A SEMANTIC APPROACH FOR INCIDENT DATABASE DEVELOPMENT

Rafael Batres[1], Yukiyasu Shimada[2], Tetsuo Fuchino[3]
[1]Department of Production Systems Engineering, Toyohashi University of Technology; Tel: +81-532-44-6716, Fax: +81-543-44-6690; e-mail: rbp@pse.tut.ac.jp
[2]Chemical Safety Research Group, National Institute of Occupational Safety and Health; Tel: +81-42-494-6230 Fax: +81-42-491-7846; e-mail: shimada@s.jniosh.go.jp
[3]Department of Chemical Engineering, Tokyo Institute of Technology; Tel: +81-3-5734-2474, Fax: +81-3-5734-2474; e-mail: fuchino@chemeng.titech.ac.jp

The use of ontologies and Formal Concept Analysis (FCA) can improve the effectiveness of incident databases by providing the structure and semantics needed to relate data. Typically, domain ontologies are developed in an ad-hoc fashion, often without sound explanations of the class structure. To avoid this, we propose the use of FCA as way to assist the development of the domain ontology. FCA is an analysis technique for knowledge processing based on applied lattice and order theory. FCA works by processing a collection of objects and their properties to identify hidden relationships represented as concept lattice. A software prototype has been developed to evaluate the proposed approach using knowledge engineering tools.

INTRODUCTION

Based on the experiences of major accidents such as in Seveso, Flixborough, and Bhopal, the adoption of process safety management legislations has resulted in the increase of reporting requirements of accidents. For example, in the United States, regulations such as the Comprehensive Environmental Response, Compensation, and Liability Act (CERCLA) and the Emergency Planning and Community Right-To-Know Act (EPCRA) require reporting when any facility releases more than a specified amount of a hazardous substance to the National Response Center (Mobley, 2001). The National Response Center keeps a large database containing details about the reported incidents (Anand et al. 2006). Similarly, in Europe chemical process industries are required to report major accidents or near misses to the Major Accident Reporting System (MARS) operated by the Major Accident Hazard Bureau (MAHB) (Nivolianitou, 2006).

Also, the Chemical Safety and Hazard Investigation Board (CSB) has made available to the public results of the investigation of accidents in the chemical and process industries (CBS, 2009). A number of freely-available and commercial databases have been developed for searching and retrieving incident information. For example, the Process Safety Incident Database (PSID) which will permit collecting, tracking, and sharing process safety incidents and experiences among participating companies (Sepeda, 2006).

The Japan Science and Technology Agency (JST) maintains a database of accidents and failures that has been growing since 2001 (JST, 2009). As of 2009, the database stores about 1160 records of incidents, including 333 records related to chemical substances, some of which are available in English.

Engineers who perform safety analysis can benefit from the significant number of reports of past incidents maintained in such databases. These databases implement a searching mechanism based on keywords. The user selects keywords which are classified into causes, damage, type of incident, etc. The database then finds those records that are classified by those keywords. However, these databases lack the capability of using the relationships that exist between the keywords. Extracting knowledge directly from such sources has a considerable number of mismatches. False positives are reported when a word has the same spelling but a different meaning such as the word tank. Also, false negatives are obtained because existing incident representations lack the ability to deal with relationships between the keywords. For example, querying a commercial database to find accidents about "crude oil that leaks from distillation columns," produces results of which only 40% are answers to the query. One of the results was a record about an accident in which ammonia leaked and caused failure of a crude-oil distillation tower. 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 only those results that satisfy the requirements of the user.

In this paper we propose an approach that aims at enhancing the effectiveness of incident databases using knowledge engineering techniques. In particular, we suggest the use of ontologies and FCA. Ontologies define the structure of knowledge by defining types and subtypes of things and their relations. For example, search for the previous crude oil query can be improved by means of a location relation that associates the crude oil, the leaking and the distillation column. The upper ontology defines basic classes and properties useful for defining concepts such as physical quantities, causality, substances, equipment, physical and chemical transformations, plant operations and personnel.

METHODOLOGY

The approach for database construction is based on the use of ontologies. Generally speaking, ontologies define the

structure of knowledge by describing in an explicit way the relationships that exist among 'things'. For example, one can use a whole-part relation to define the class pump as equipment that contains an impeller. One of the advantages of ontologies is that they can be processed by data-processing software so that hidden relations between things can be discovered. Domain ontologies describe the structure of knowledge about things such as kinds of explosion, processing equipment, substances, operation modes, process plants, etc. In this approach, the domain ontologies are developed as extensions of a generic upper ontology defined in the ISO 15926 standard. The domain ontologies are developed using FCA which is a data mining method for data analysis and knowledge discovery (Priss, 2005).

ISO 15926

ISO 15926 Part 2 (standardized as ISO 15926-2:2003) specifies an upper ontology for long-term data integration, access and exchange (ISO-TC184, 2003). It was developed in ISO TC184/SC4-Industrial Data by the EPISTLE consortium (1993–2003) and designed to support the evolution of data through time (Batres et al., 2007). The upper ontology was designed to be generic enough for any engineering application but it was developed as a conceptual data model for the representation of technical information of process plants including oil and gas production facilities (ISO-15926-2, 2003).

Every class in the upper ontology is derived from the class *thing* which is divided into two classes: *abstract_object* and *possible_individual*. When something exists in space and time, it can be classified as a *possible_individual*. This includes things that are non-physical such as a policy or physical such as a compressor. Because a *possible_individual* exists in time it has a life cycle that starts by a beginning *event* and ends by an ending *event*. On the other hand, *abstract_object* is a class for those things that do not exist at a particular place and time. Examples include entities such as numbers or sets: *possible_individual* includes classes such as *arranged_individual*, *physical_object*, *activity*, *period_in_time* and *event*.

The class *arranged_individual* is used to describe such things that are made of parts, each of which plays a distinct role with respect to the whole. For example, a centrifugal pump is an *arranged_individual* composed of an impeller and a diffuser. The impeller has the role of imparting velocity head to the fluid and diffuser has the role of capturing the liquid off the impeller. As it can be noted from the example, a *role* indicates what some thing has to do with an activity.

An *activity* is a *possible_individual* that brings about change. Like possible individuals, activities can have a life cycle bounded by beginning and ending events. An event is a *possible_individual* that has zero extent in time, which means that it occurs at an instant in time. For example, an event related through the ending relation to an activity is the culmination of the activity. A *point_in_time* is an event that is zero extent in time.

Aspects related to activities include:

1. A description of the sub-activities that compose the activity
2. All the conditions of an activity that must be satisfied in order for that activity to take place
3. The participants required by an activity
4. The sequence of activities
5. The constraints and requirements of the activity
6. The characteristics of the things produced by or used in the activity
7. The agent that implements the activity (politician, organizations)

Scenarios can be described in terms of a causal relation (*cause_of_event*) which links an activity to an event which can be the beginning of another activity. For example, let us assume that there is an ammonia pipe-line in which a valve fails open. This can be represented as (*cause_of_event* activity1 event1), (*beginning* activity2 event1), where the activities and events are described as follows:

activity1. Actuator-spring sets valve cv-02 to fail-open position.
event1. Valve cv-02 in fail-open position.
activity2. Flow rate of ammonia increasing.

The participation relation is used to express that a possible_individual is involved in an activity.

DOMAIN ONTOLOGIES

Although incident databases include categories of things such as the classes of material or equipment involved in the incident, these categories are developed ad-hoc, which may limit the search of data. For example, a substance release normally would be a class of consequences but it is possible for some incidents to be caused by it. A data analysis method such as FCA provides a systematic way to obtain sound categories. In addition, FCA permits the discovery of key properties about the objects classified in the categories.

FCA is a method for data analysis that was originally developed by Wille et al. (1982). In FCA there are two kinds of elements, namely *formal objects* and *formal attributes*. The sets of formal objects and formal attributes together with a binary relation between them constitute the foundation of FCA that is called *formal context*. Formal contexts can be represented by a cross table such as the table of concepts about separation processes shown in Figure 1. In a cross table, the formal objects are listed in the rows and the formal attributes in the columns of the table. In this example, the objects are some processes commonly found in chemical plants. The attributes define physical characteristics. If a formal object has an attribute, which means that there is a binary relation between them, a checkmark is inserted in that cell.

In FCA concept lattices are used for visualisation of the sets of formal objects, formal attributes, and their

	separating	purifying	involves conversion of vapor to liquid	drived by centrifugal force	separates gas	separates liquid	separates solid	uses filter	drived by heat transfer	drived by relative solubilities	drived by boiling point difference	involves conversion of liquid to gas	drived by magnetic attraction	involves adding a substance	involves conversion of liquid to solid
distillation	×	×	×			×			×		×	×			
centrifugation	×			×		×	×								
filtration	×				×	×	×	×							
evaporation	×					×	×		×		×	×			
solvent extraction	×					×				×					
precipitant–based crystallisation	×					×	×							×	×
magnetic separation	×						×						×		
cooling–based crystallisation	×					×	×		×						×

Figure 1. Cross table of formal concepts about separation processes

relations. The lattice corresponding to the example of Figure 1 is shown in Figure 2. From the formal contexts, it is possible to identify all the formal attributes that a particular set of formal objects has in common. In the separation example, centrifugation and filtration both have the "separates liquid" and "separates solid" attributes, which means that they can be grouped together in the same category. Also, it can be seen that what solvent-extraction and filtration have in common is that both can separate liquids. We can see that in this formal context no other formal objects have both these formal attributes. Thus, when two formal objects have a set of formal attributes and no other object have that set of attributes, the pair of formal objects together with the set of formal attributes is called a *formal concept*.

In the lattice, a node such as node *B* (Figure 2) represents a *formal concept*. Traditionally, the labels of the objects in the node are set slightly below and the labels of the attributes slightly above the node. In order to obtain the set of formal objects of a node we follow all paths which lead down from that node. In this example, the formal objects of node *B* are cooling-based crystallisation, evaporation, and distillation. To obtain the set of formal attributes of a node we trace all paths which lead up from that node. The formal attributes of node *B* are "separates liquid," "driven by heat-transfer," and "separates."

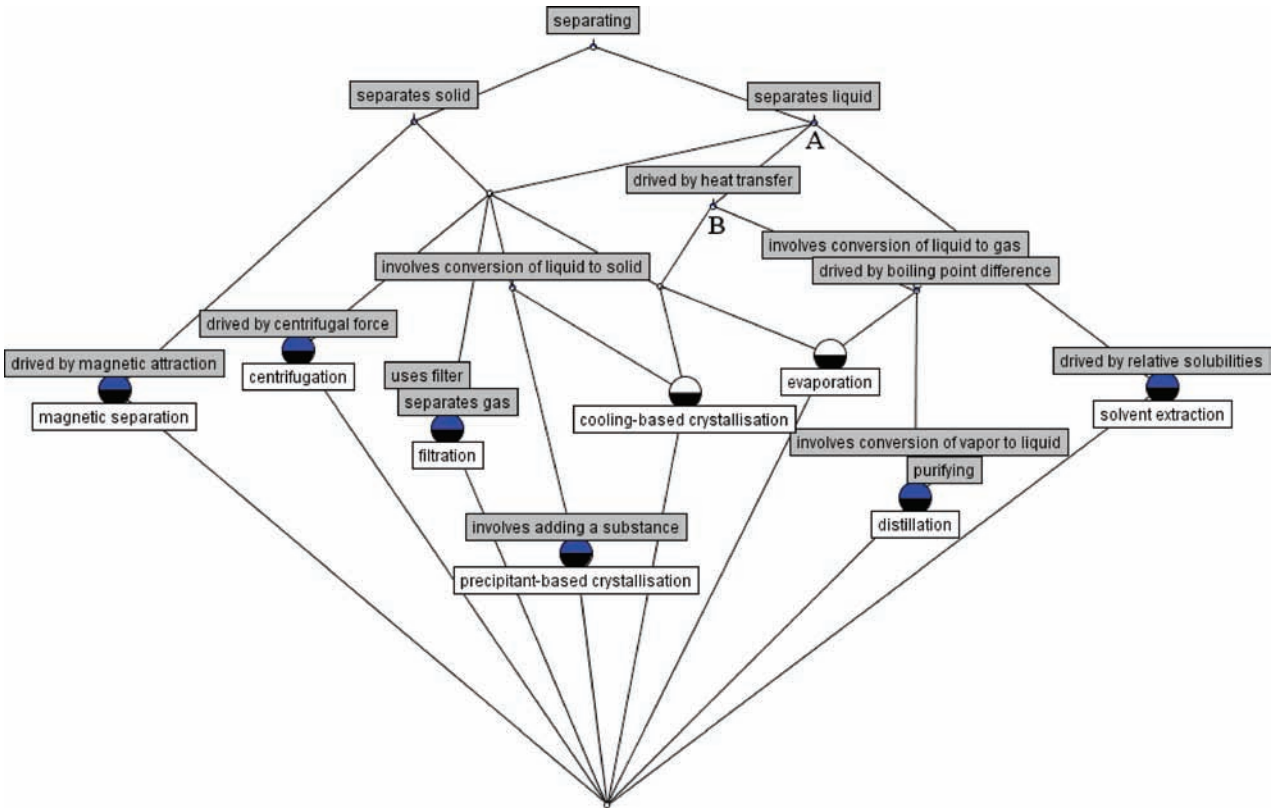

Figure 2. Concept lattice of separation processes

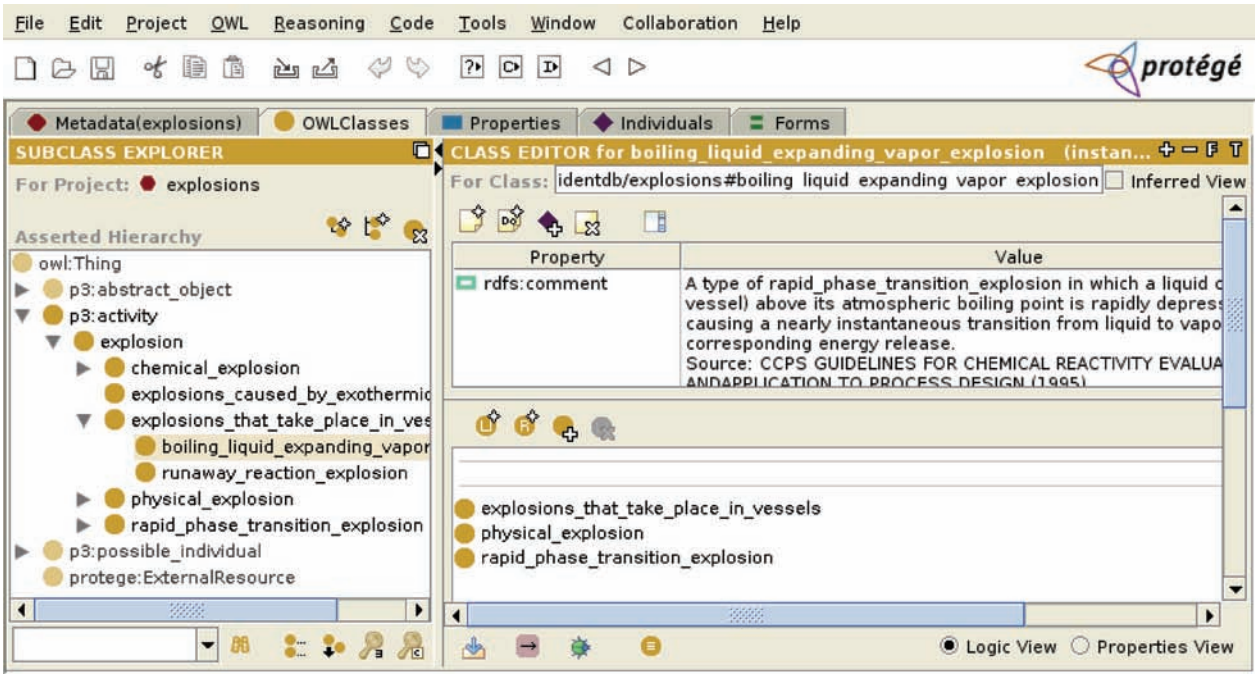

Figure 3. Screen dump of the Protégé ontology editor

An arc in the lattice means that a concept is a subconcept of another concept (superconcept—subconcept relation). In the example, the concept represented as node *B* is a subconcept of *A* (Figure 2).

The top and bottom concepts have a particular meaning. The top concept includes all formal objects of the nodes below. In the example, the top node has the attribute "separates" which means that all the other nodes share that attribute. However, it could be the case that the top concept has no attributes, which may be the case if we expand the processes by including non-separation processes such as chemical reactions. The bottom concept has all the formal attributes of the nodes above. However, if some attributes are mutually exclusive then the bottom concept is a null or empty concept.

The superconcept-subconcept relation is transitive. Consequently, if a node *A* is a subconcept of *B*, and *B* is also a subconcept of *C* (Figure 2) then *A* is a subconcept of *C*. That means that a sub-concept inherits all the attributes from all its super-concepts.

The resulting concept lattices can now be used to construct the ontology. As an example, the Protégé ontology editor is used (Figure 3). Protégé is a tool for editing, browsing, and deploying ontologies (Tudorache et al., 2008). The editor can export the ontologies in the OWL language. OWL is an ontology language originally developed for the Web by the World Wide Web Consortium (W3C) Web Ontology Working Group (Bechhofer, 2004) but can also be used in other computer environments (Finin and Ding, 2006).

The subClassOf relation is used to describe specializations of a more generic class. A class can be defined in terms of the properties that characterize it. For example, if we assert that every filtration is a kind of liquid-solid separation that involves the use of a filter the definition of centrifugal pump can be represented in OWL as follows:

```
(Class filtration
  (subClassOf liquid_solid_separation)
  (subClassOf
  (Restriction composition_of_individual
    (someValuesFrom filter))))
```

EXPLOSIONS ONTOLOGY

This example shows how to obtain an ontology of explosions using the method described above. There are a number of explosion categories mentioned in incident reports. Chung and Jefferson (1998) identified explosion, BLEVE, overpressure, dust explosion, vapour cloud explosion and boiler explosion. Definitions of these and other categories were obtained from CCPS (1995), CCPS (2003), Martin et al. (1999), Perry and Green (1997) and Eckhoff (1997). The cross table is shown in Figure 4. The formal attributes were extracted from the definitions of the sources mentioned above.

After entering the cross table in the software Concept Explorer (Yevtushenko, 2009), the FCA generates the lattice as shown in Figure 5. It is apparent from the lattice that explosion is the most generic concept from which all the other concepts are derived. When a concept has all the attributes that characterize an object in the context table that concept is named after the object. However, for nodes

	involves liquid substantially above its atmospheric boiling point	takes place in a vessel	involves energy release	involves rapid phase transition	does not involve chemical reaction	caused by chemical reaction	caused by exothermic reaction	reaction propagates	reaction propagates at supersonic speed	reaction propagates at subsonic speed	involves extremely rapid chemical reaction	involves rapid chemical reaction	involves flammable material
BLEVE	×	×	×	×	×								
explosion			×										
physical_explosion			×		×								
rapid_phase_transition_explosion			×	×									
chemical_explosion			×			×							
runaway_reaction_explosion		×	×			×	×						
detonation			×			×		×	×		×		
deflagration			×			×		×		×		×	
vapor_cloud_explosion			×	×		×	×	×					×
dust_explosion			×			×	×	×			×	×	×

Figure 4. Cross table of explosion concepts

such as *A*, *B*, *C*, and *D* (Figure 5) there is no object name associated to them but they are characterized by their attributes. Those nodes correspond to newly identified classes. *A* corresponds to those kinds of explosion that take place in a vessel. *B* corresponds to those chemical explosions that are caused by exothermic reactions. *C* is the class of chemical explosions that are caused by propagating exothermic reactions which involve flammable materials. *D* denotes the classes of chemical explosions with chemical reactions that propagate.

The next step is to create the ontology by extending the upper ontology. The top node in the lattice is explosion,

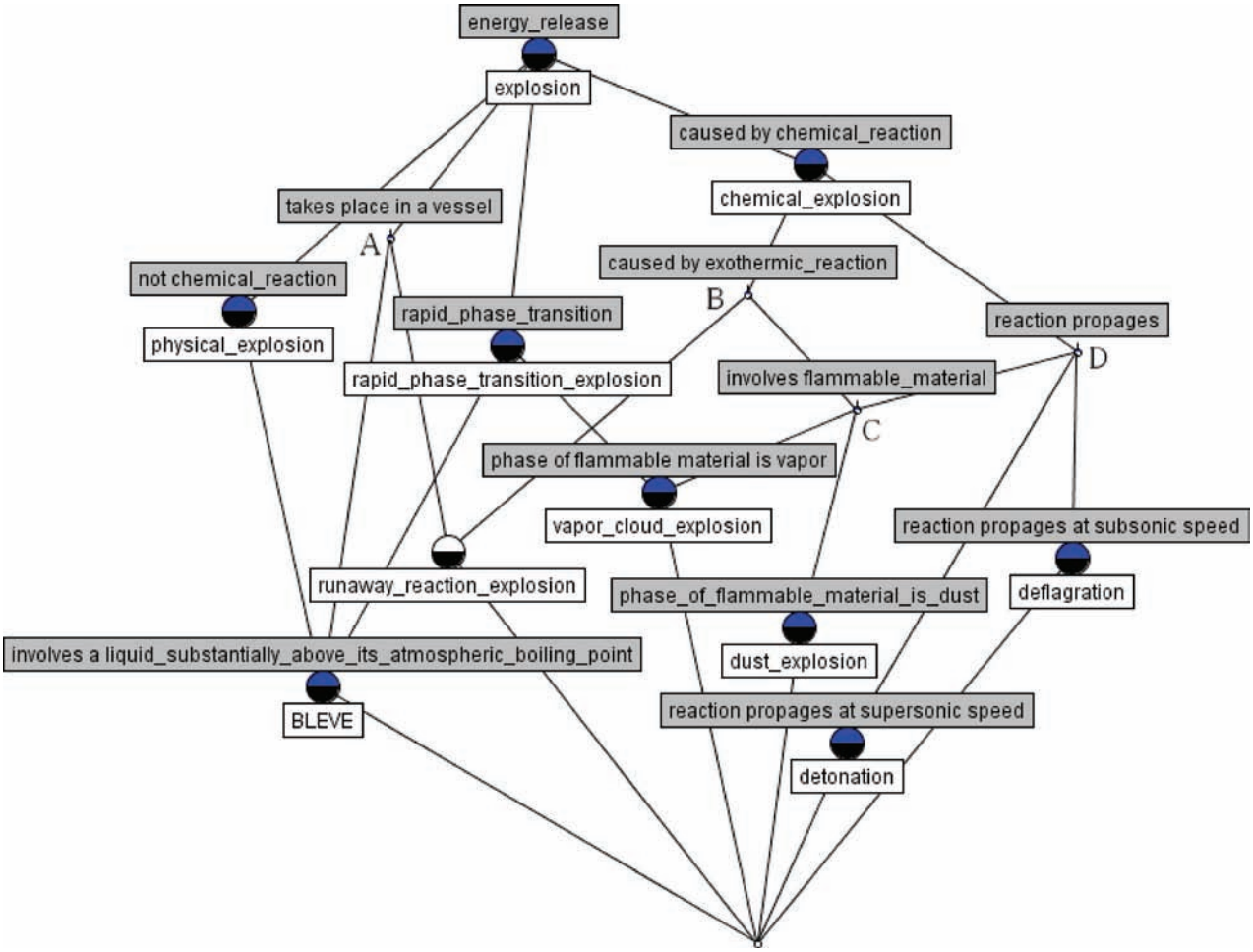

Figure 5. Lattice of explosion concepts

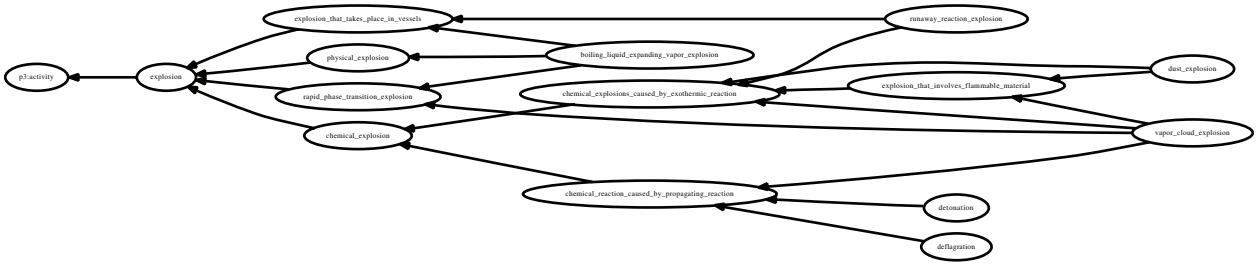

Figure 6. Classes of explosion obtained with the Protégé editor

which can be merged as a subclass of *activity*, because it is bounded by time and brings about changes. Then the newly identified classes are named using information about their attributes and their superconcepts. The resulted classes in the ontology are obtained using the Protégé editor as shown in Figure 6.

PROTOTYPE IMPLEMENTATION

A tool that allows the user to search incident information was developed in the Java programming language (Figure 7). The tool loads ontologies encoded in the OWL language which can be loaded into the Java Theorem Prover (JTP) (Fikes, et al., 2003). In order to search the database, the user constructs queries each of which is formed by combining relationships and classes. For example, to find incidents in which "crude oil that leaks from/to distillation columns" the query is formulated as:

```
(and (relative_location leak_instance
distillation_column_instance)
(participation leak_instance
crude_oil_instance))
```

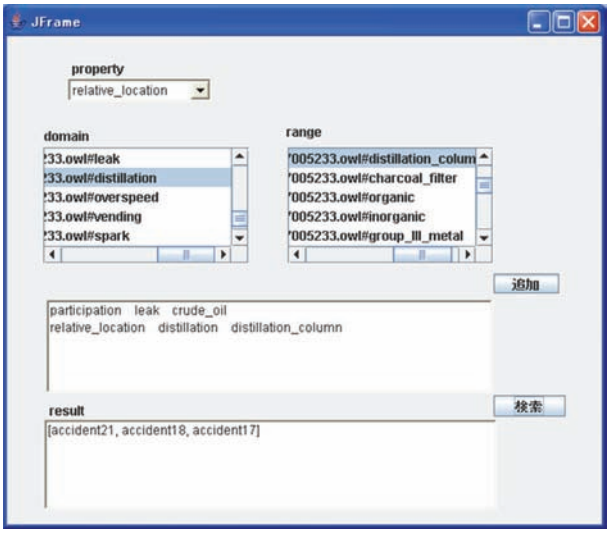

Figure 7. User interface of the prototype

CASE STUDY

Here we discuss an example of the proposed approach based on data extracted from a commercial incident database. The objective is to find past incidents that involve oil that leaks *from* a distillation equipment. The resulted data is summarized in Table 1.

Incidents 1, and 4 are good matches assuming that the bottoms heat exchangers, and pumps are considered part of the distillation equipment. The connectivity relations (*connection_of_individual*) can be used to obtain a numeric score in regards to the degree of matching (for example the closer the leaking equipment to the column the higher the match). Incidents 2 and 5 refer to flushing oil that leaks. Flushing oil enters the 'tower pipe still' in incident 2, while in incident 5 flushing oil leaks directly from the gasket of the flushing oil line. In incident 3, there is a leak but that originates from the desalter.

The query in terms of the ontology formulated as:

```
(and (relative_location leak_instance
distillation_column_instance)
(participation leak_instance
oil_instance))
```

can be used to obtain matches for incidents 1, 2 and 4. Incidents 1 and 4 are marked-up with the whole-part relation between the distillation column and the heat exchangers and pumps. The whole-part relation in the ontology is the one defined in ISO 15926 as *composition_of_individual*. Thus, because the leak is located in the bottoms heat exchangers and the bottoms heat exchangers are part of the distillation equipment then the leak also is located at the distillation equipment. As the desalter and the flushing oil line are not part of the distillation equipment, incidents 3 and 5 can be eliminated from the results. Incident 2 presents more complexity that require definitions not yet developed that would address the direction of the leak and would probably distinguish between the flushing and oil and oil that is normally processed in the distillation operation.

CONCLUSIONS

There is a significant amount of data on past incidents such as that stored and maintained by the National Response Center. As incident databases increase in size, finding useful and relevant information in the database face a

Table 1. Summary of matches obtained by a commercial incident database

	Incident description	Information about the leak	Comments on the commercial database
Incident 1	Leak in a crude oil fractionation unit.	Leak was developed on the inlet of the heat exchangers of the column.	This is a good match.
Incident 2	The incident occurred on a vacuum tower pipe still.	The leak refers to flushing oil that entered the tower through the bottoms pump suction line. The flushing oil entered the suction line from a leaking valve.	The keyword 'still' was necessary to find this record.
Incident 3	A vent fire that occurred on a crude oil unit desalter surge drum.	The desalter surge drum overflows sending oil through its relief valve venting to the atmosphere 10 feet above the surge drum.	The incident was classified under leak and distillation. This is typical example of a false positive.
Incident 4	About a fire at the base of a vacuum distillation column	Bottoms residue pumps leak.	This is a good match
Incident 5	The gasket area of a flushing oil line to a vacuum distillation column bottoms, froze and ruptured the gasket.	Leak of the flushing oil	Another false positive.

number of challenges. Current databases process queries using keywords that are classified into categories such as causes, damage, type of incident, etc. The database then finds those records that are classified by those keywords. However, search often produces a number of mismatches including false positives and false negatives.

An approach has been presented for incident database development based on semantic processing. In other words, incident representations are complemented (marked up) with definitions extracted from ontologies that add the ability to process: 1) the meaning of the things that are mentioned in the reports; 2) relationships between the keywords.

More research is needed to address cases that require finer semantic descriptions such as in incident 2.

This paper also presents a systematic method based on FCA to obtain the categories or classes of things represented in the ontology. This is illustrated by means of an explosion ontology and an separation-process ontology. Future work will also investigate the use of the formal attributes in the context tables of FCA to enrich the ontologies. With this extension, users may formulate queries while avoiding using the class name. For example, a query to refers to equipment used for distillation processes other than distillation columns.

ACKNOWLEDGEMENTS

The work carried out in this paper was supported by a grant from the Japan Society for the Promotion of Science (Grant No. 21310112). The authors would also like to thank the reviewers of this paper for their useful comments and suggestions.

REFERENCES

Anand, S., Keren, N., Tretter, M., Wang, Y., O'Connor, T. M., Mannana, M. S., 2006, Harnessing data mining to explore incident databases, Journal of Hazardous Materials, **130**: 33–41.

Batres, R., West, M., Leal, D., Price, D., Masaki, K., Shimada, Y., Fuchino, T., Naka, Y., 2007, An Upper Ontology based on ISO 15926, Computers and Chemical Engineering, **31**: 519–534.

Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F. and Stein, L., 2004, A. OWL Web Ontology Language Reference, Available from: http://www.w3.org/TR/owl-ref/

CBS, 2009, Chemical Safety and Hazard Investigation Board, Available from: http://www.chemsafety.gov/

CCPS, 1995, Guidelines for Chemical Reactivity Evaluation and Application to Process Design, Wiley-AIChE.

CCPS, 2003, Guidelines for Chemical Reactivity Evaluation and Application to Process Design, Center for Chemical Process Safety/AIChE.

Chung, P. W. H., M. Jefferson, 1998, A Fuzzy Approach to Accessing Accident Databases, Applied Intelligence, **9**: 129–137.

Eckhoff, R.K., 1997, Dust Explosions in the Process Industries (2nd Edition), Elsevier, ISBN: 978-0-7506-3270-6.

Fikes R., Jenkins, J., Gleb, F., 2003, JTP: A System Architecture and Component Library for Hybrid Reasoning, Proceedings of the Seventh World Multiconference on Systemics, Cybernetics, and Informatics. Orlando, Florida, USA, July 27–30, 2003.

Finin, T. & Ding, L., 2006, Search Engines for Semantic Web Knowledge, Proceedings of XTech 2006: Building Web 2.0, Amsterdam, May 16–19, 2006.

ISO 15926-2, 2003, ISO-15926:2003, Integration of lifecycle data for process plant including oil and gas production facilities: Part 2 – Data model.

JST, 2009, Failure Knowledge Database, Available from: http://shippai.jst.go.jp/en/Search

Martin, R. J., Reza, A., Anderson, L. W., 1999, What is an explosion? A case history of an investigation for the insurance industry, Journal of Loss Prevention in the Process Industries **13**: 491–497.

Mobley, R. K., 2001, Plant Engineer's Handbook, Butterworth-Heinemann.

Nivolianitou, Z., Konstandinidou, M., Michalis, C., 2006, Statistical analysis of major accidents in petrochemical industry notified to the major accident reporting system (MARS), Journal of Hazardous Materials **A137**: 1–7.

Perry, R.H., Green, D.W., 1997, Perry's Chemical Engineers' Handbook (7th Edition), McGraw-Hill, ISBN: 978-0-07-049841-9.

Priss, U., 2005, Formal Concept Analysis in Information Science, Annual Review of Information Science and Technology, ASIST, **40**.

Sepeda, A., 2006, Lessons learned from process incident databases and the process safety incident database (PSID) approach sponsored by the Center for Chemical Process Safety, **130**: No. 1–2, 9–14.

Tudorache, T., Noy, N. F., Tu, S. W., Musen, M. A., 2008, Supporting collaborative ontology development in Protégé, Seventh International Semantic Web Conference, Karlsruhe, Germany, Springer.

Wille, R., 1982, Restructuring lattice theory: an Approach based on Hierarchies of Concepts, In: I. Rival (Ed.), Ordered sets. Reidel, Dordrecht-Boston, 445–470.

Yevtushenko, S., 2009, Concept Explorer, Open source java software, Available from: http://sourceforge.net/projects/conexp, Release 1.3.