Contents lists available at ScienceDirect

Process Safety and Environmental Protection

journal homepage: www.elsevier.com/locate/psep

The use of ontologies for enhancing the use of accident information

Rafael Batres[a,*], *Shinya Fujihara*[a], *Yukiyasu Shimada*[b], *Testuo Fuchino*[c]

[a] *Department of Mechanical Engineering, Toyohashi University of Technology, Japan*
[b] *Chemical Safety Research Group, National Institute of Occupational Safety and Health, Japan*
[c] *Department of Chemical Engineering, Tokyo Institute of Technology, Japan*

ABSTRACT

Safety engineering is multi-disciplinary in nature, requiring many kinds of information. Particularly, the identification of accident scenarios and the reuse of accident information can be benefited from the computational integration of different sources of information. However, enabling software tools to share, exchange and search information in this area is difficult due to the lack of an unambiguous knowledge representation. Ontologies are formal models based on mathematical logic that describe classes of things and their relations and can facilitate the sharing and exchange of accident scenarios and bring with them the support of automated reasoning which facilitates the location of information of past accidents. This paper discusses the use of ontologies (and the ISO 15926 in particular) for capturing descriptions of accidents and locating them.

Keywords: HAZOP; Accident scenario; Ontologies; ISO 15926

1. Introduction

Having its origins in ICI in the 1960s (Kletz, 1999), HAZOP (Hazard and Operability Analysis) is one of the most widely used safety analysis techniques in the process industries. Typically, in a HAZOP study, the team of specialists combines guidewords and parameters to suggest possible deviations of the design intent. Then possible causes of the design deviation are identified and for each cause, an assessment of the consequences is carried out. Existing safeguards are identified and when the team judges that inadequate protection exists, actions are recommended. This process is applied to the so-called nodes which often represent major vessels or pipelines linking to major vessels. After all the nodes that are under the scope of the project are analyzed, the result is a series of accident scenarios that are often risk-ranked for further analysis.

A number of tools are available in the market that support the documentation of HAZOP sessions. Broadly speaking, tools that support a HAZOP study can be grouped into two categories: tools that are used to record and document a HAZOP and tools that automate it (Smith and Harrison, 2002). Tools in the first category provide facilities that help to transcript the study as an alternative to word processing software or spreadsheets. One limitation of the tools that fall in this category is that the text format used to represent hazard-analysis information limits the subsequent computer processing of the HAZOP results. For example, references about the equipment where a consequence occurs are often written in the consequence itself. This could be error-prone and limit integration with other engineering tools.

Venkatasubramanian et al. (2000) presented a review of tools in the second category. For example, HAZID (McCoy et al., 1999) can retrieve data from a computer-processable P&ID and generate a full detailed HAZOP report. Automated tools are based on causality models that predict changes in the process variables from unit to unit. Tools in this category do have a formal representation of the equipment and their behaviors with which a detailed accident scenario can be obtained. The challenge with this second category is in the horizontal and vertical integration with other software tools. Horizontal integration ensures that the safety analysis software is

* *Corresponding author at*: Department of Mechanical Engineering, Toyohashi University of Technology, Hibarigaoka 1-1, Tempaku-cho, Toyohashi 441-8580, Japan. Tel.: +81 532 44 6716; fax: +81 532 44 6690.
E-mail address: rbp@tut.jp (R. Batres).
Received 9 February 2012; Received in revised form 11 July 2012; Accepted 15 November 2012
0957-5820/$ – see front matter
http://dx.doi.org/10.1016/j.psep.2012.11.002

adequately interfaced to CAD systems and other design tools. On the other hand, vertical integration requires the information to flow across disciplines and life-cycle stages. Both kinds of integration require robust domain definitions and models for data exchange.

Ontologies, which are "formal models that use mathematical logic to clarify and define" things (Madin et al., 2008), can enhance the sharing and exchange of the HAZOP results between computer systems. For example, when a HAZOP produces scenarios that can be reached from several deviations (Limb, 2009), an ontology-based tool can be used to check for consistency and identify missing parts of a given scenario.

In a different vein, engineers who perform safety analyses can benefit from the significant number of reports of past incidents which are stored in accident databases. Existing accident databases implement a search mechanism that is based on keywords. Typically, an accident database displays the keywords as an expandable list of categories such as causes, damage, and type of incident. The database then finds the records that are classified by those keywords. However, the keyword approach often produces a considerable number of mismatches. For example, querying a typical commercial database to find accidents about "hydrocarbon oil that leaks from distillation columns," produced results of which only 40% were correct answers to the query. Among the results was a record about an accident in which ammonia leaked and caused failure of a crude-oil distillation tower, which is clearly a mismatch. When the database is small, the user can discard the results that are irrelevant but as the database grows in size it becomes difficult to identify those results that satisfy the requirements of the user. Here the use of ontologies can enhance the effectiveness of incident databases to locate data. For example, search for the previous oil query can be improved by means of a location relation that associates the hydrocarbon oil, the leaking and the distillation column.

This paper discusses the use of ontologies based on the ISO 15926 standard in capturing and using accident descriptions. This paper is structured as follows. Ontologies are explained in Section 2. Section 3 describes ISO 15926 with emphasis on the definitions that can be used to represent accident information. The concept of class hierarchy is explained in Section 4. ARC, a graphical representation of accident scenarios based on the ISO 15926, is presented in Section 5. Next, the method for the construction of an ARC diagram is presented in Section 6 and illustrated with a fragment of the hazard analysis of a DAP (diammonium phosphate) production process. Subsequently, the construction of an ARC for a past accident is discussed in Section 7. Section 8 describes an editor for creating ARC charts. Section 9 illustrates the use of ontologies for facilitating the retrieval of past accidents. Finally, Section 10 presents the conclusions and discussion.

2. Ontologies

Ontologies are formal models that describe a shared and common understanding of a domain that can be communicated between people and heterogeneous software tools. Ontologies define classes of things, their taxonomy, and the possible relations between things. A class represents a set of things that share common properties. Class hierarchies are defined with the use of the subclass relation (also known as the *is-a* relation) which states that every member of a subclass is also a member of the superclass, inheriting all the characteristics of the superclass. Ontology developers can also define their own relations to describe how things interact. Examples of relations are whole-part relations, connectivity relations, causality relations, and containment relations.

Ontologies can be used for the following purposes:

- To facilitate sharing and exchange of information.
- To support integration of tools.
- To provide the same perspectives with collaborating teams and tools.
- To facilitate the use of automated reasoning.

Information is more easily exchanged with ontologies because ontologies provide an agreement of the meaning of the terms that are communicated between software components. Also because ontologies are based on mathematical logic, ontologies provide the structure and semantics that ensures the validation of information that is to be communicated. In addition, tools that are developed based on ontologies are easier to integrate and maintain.

Multi-disciplinary information sharing and exchange, such that which could facilitate a HAZOP study, often have diverse but overlapped views that need to be harmonized. The lack of harmonization is a limitation for a consistent sharing and exchange of information between engineering applications. Here, ontologies can be used to verify the consistency of the information that is to be communicated.

Moreover, ontologies can be used together with the so-called logic reasoners or inference engines to deduce logic implications. For example, an accident of an explosion of a 3000 gallon tank containing diazinone was found in a typical accident database. The explosion was probably caused by a cyclohexanone–vapor mixture, which was ignited by welding operations. Looking at the keywords for this particular record shows that this incident was classified under 'welding', tank, 'explosion' and 'management system inadequate'. If a user wished to search for accidents involving chemical explosions, this particular record would be missing in the list of results. However, with the help of a logic reasoner, an ontology of explosions can provide the means that can automatically classify this explosion as a chemical explosion so that the record could be found.

Ontologies are typically encoded in ontology languages which are designed to assist in the realization of the four purposes mentioned above. OWL (Web Ontology Language) is probably the most commonly used ontology language which was originally intended for improving information exchange over the Internet. OWL was developed by the World Wide Web Consortium (W3C) Web Ontology Working Group (McGuiness and van Harmelen, 2004) and is being used to encode knowledge and enable interoperability in distributed computer systems (Finin and Ding, 2006). Several logic reasoners have been developed that support OWL, including JTP (Fikes et al., 2003), FacT++ (Tsarkov and Horrocks, 2006), Pellet (Sirin et al., 2007), and Hermit (Shearer et al., 2008).

Several tools have been developed for editing and building ontologies. One of such tool is the Protégé ontology editor which is a software application for editing, browsing, and deploying ontologies (Tudorache et al., 2008). Protégé has a graphic user interface with which classes, relations and logical axioms can be defined. Ontologies can also be integrated with other ontologies which is particularly useful when using an upper ontology such as the one described in the next section. Once the ontology has been developed, the ontology editor can