process parameters are described qualitatively. The class *Deviation* is composed of the classes *Parameter* and *Guideword*:

$$Parameter + Guideword \ \Rightarrow Deviation. \tag{1}$$

An exemplary relation of eqn (1) is presented in Table 1 using the Owlready2 (Python) syntax. For instance, the *Deviation* class *HighTemperature* is composed of the classes *More* and *Temperature* and the properties *hasGuideword* and *hasParameters.*

Table 1: Exemplary Owlready2 syntax of the *HighTemperature* class.

```
class HighTemperature(Deviation):
    equivalent_to = [Deviation &
                     hasGuideword.some(More) &
                     hasParameter.some(Temperature)]
```

According to the developed knowledge model concept, *Causes* can be constructed from the *Deviation* and the *Abstraction level*:

$$Deviation + Abstraction\ level \ \Rightarrow Cause. \tag{2}$$

The *Abstraction level* includes specific details about the contexts of the causes, such as the subsystem of the equipment or components and the function of the (process) equipment. Furthermore, an abstraction level represents additional information, such as hazardous properties or the state of aggregation. The semantic context of this is illustrated in Fig. 2. In addition, an exemplary cause is shown in Table 2, where the cause *ExcessiveDischargeTemperature* is constructed from the deviation *HighTemperature* and the properties *takesPlaceIn* and *hasStateOfAggregation*. The semantic context of these properties is also presented in Fig. 2.

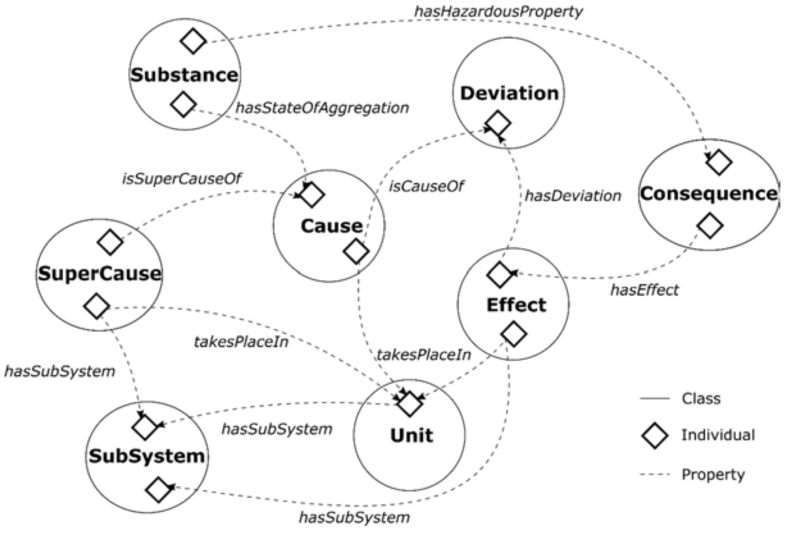


Figure 2: Visual representation of the concept graph of classes, its individuals and its semantic connection through properties as part of a process safety knowledge base.

Table 2: Exemplary Owlready2 syntax of the *ExcessiveDischargeTemperature* cause class.

```
class ExcessiveDischargeTemperature(Cause):
    equivalent_to = [Cause &
                     hasStateOfAggregation.some(Gaseous) &
                     takesPlaceIn.some(Compressor) &
                     isCauseOf.some(HighTemperature)]
```

Causes can have a superordinate cause, which is defined as a *SuperCause* (see Fig. 2):

$$Cause + Abstraction\ level\ \Rightarrow SuperCause. \tag{3}$$

For instance, the cause *ExcessiveDischargeTemperature* could have the superordinate cause *FaultyLubrication*. This distinction enables a more detailed representation of causal relationships.

Deviations can also lead to consequences such as hazardous events or malfunctions. In this approach, (*intermediate*) *Effects* and *Consequences* are distinguished. Within our model, representation consequences follow effects, e.g., the effect is the leakage of a flammable liquid, and the consequence is ignition and fire. Furthermore, *Effect* knowledge models are composed from the deviations and abstraction level details:

$$Deviation + Abstraction\ level \Rightarrow Effect. \tag{4}$$

The exemplary effect model of a *Rupture* is presented in Table 3. In this model, concept effects lead to consequences. For instance, an effect could be a *Rupture*, for which the corresponding consequence could be a *LossOfContainment*. A Consequence (see Fig. 2) is derived from the Effects and the details from the Abstraction level:

$$Effect + Abstraction\ level \Rightarrow Consequence. \tag{5}$$

For instance, the consequence *ToxicRelease* has the effect *LossOfContainment* and the hazardous property *Toxic* within the abstraction level.

Table 3: Exemplary Owlready2 syntax of the Rupture effect class.

```
class Rupture(Effect):
    equivalent_to = [Effect &
                     hasStateOfAggregation.some(Gaseous | Liquid) &
                     (hasDeviation.some(HighTemperature) &
                     takesPlaceIn.some(PressurizedContainment | Piping)) |
                     hasDeviation.some(HighPressure)]
```

3.5 Methodology and hazard identification algorithm

A program flow chart of the proposed method is provided in Fig. 3. First, HAZOP deviations are generated. Then, the first unit is selected and all possible deviations are applied to it. For every deviation possible, (super) causes, effects and consequences are inferred. Thus, all ontologies involved are evaluated simultaneously on abstraction layers 1–3 (see Section 3.4). In case there are more units, the procedure is repeated while the propagation of deviations, hazards or malfunctions between units is considered on the abstraction layer (4).

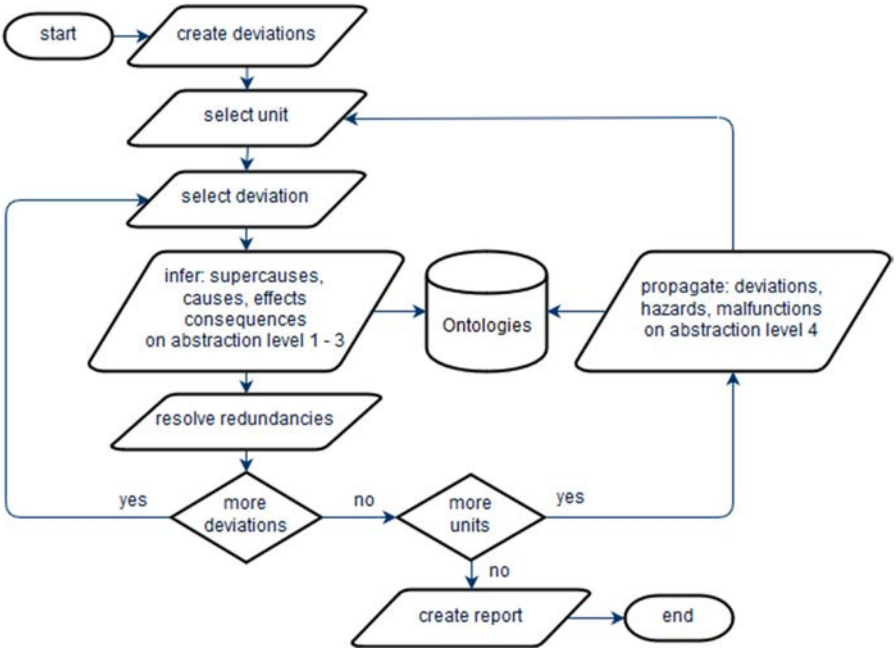
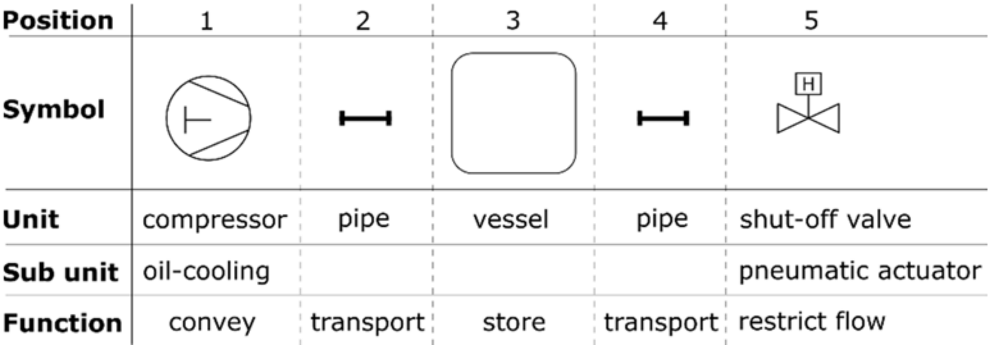

start
create deviations
select unit
select deviation
infer: supercauses, causes, effects consequences on abstraction level 1 - 3
Ontologies
propagate: deviations, hazards, malfunctions on abstraction level 4
resolve redundancies
yes
more deviations
no
more units
yes
no
create report
end

Figure 3: Flow chart of the method for implementing the hazard identification algorithm.

4 CASE STUDY

The methodology presented was applied to a process plant within a case study. Therefore, a comparatively simple system without chemical reactions or a complex arrangement of components was chosen to illustrate the concept and the generated results. This system consists of a compressor, a vessel, a shut-off valve and connection pipes (see Fig. 4) and is operated using air. The compressor takes in (dry) air at ambient pressure (1 bar), compresses it (15 bar) and conveys it into the vessel, which is shut off by a valve. Furthermore, the compressor has an oil-cooling subsystem. Hence, besides air, oil is present in the piston compressor. The shut-off valve is pneumatically actuated.



Position	1	2	3	4	5
Symbol					H
Unit	compressor	pipe	vessel	pipe	shut-off valve
Sub unit	oil-cooling				pneumatic actuator
Function	convey	transport	store	transport	restrict flow

Figure 4: Exemplary compressor-vessel system.

The procedure of the algorithm is explained on the basis of Fig. 3. First, the deviations (see Tables 4 and 5) are generated from the process parameters “flow” and “pressure” and the guidewords “no”, “less”, and “more”. These process parameters and guidewords have only been selected as an example. After the deviations have been generated, the first unit (compressor) is selected and the deviations are applied to find (super)causes, effects and consequences on the *Specific unit layer* and *Abstract object layers*. For instance, the supercause *LiquidHammer* is detected on the *Specific unit layer*. Additionally, an investigation on the *Substance layer* shows that oil has the “flammable” hazardous attribute. Taking the *Abstract object layer* into account, a defective seal (cause) can lead to leakage (effect) and fire (consequence). This coherence is not bound to a specific deviation (see Table 4). Subsequently, the next deviation is applied and the procedure is repeated (see Fig. 3). After the examination of the first unit, the next unit is examined. This procedure is continued until no units remain.

Table 4: Exemplary automatically generated results of the compressor (incomplete).

Plant component: **Piston compressor**

Deviation	**SuperCause**	**Cause**	**Effect**	**Consequence**

noFlow	Insufficient-Lubrication;
	MalfunctionCooling
	PowerFailure;
	LiquidSlugging
		Failure	Faulty-Plant	LossOf-Production

highFlow	MalfunctionSpeed-Control; IncreasedSuction-Pressure	DecreasedDifferential-Pressure	[*Propa-gation*]	None
highTemperature	MalfunctionCooling-System; InsufficientVentilation	ExcessiveDischarge-Temperature	[*Propa-gation*]	None
–	Embrittlement	DefectiveSeal	Leakage (oil)	Fire

Table 5: Exemplary automatically generated results of the vessel (incomplete).

Plant component: **Vessel**

Deviation	**SuperCause**	**Cause**	**Effect**	**Consequence**
highPressure	AbnormalHeatInput	ThermalExpansion; AbnormalGasIntake	Rupture	Blast

lowPressure	Corrosion; Erosion;
	MechanicalDamage;
	DefectiveSeal
		Leakage	LossOf-Containment	None

lowTemperature	FastGasRelaxation	AbruptReliefOf-VesselContent	BrittleFracture	None

From the second unit onward, the *Hazard propagation layer* is considered (see Fig. 3). For instance, valves or pressure raisers are recognized because they can have a direct effect on upstream or downstream equipment.

The consideration of the *Hazard propagation layer* (marked by [Propagation]) shows that the deviation *highFlow* in the compressor and a closed shut-off valve lead to the deviation *highPressure* in the vessel. This connection is not apparent when looking at the individual units (see Tables 4 and 5). The adjustment of the *Specific unit layer* and *Abstract object layers* prevents redundant results and also ensures plausibility, since the context of hazards and malfunctions are given greater consideration on the *Abstract object layer*. For instance, the subsystem oil-cooling is considered on the *Abstract object layer* which led to the potential consequence *Fire* (see Table 4).

5 CONCLUSIONS AND FUTURE WORK

The automation of HAZOP studies, or more precisely, the automation of selected components of HAZOP studies (such as the identification of hazards), has long been the topic of research for more than 30 years. The identification of hazards is a knowledge-intensive process and the quality and semantic richness of knowledge models has a direct effect on the quality and completeness of a safety assessment. In this paper, a knowledge structure for computer-aided HAZOP systems has been proposed. Thus, coherent, context-aware and semantically rich knowledge models in the form of ontologies were designed, formalized and implemented in this research approach, from a process safety perspective. The model was implemented using the OWL ontology language within Owlready2. The semantically correct and complete modeling of deviation–cause–consequence relationships transpired to be particularly important. The hazard identification within this approach was conducted based on the proposed knowledge structure on different layers of abstraction: (1) Substance Layer, (2) Specific Unit Layer, (3) Abstract Object Layer, and (4) Hazard/Malfunction Propagation Layer. The results demonstrate that the investigation on different layers led to more complete and fewer redundant results. Furthermore, the proposed methodology and its principles are not solely limited to process plants and can be applied to other technical systems. Therefore, the ontologies must be extended and refined with further knowledge. Also, case studies of more complex systems, such as chemical reactors, must be conducted in order to validate the proposed methodology. Moreover, the ontologies can be used to model accidents or hazardous incidents and can therefore serve as a coherent long-term digital memory and a comprehensive process safety knowledge base. The focus of future research should be the consideration of chemical reactions and safeguards, the usage of an existing upper ontology, and the enhancement of the hazard identification algorithm using case-based reasoning.

REFERENCES

[1] Kletz, T.A., Hazop: Past and future. *Reliability Engineering and System Safety*, **55**, pp. 263–266, 1997.

[2] Shum, S.K., Davis, J.F., Punch, W.F. & Chandrasekaran, B., An expert system approach to malfunction diagnosis in chemical plants. *Computers and Chemical Engineering*, **12**, pp. 27–36, 1988.

[3] Bassiliades, N., Governatori, G. & Paschke, A., Rule-based reasoning, programming, and applications. *Proceedings of the 5th International Symposium, RuleML*, pp. 305–312, 2011.

[4] Khan, F.I. & Abbasi, S., Tophazop: A knowledge-based software tool for conducting hazop in a rapid, efficient yet inexpensive manner. *Journal of Loss Prevention in the Process Industries*, **10**, pp. 33–343, 1997.

[5] Suh, J.C., Lee, S. & Yoon, E.S., New strategy for automated hazard analysis of chemical plants. Part 1: Knowledge modelling. *Journal of Loss Prevention in the Process Industries,* **10**, pp. 113–126, 1997.

[6] Kang, B., Lee, B., Kang, K., Suh, J. & Yoon, E., Aha: A knowledge based system for automatic hazard identification in chemical plant by multimodel approach. *Expert Systems with Applications*, **16**, pp. 183–195, 1999.

[7] Vaidhyanathan, R. & Venkatasubramanian, V., Experience with an expert system for automated hazop analysis. *Computers and Chemical Engineering*, **20**, pp. 1589–1594, 1996.

[8] McCoy, S.A., Wakeman, S.J., Larkin, F.D., Chung, P., Rushton, A.G. & Lees, F.P., Hazid: A computer aid for hazard identification. *Process Safety and Environmental Protection*, **77**, pp. 328–334, 1999.

[9] Eizenberg, S., Shacham, M. & Brauner, N., Combining HAZOP with dynamic simulation: Applications for safety education. *Journal of Loss Prevention in the Process Industries*, **19**, pp. 754–761, 2006.

[10] Batres, R., Fujihara, S., Shimada, Y. & Fuchino, T., The use of ontologies for enhancing the use of accident information. *Process Safety and Environmental Protection*, **92**, pp. 119–130, 2014.

[11] Rossing, N.L., Lind, M., Jensen, N. & Jørgensen, S.B., A functional hazop methodology. *Computers and Chemical Engineering*, **34**, pp. 244–253, 2010.

[12] Rodríguez, M. & de La Mata, J.L., Automating hazop studies using d-higraphs. *Computers and Chemical Engineering*, **45**, pp. 102–113, 2012.

[13] Zhao, J., Cui, L., Zhao, L., Qiu, T. & Chen, B., Learning hazop expert system by case-based reasoning and ontology. *Computers and Chemical Engineering*, **33**, 371–378, 2009.

[14] ISO 18629-1:2004, Industrial automation systems and integration: Process specification language – Part 1: Overview and basic principles. International Organization for Standardization, 2004.

[15] DIN EN 62424:2017-12, Representation of process control engineering: Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools (IEC 62424:2016). German Institute for Standardisation (DIN, Deutsches Institut fr Normung e.V.), 2017.

[16] Studer, R., Grimm, S. & Abecker, A. (eds), *Semantic Web Services*, Springer: Berlin and Heidelberg, pp. 51–105, 2007.

[17] Minsky, M., A framework for representing knowledge. *Computation and Intelligence,* ed. G.F. Luger, American Association for Artificial Intelligence: Menlo Park, CA, pp. 163–189, 1995.

[18] Bittner, T., From top-level to domain ontologies: Ecosystem classifications as a case study. *Proceedings of the 8th International Spatial Information Theory Conference*, pp. 61–77, 2007.

[19] van Harmelen, F., Lifschitz, V. & Porter, B. (eds), *Handbook of Knowledge Representation*, Elsevier: Amsterdam, pp. 135–179, 2008.

[20] Batres, R., An ontology approach to support HAZOP studies. *Proceedings of the Asian Pacific Confederation of Chemical Engineering Congress Program and Abstract,* 2004.

[21] Daramola, O., Stålhane, T., Omoronyia, I. & Sindre, G., Using ontologies and machine learning for hazard identification and safety analysis. *Managing Requirements Knowledge,* eds W. Maalej & A.K. Thurimella, Springer: Heidelberg, pp. 117–141, 2013.

[22] Baumeister, J. & Striffler, A., Knowledge-driven systems for episodic decision support. *Knowledge-Based Systems*, **88**, pp. 45–56, 2015.

[23] Musulin, E., Roda, F. & Basualdo, M., A knowledge-driven approach for process supervision in chemical plants. *Computers and Chemical Engineering*, **59**, pp. 164–177, 2013.

[24] Dutta, S. & Bonissone, P.P., Integrating case- and rule-based reasoning. *International Journal of Approximate Reasoning*, **8**, pp. 163–203, 1993.

[25] Kolodner, J.L., An introduction to case-based reasoning. *Artificial Intelligence Review*, **6**, pp. 3–34, 1992.

[26] Khardon, R. & Roth, D., Defaults and relevance in model-based reasoning. *Artificial Intelligence*, **97**, pp. 169–193, 1997.

[27] Hitzler, P., Krotzsch, M. & Rudolph, S., *Foundations of Semantic Web Technologies*, CRC Press: New York, pp. 111–210, 2010.

[28] Lamy J.B., Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. *Artificial Intelligence In Medicine*, **80**, pp. 11–28, 2017.

[29] Glimm, B., Horrocks, I., Motik, B., Stoilos, G. & Wang, Z., HermiT: An OWL 2 reasoner. *Journal of Automated Reasoning*, **53**(3), pp. 245–269, 2014.