The final version of this paper appears in *Journal of Chemical Engineering of Japan, Vol. 46, No. 6, pp. 396–406 (2013).*

A Formal Concept Analysis-Based Method for Developing Process Ontologies

Rafael BATRES and Suriati AKMAL
Toyohashi University of Technology, Hibarigaoka 1-1, Tempaku-cho, Toyohashi 441-8580, Japan

Keywords: Ontologies, Processes, Phenomena, Formal Concept Analysis, Explosions Ontology

This paper addresses ontologies of intentional and unintentional processes. Specifically, a methodology for developing processes ontologies is described. Typically, domain ontologies are developed in an ad-hoc fashion, without the reasons and justifications of the class structure. To resolve this issue, we propose a methodology based on Formal Concept Analysis (FCA) as a way to assist the development of a domain ontology. FCA is an analysis technique for knowledge processing based on applied lattice and order theory. The methodology is illustrated with the development of an explosion ontology.

Introduction

Typical chemical engineering textbooks define a process as "an operation or a series of operations" that "cause a physical or chemical change in a substance or mixture of substances" (Felder and Rousseau, 2000). Textbooks also explain that processes commonly have several steps, each of which represents a specific physical or chemical change. Such definitions assume that during the realization of a process, a particular objective is accomplished. In other words, according to these definitions, a process has a design intention.

However, unintentional phenomena are also of concern to chemical engineers. For example, explosions (such as those that result in property damage) may happen as a result of an abnormal situation rather than a well-designed series of steps. Despite differences related to whether an objective is involved or not, both intentional and unintentional processes share the ability to transform material or energy through one or more changes. This paper addresses both kinds of processes. Specifically, a methodology for developing processes ontologies is described.

Ontologies are models based on logic that define the structure of knowledge in terms of classes (types) and subclasses (subtypes) of things and their relations. One of the advantages of ontologies is that they can be processed by knowledge reasoning algorithms so that hidden relations between things can be discovered. In other words, ontologies are useful for generating new conclusions from existing data. In addition, since they have an intrinsic foundation in mathematical logic, ontologies provide the structure and semantics needed for validating information.

Several efforts have been reported on the use of ontologies in chemical engineering. For example, Hailemarian *et al.* (2008) reported the development of ontologies for predicting chemical reactions of drug compounds as an approach for identifying potential reactions between a drug and the excipients and among the excipients that accompany the product. In another related effort, Morbach *et al.* (2009) described how the principles of coherence, conciseness, intelligibility, adaptability, minimal ontological commitment, and efficiency can be used to design a process engineering ontology that is easy to customize, reuse, and extend.

More recently, Batres *et al.* (2009) proposed the use of ontologies as a way to enhance the effectiveness of incident databases. It is worth mentioning that incident databases refer to both intentional processes such as unit operations, and unintentional processes such as runaway-reaction explosions.

Typically, domain ontologies are developed in an ad-hoc fashion, without the reasons and justifications of the class structure. To resolve this issue, we propose an ontology development methodology based on Formal Concept Analysis (FCA).

Corresponding author: Rafael Batres

FCA is an analysis technique for knowledge processing based on applied lattice and order theory (Priss, 2006).

This paper is structured as follows. The methodology is presented in Section 1. Subsequently, the development of an explosion ontology is discussed in Section 2. Finally, the paper ends with conclusions and suggestions for further research.

1. Methodology

The proposed methodology for ontology development involves six steps:

1. Identify potential classes,
2. Identify attributes that are associated to the potential classes,
3. Use FCA to create a concept lattice,
4. Build a class hierarchy based on the concept lattice obtained in step 3,
5. Integrate the class hierarchy with an upper ontology,
6. Add extra classes and constraints to define the meaning of the classes.

These steps are explained in the following subsections.

1.1 Identify potential classes

A class represents a set of objects that share exactly the same properties. This step consists of identifying a list of potential classes for the ontology. Different strategies are possible, including expert consultation, and reviews of technical and scientific literature. Alternatively, text mining tools can be used to process sources that are stored in an electronic form.

For example, an ontology of chemical reactions would include classes such as exothermic reaction, endothermic reaction, polymerization, corrosion, etc.

1.2 Class characterization

Class characterization consists of describing the different qualities or characteristics common to all members of a given class that distinguish them from members of another class.

In this paper, we propose a guideline for the characterization of classes of processes such as chemical reactions, mass transport phenomena, or explosions.

A process can be represented as a member of the class *activity* defined in ISO 15926 (ISO 15926-2, 2003). In ISO 15926, *activity* is defined as a possible individual that brings about change. It can be concluded that:

1. An activity can be composed of other activities (process composition).
2. One or more objects participate in an activity, such as tools and resources.

Similar process definitions can be found in IDEF0 (Marca and McGowan, 2005) and SUMO (Pease, *et al.*, 2012).

In summary, we identify four classes of things for characterizing a process (characterization checklist):

- Objects that are always changed by the process (a.k.a. inputs),
- Objects that are always produced by the process (a.k.a. outputs),
- Participating physical objects (including locations, agents, and performers) other than inputs and outputs (a.k.a. other participating physical objects),
- Sub-activities that compose the process (a.k.a. sub-activities).

The use of the characterization checklist is illustrated with the following examples.

Example 1. According to Crowl (2003), (BLEVE) is defined as “an explosion that occurs when a vessel containing liquefied gas stored at a temperature above its normal boiling point fails catastrophically.” The catastrophic failure of a vessel is defined as a “disruption of the vessel which requires major repair or scrapping” (Lees 1996). Based on these and similar sources, this type of explosion was characterized. The result is shown in **Table 1**.

Table 1 Characterization of BLEVE explosions

Checklist	Characterization
Classes of objects that are always changed by the process	Liquid above its normal boiling point (pressurized liquid), liquid (by implication), vessel
Classes of objects that are always produced by the process	Gas-or-vapor
Classes of other participating physical objects	Vessel
Classes of sub-activities that compose the process	Pressure increase, vessel rupture, flashing, phase change (by implication), vaporization (by implication), vapor expansion, explosion

Example 2. According to Crowl (2003), a runaway reaction is "a reaction that occurs when the heat released by the reaction exceeds the heat removal, resulting in a temperature and pressure increase." In the CCPS Guidelines for Safe Storage and Handling of Reactive Materials (CCPSa, 1995), it is also stated that the heat transfer occurs between the environment and the process container or vessel: "runaways occur when the rate of heat generation from a process exceeds the rate of heat loss to the environment from the process container or vessel." The increase in temperature results in gas generation and an increase in the reaction rate, which in turn are responsible for the pressure increase (CCPSb, 1995). The characterization is shown in **Table 2**.

Table 2 Characterization of runaway reactions

Checklist	Characterization
Classes of objects that are always changed by the process	Liquid-or-gas
Classes of objects that are always produced by the process	Gas
Classes of other participating physical objects	Vessel
Classes of sub-activities that compose the process	Temperature increase, gas generation, pressure increase

1.3 Concept lattice generation

A lattice is a partially ordered set with a least upper bound (also known as supremum) and a greatest lower bound (also known as infimum) (Davey and Priestly, 1990). In this paper, the nodes in the lattice represent classes and the edges represent subclass relations. Formal Concept Analysis (FCA), which is a method based on applied lattice and order theory (Wille, 1982), is selected as the lattice generator.

In order to construct the lattice, FCA requires information to be organized in a so-called *formal context*. A formal context is defined as a set $K := \langle O, A, Y \rangle$, where O is a set whose elements are called formal objects, A is a set whose elements are called formal attributes, and Y is an incidence relation. The relation Y is defined for all pairs $\langle o, a \rangle \in Y$ such that formal object o has formal attribute a as in (*bicycle*, *has wheels*).

Formal contexts can be represented by a cross table, such as the one shown in **Table 3,** or as an incidence matrix. In either case, the formal objects are listed in the rows and the formal attributes in the columns of the table. If a formal object has an attribute, which means that there is a binary relation between them, a checkmark is inserted in that cell. Alternatively, a formal context can be represented by an incidence matrix, by replacing the checkmarks with 1s and empty cells with 0s. In the proposed methodology, the potential classes are considered as formal objects, and the information obtained in the characterization step is considered as attributes.

Table 3 A context table

	att_1	att_2	att_3	att_4	att_5	att_6
ob_1	×					
ob_2		×				
ob_3	×		×	×		
ob_4			×			×
ob_5			×	×	×	

A formal concept is defined as the pair $\langle O_i, A_i \rangle$ such that:

1. $O_i \subseteq O$, $A_i \subseteq A$,
2. Every object in O_i has every attribute in A_i. Conversely, A_i is the set of attributes shared by all the objects in O_i,
3. For every object in O that is not in O_i, there is an attribute in A_i that the object does not have,
4. For every attribute in A that is not in A_i, there is an object in O_i that does not have that attribute.

In other words, a formal concept $\langle O_i, A_i \rangle$ is obtained when

$A' := \{a \in A \mid \langle o, a \rangle \in Y \; \forall o \in O_i\}$

$O' := \{o \in O \mid \langle o, a \rangle \in Y \; \forall a \in A_i\}$

$O_i \subseteq O$, $A_i \subseteq A$, $O' = A_i$, $A' = O_i$

where A' is the set of formal attributes common to all formal objects in O_i, and O' represents the set of formal objects that has all the attributes in A_i. O_i and A_i are respectively the extent and the intent of the formal concept. The formal concepts obtained from the context table, Table 3, are shown in **Table 4**.

Formal concepts can be partially ordered into a lattice, such that a concept is a subconcept of another concept:

$\langle O_i, A_i \rangle \le \langle O_j, A_j \rangle$ iff $A_i \subseteq A_j$.

Table 4 Formal concepts from the context table, Table 3

ID	Formal Concept
c_1	($\{ob_1, ob_2, ob_3, ob_4, ob_5\}$, $\{\emptyset\}$)
c_2	($\{ob_2\}$, $\{att_2\}$)
c_3	($\{ob_1, ob_3\}$, $\{att_1\}$)
c_4	($\{ob_3, ob_4, ob_5\}$, $\{att_3\}$)
c_5	($\{ob_3, ob_5\}$, $\{att_3, att_4\}$)
c_6	($\{ob_4\}$, $\{att_3, att_6\}$)
c_7	($\{ob_3\}$, $\{att_1, att_3, att_4\}$)
c_8	($\{ob_5\}$, $\{att_3, att_4, att_5\}$)
c_9	($\{\emptyset\}$, $\{att_1, att_2, att_3, att_4, att_5\}$)

Several lattice-construction algorithms have been proposed. When the lattice has been obtained, it can be visualized and analyzed. In these algorithms, operations are applied to identify the concepts that can be obtained by the intersection of others.

In FCA, a lattice also serves as a visual aid that helps to explain the relations between the formal concepts. The lattice corresponding to the concepts of Table 4 is shown in **Figure 1**. A circle labeled by an object (a filled circle in **Figure 1**) represents the concept with the smallest extent containing that object. Conversely, a circle labeled by an attribute (a small circle in **Figure 1**) represents the concept with the smallest intent containing that attribute.

From a concept lattice, the set of formal objects of a concept can be obtained by following all the paths that lead down from that concept. For example, the objects of c_3 in Figure 1 are {ob_1, ob_3}. Conversely, to obtain the set of formal attributes of a concept, we trace all the paths that lead up from that concept. For example, the formal attributes of c_7 are {att_1, att_3, att_4}.

As a result, an edge in the lattice means that a concept is a subconcept of another concept (superconcept—subconcept relation). The superconcept-subconcept relation is transitive. Consequently, if a node *A* is a subconcept of *B*, and *B* is also a subconcept of *C*, *A* is a subconcept of *C*. This means that a subconcept inherits all the attributes from all its superconcepts.

The top (supremum) and bottom (infimum) concepts have a particular meaning. The top concept includes all the formal objects of the nodes below. The bottom concept has all the formal attributes of the nodes above.

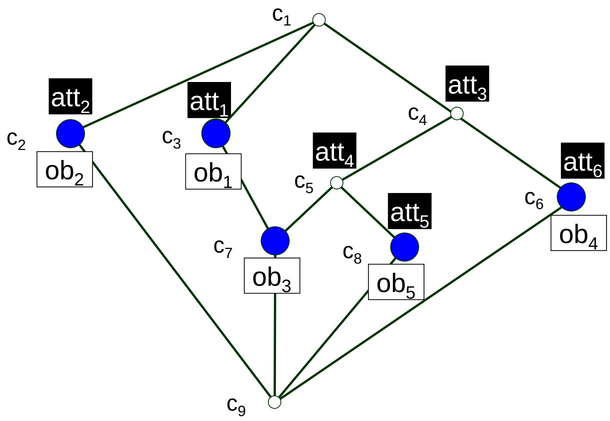

Fig. 1 A concept lattice

1.4 Ontology coding

The resulting concept lattice can now be used to construct the ontology. In this step, ontology tools such as the Protégé ontology editor can be used. Protégé is a tool for editing, browsing, and deploying ontologies (Tudorache, et al., 2008). One useful feature of Protégé is that it can store the ontologies in the OWL language, so that they can be processed by reasoning systems. OWL is an ontology language originally developed for the Web by the World Wide Web Consortium (W3C) Web Ontology Working Group (Bechhofer, *et. al.*, 2004), but it can also be used in other computer environments (Finin and Ding, 2006).

1.5 Integration with the upper ontology and further development

In ISO 15926 Part 2 (standardized as ISO 15926-2:2003), an upper ontology for long-term data integration, access, and exchange is specified (ISO 15926-2, 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. The upper ontology was designed to be sufficiently generic 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. The original ontology was documented in EXPRESS, but it has also been implemented in OWL (Batres, *et al.*, 2007).

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 at a beginning event and ends at 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. In addition, *possible_individual* includes classes such as *arranged_individual*, *physical_object*, *activity*, *period_in_time* and *event*.

The class *arranged_individual* is intended for describing 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 a fluid and the 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 has zero extent in time.

The *participation* relation is used to express that a *possible_individual* is involved in an activity. An activity consists of the temporal parts of those members of the possible individual that participate in the *activity*. For example, a reactor during a runaway reaction has both a beginning and an end. The temporal parts of the participating entities are also possible individuals.

Causality is described by means of the *cause_of_event* relation. For convenience, in this paper we define the relation *event_caused_by_activity* as the inverse relation of *cause_of_event*. Similarly, we define *activity_of_beginning_event* as the inverse relation of *beginning*.

A causality relation that associates two activities is not defined in the standard, but it can be easily implemented. In general, causality between activities follows the properties described by Shoham (Shoham, 1998) some of which are listed here:

1. Causality is antisymmetric. Activity A cannot cause activity B if B is the cause of A,
2. Causality is irreflexive. A cannot cause itself,
3. Causes cannot succeed their effects in time. $A(s)$ causes $B(t) \Rightarrow s \prec t$,
4. Entities participating in a causal relation have a temporal dimension. In other words, they are bounded by a beginning and an ending,
5. Domotor adds the property of transitivity: If A causes B and B is the cause of C, A is also the cause of C (Findler and Bickmore, 1996).

Based on these properties, causality between two activities can be expressed by introducing a new relation, *caused_by_activity*, which must be transitive, asymmetric, and irreflexive. However, based on the OWL2 specifications (Motik, *et al.*, 2009), DL reasoners that comply with OWL2 cannot allow a relation that is transitive to be both asymmetric and irreflexive.

In reality, this is not an issue if transitivity is expressed in terms of SWRL rules (Horrocks, *et al.*, 2004)(Horrocks, et al. 2004) which combine *event_caused_by_activity* and *activity_of_beginning_event*:

Rule 1. `activity_of_beginning_event(?ev1, ?act2), event_caused_by_activity(?act1, ?ev1), event_caused_by_activity(?act2, ?ev2)⇒ event_caused_by_activity(?act1, ?ev2)`

Rule 2. `activity_of_beginning_event(?ev1, ?act2), event_caused_by_activity(?act1, ?ev1) ⇒ caused_by_activity(?act2, ?act1)`

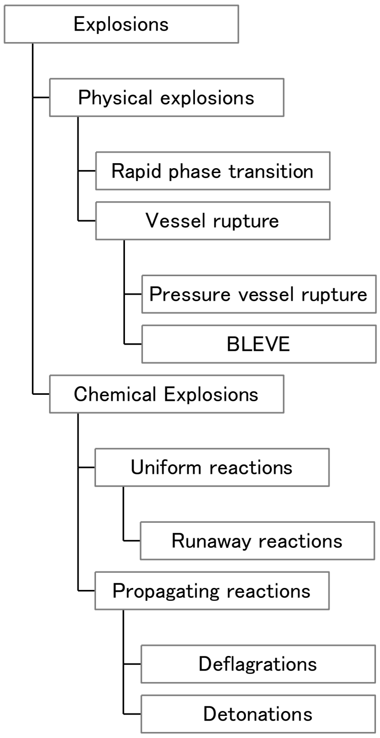

Fig. 2 Crowl's classification of explosions.

Therefore, it becomes possible to define *caused_by_activity* as an asymmetric and irreflexive relation. The definition in OWL is:

```
AsymmetricProperty:caused_by_activity
IrreflexiveProperty:caused_by_activity
```

The above expressions will be used in the example in Section 2.

1.6 Adding axioms to the ontology

Axioms are elements in the ontology that formally define the classes and relations by constraining their interpretation (Grüninger and Fox, 1995). In addition, axioms increase the ability of the computer program to infer new conclusions from existing data. As stated by Morbach *et. al.* (2009), excessive axiomatization can lead to ontologies that cannot be used in a large number of application contexts. In order to avoid this situation, axioms must be used only to define the classes and not to tailor them to a specific purpose. In the proposed methodology, axioms are derived from the results of the formal concept analysis, whose objective is to define the different classes of things in the ontology. Consequently, only the necessary axioms are implemented.

A class can thus be defined with axioms based on the formal attributes associated with 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 filtration can be represented in OWL as follows:

```
Class filtration:
   SubClassOf:
      liquid_solid_separation
   SubClassOf:
      participating_individual some filter
```

where the last statement is an axiom that expresses that one of the participating individuals in a filtration must be of class "filter".

2. Explosion Ontology

The example given below shows how to develop an ontology of explosions using the method described above.

Several sources in the literature were consulted to construct the context table. However, care was taken to select those definitions that describe inputs, outputs, other participating objects, and/or subactivities.

Martin *et al.* (2000) analyzed several established definitions and concluded that an explosion must exert significant pressure-volume work (pV work) on the environment. They identified two sources of energy for the pV work: exothermic reactions and expansions of a compressed gas.

Many explosion categories are mentioned in the literature. Chung and Jefferson (1998) used text mining and data mining techniques to analyze past incident reports and found several categories. The explosion categories consisted of BLEVE, overpressure, dust explosion, vapor cloud explosion, and boiler explosion. In the scientific and engineering literature, other explosion categories can be found including rapid phase transition, chemical explosions, detonation, deflagration, and runaway reaction explosions. Definitions of these categories were obtained from Crowl (2003), CCPSa (1995), CCPSb (1995), Lees (1996), Martin *et al.* (2000), and Eckhoff (1997). In particular, Crowl (2003) provided an extensive description of different classes of explosion and was the sole author to provide a picture showing the relationships among the different types of explosion, as shown in **Figure 2**.

BLEVE is defined as an explosion "that occurs when a vessel containing liquefied gas stored at a temperature above its normal boiling point fails catastrophically." Abbasi and Abbasi (2007) explained that a BLEVE involves the following subprocesses: vaporization of the liquid, pressure increase, vessel rupture, flashing, vapor expansion, and blast wave generation. In order for the flashing, vapor expansion, and explosion to occur, the liquid in the vessel must be superheated at a temperature above its superheat limit temperature (SLT). A BLEVE is often accompanied by a large fireball when a flammable liquid is involved (CCPSb 1995). Vessels affected by BLEVE explosions include process vessels, pipelines, rail road tank cars, and tankers (Abassi and Abassi, 2007).

Overpressure explosions are better known as vessel rupture explosions, which occur when a vessel containing a pressurized material fails suddenly (Crowl 2003). The pV work involved when a pressurized vessel ruptures is due to the sudden expansion of the compressed gases and vapors inside the vessel.

A dust explosion is a phenomenon in which a release of mechanical energy is generated by the combustion of dust (Crowl, 2003); (Carson and Mumford, 2003). In BS 2955: 1958, dust is defined as a material with a particle diameter of less than 76 microns (Lees, 1996). In OSHA the condition that the dust must be in a confined space (such as a container, room, or piece of equipment) for a dust explosion to take place is added (OSHA, 2005).

A vapor cloud explosion occurs when flammable gas is released, mixes with air, and ignites with sufficient energy to create an overpressure (Crowl, 2003); (Lees, 1996). The ignition causes a flame that propagates. In most cases, the mode of flame propagation is a deflagration explosion (see below), but in a few cases a detonation can occur (Davletshina and Cheremisinoff, 1998). In a study of 205 vapor cloud explosions, Lenoir and Davenport (1993) concluded that a high degree of containment or obstruction is observed.

Boiler explosions occur when a boiler fails catastrophically. As boiler explosions occur when holding superheated water, Abbasi and Abbasi (2007) explained that boiler explosions may be considered BLEVE explosions.

A rapid phase transition explosion occurs when a liquid or solid undergoes a very rapid change in phase that results in a change in material volume (vapor expansion) (Crowl 2003). For example, when very hot oil in a pipe comes into contact with water, it can cause the water to flash explosively.

In a chemical explosion, an exothermic chemical reaction, such as a combustion reaction or a decomposition reaction, causes an explosion (Crawl, 2003). The reaction medium can be in either vapor, liquid, or solid phases.

Detonations and deflagrations are also referred to as propagating reactions, because in these phenomena the chemical reaction propagates spatially through the reaction mass. In a detonation, the reaction front propagates at a rate equal to or exceeding the speed of sound (Crowl, 2003); (CCPSb, 1995).

A deflagration is a process where the reaction front is propagated at subsonic speeds. There is also a well-known difference in the propagation mechanism. In a detonation, the reaction front is propagated mainly by compressive heating of the unreacted gases (Crowl, 2003); (Carson and Mumford, 1994). On the other hand, the reaction front of a deflagration propagates mainly by heat conduction and free-radical diffusion. Combustion is the most common type of reaction in detonations and deflagrations. However, other types of reactions also occur. For example, peroxides and organometallic chemicals can undergo a violent decomposition leading to either deflagrations or detonations (CCPSb, 1995).

A runaway-reaction explosion occurs when the heat released by the reaction exceeds the heat that is removed, resulting in temperature and pressure increase (Crowl, 2003). The increase in temperature results in gas generation and an increase in the reaction rate, which in turn are responsible for the pressure increase (CCPSa, 1995).

Finally, a physical explosion occurs due to the sudden release of mechanical energy and does not involve a chemical reaction (Crowl, 2003); (Lees, 1996).

In order to avoid any bias in the development of the class hierarchy, physical explosion and chemical explosion categories were deliberately omitted.

Using the checklist explained in Section 3, information from the previously mentioned definitions was organized in tabular form, as shown in **Table 5**.

Subsequently, a context table was created. Each formal attribute was obtained by combining a check-list category with an individual item. For example, flashing, which is listed in the subactivites column for BLEVE explosions, becomes "subactivity is flashing". When all the formal objects and formal attributes had been inserted in the context table, the incidence relations were added. The final context table is shown in **Table 6**.

After the context table had been prepared, the lattice was generated by means of an FCA algorithm. In this example, the software Concept Explorer which implements the Grail algorithm (Yevtushenko, 2004) was used. The lattice of explosions is shown in **Figure 3**. It is apparent from the lattice that explosion is the most generic concept from which all the other concepts are derived. Most of the lower nodes correspond to objects in the context table. 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 A to I, there is no associated object name. These nodes correspond to newly identified classes. A corresponds to the category of chemical explosions described in the literature. D denotes the classes of chemical explosions with chemical reactions that propagate. E represents the class of explosions that involve combustion. F is the class of chemical explosions that occur in a confined space. G corresponds to explosions in which a vessel participates. H represents the class of vapor expansion explosions that involve phase change. I represents propagating explosions that have a reaction front.

The next step was to integrate the resulting class hierarchy with the upper ontology. The top node in the lattice is explosion, which can be merged as a subclass of activity, because it is bounded by time and brings about changes. The classes in the ontology were defined using the Protégé editor, as shown in **Figure 4**. Then, axioms were added as part of the formal definition of the classes in the ontology. In the proposed methodology, axioms are based on the textual descriptions in the formal attributes of a given class. Below are some examples using the Manchester syntax (Horridge and Patel-Schneider, 2012).

Table 5 Characterization of explosion classes

Potential class	Inputs	Outputs	Other participating physical objects	Subactivities	Notes
explosion		gas-or-vapor		blast	
bleve	liquid substance, liquid above its normal boiling point	gas-or-vapor	vessel, road tank car, tanker, confined unit	vaporization, phase change, pressure increase, vessel rupture, flashing, phase change, vapor expansion, blast	phase change is inferred from vaporization
rapid phase transition explosion	liquid-or-solid substance	gas-or-vapor		phase change, vapor expansion	
runaway reaction explosion		gas-or-vapor	vessel, confined unit	chemical reaction, exothermic reaction, rapid increase in temperature and reaction rate, blast	
detonation		reaction front, supersonic reaction front, propagating material, gas-or-vapor		exothermic reaction, chemical reaction, reaction front propagates by compressive heating	1. chemical reaction is inferred from exothermic reaction 2. although detonation and deflagration are sometimes categorized as combustion processes, detonation and deflagration without combustion has also been observed.
deflagration		reaction front, subsonic reaction front, propagating material, gas-or-vapor		exothermic reaction, chemical reaction, reaction front propagates by conduction and free radical diffusion	chemical reaction is inferred from exothermic reaction
vapor cloud explosion	flammable material, vapor cloud	propagating material, gas-or-vapor		combustion, exothermic reaction, chemical reaction	exothermic reaction and chemical reaction are inferred from exothermic reaction
dust explosion	dust	gas-or-vapor	confined unit (building, room, vessel, process equipment, etc.)	combustion, exothermic reaction, chemical reaction	exothermic reaction and chemical reaction are inferred from exothermic reaction
vessel rupture explosion		gas-or-vapor	vessel	vapor expansion	also known as overpressure explosion.

Table 6 Context table for the explosion classes

	takes place in a vessel	takes place in a confined unit	transforms a liquid	transforms a liquid above its NBP	transforms a liquid-or-solid	transforms flammable material	transforms a vapor cloud	transforms dust	produces gas-or-vapor	produces a reaction front	produces propagating material	produces supersonic reaction front	produces subsonic reaction front	subactivity is flashing	subactivity is phase change	subactivity is vaporization	subactivity is vapor expansion	subactivity is chemical reaction	subactivity is exothermic reaction	subactivity is rapid increase in temperature	subactivity is rapid increase in reaction rate	subactivity is combustion	subactivity is vessel rupture
explosion	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
rapid phase transition	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0
runaway reaction explosion	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0
detonation	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0
deflagration	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	1	1	0	0	0	0
vapor cloud explosion	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	1	1	0	0	1	0
dust explosion	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	1	0
BLEVE	1	1	1	1	0	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	1
vessel rupture explosion	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Definition of explosion_with_subactivity_in_ vessel. One of the parts of any member of this class of explosion is an activity that is located in some vessel (from the respective formal attribute):

```
composition_of_individual some
    (activity and
    (containment_of_individual some vessel))
```

Definition of explosion_that_involves_ exothermic_reaction. The blast in the whole explosion process is caused by an exothermic reaction (from the respective formal attribute):

```
composition_of_individual some
 (blast and
 (caused_by_activity some exothermic_reaction))
```

Definition of run_away_explosion. A runaway reaction explosion is a chemical explosion in which a blast is caused by an exothermic reaction that takes place in a vessel:

```
chemical_explosions_that_involve_exothermic_reaction
 and composition_of_individual some
    (exothermic_reaction
     and (containment_of_individual some vessel))
```

3. Verification of the Explosion Ontology

Let us assume that explosion_1 is an instance of an explosion that is composed of exothermic reaction react_1, intermediate activity act_2, and blast b_1. The chemical reaction takes place in vessel v_1. The causality of the sub-activities is reaction_1 → act_1 → b_1. **Figure 5** shows a graphical representation of these instances. Here, ev_1 and ev_2 are events that are caused by react_1 and act_2, respectively. From the textbook definitions, we know that these instances may describe a runaway reaction explosion. The objective of this experiment was to verify that a reasoning engine can arrive at the same conclusion. In this experiment, we used Hermit, which is a general-purpose reasoning engine that implements the hypertableau algorithm (Shearer, *et al.*, 2008). The significant performance improvement of the hypertableau algorithm means that ontologies that previously required minutes or hours to classify can often by classified in seconds.

Figure 6 shows the classification of explosion_1 before and after running the reasoner. After running Hermit, explosion_1 is automatically classified as a runaway_reaction. This conclusion is obtained from the following axioms and facts as listed in the inference-explanation report of Hermit:

- `act_2 event_caused_by_activity ev_2`
- `activity_of_beginning_event(?ev1, ?act2), event_caused_by_activity(?act1, ?ev1), event_caused_by_activity(?act2, ?ev2) -> event_caused_by_activity(?act1, ?ev2)`

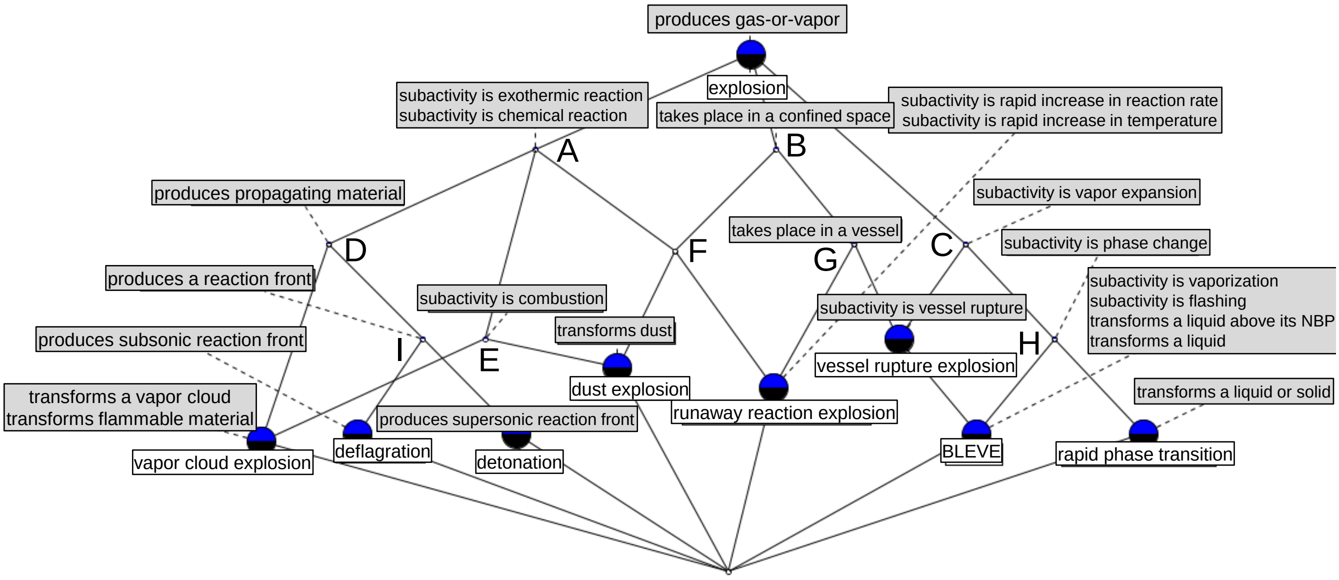

Fig. 3 Concept lattice for the explosion ontology

- `activity_of_beginning_event(?ev1, ?act2), event_caused_by_activity(?act1, ?ev1) -> caused_by_activity(?act2, ?act1)`
- `b_1 Type blast`
- `chemical_explosions_that_involve_exothermic_reaction EquivalentTo composition_of_individual some (blast and (caused_by_activity some exothermic_reaction))`
- `composed_of EquivalentTo composition_of_individual`
- `contained_in EquivalentTo containment_of_individual`
- `ev_1 activity_of_beginning_event act_2`
- `ev_2 activity_of_beginning_event b_1`
- `expl_1 composed_of b_1`
- `expl_1 composed_of react_1`
- `react_1 Type exothermic_reaction`

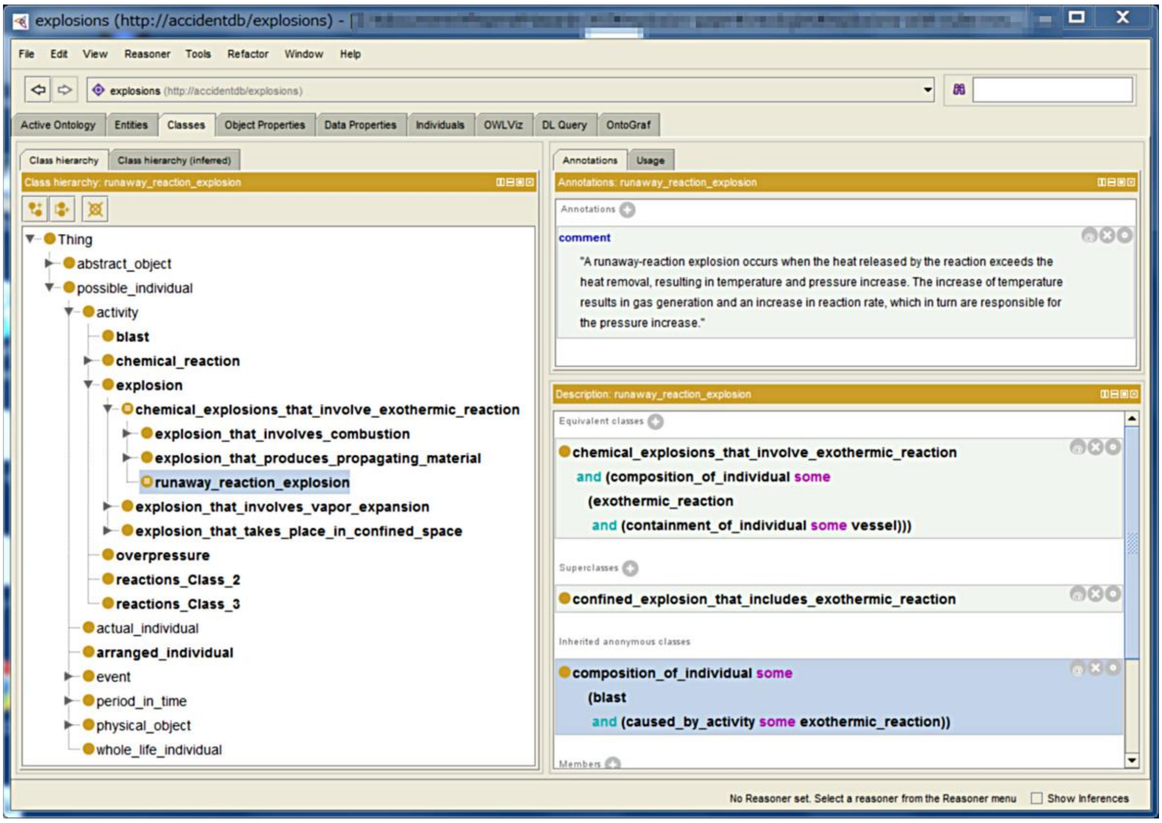

Fig. 4 Protégé editor showing the definition of runaway_reaction_explosion

- `react_1 contained_in v_1`
- `react_1 event_caused_by_activity ev_1`

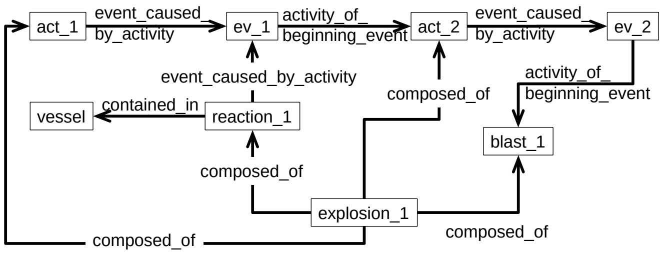

Fig. 5 Instances for an unknown explosion

- `runaway_reaction_explosion EquivalentTo`
- `chemical_explosions_that_involve_exothermic_reaction and (composition_of_individual some (exothermic_reaction and (containment_of_individual some vessel)))`
- `v_1 Type vessel`

The OWL language that we used to encode the ontologies is based on the open world assumption (Sirin, et al., 2008) under which information is always assumed to be incomplete. This is in contrast to the closed world assumption of databases, which assumes that all that is unknown to the database is false. The advantage of the open world assumption is that information systems can be developed incrementally by incorporating new knowledge. For example, after developing a chemical reaction ontology, it may be possible to draw new conclusions with no or some minor changes to the explosion ontology to which it is related.

Conclusions

This paper presented a systematic method based on Formal Concept Analysis to develop process ontologies. Using it, not only the class hierarchy can be obtained but also the axioms that define the different classes in the ontology. Consequently, the soundness of the conclusions derived from the axioms is controlled by the meaning of the classes, which in turn can be verified by visually inspecting the lattice. This approach contrasts with existing ontology development practice in which both the class hierarchy and the axioms are decided in an ad-hoc fashion. The open world assumption enables an incremental development that can be carried out for several process ontologies, including chemical reactions, separation processes, and plant operations.

Well-designed ontologies can be used in numerous applications. For example, ontologies can improve the efficiency of search in past accident data as a replacement of keyword-based approaches, which produce a number of mismatches.

Another use is for determining facts that can be used for legal purposes, such as in the problem described by Martin *et al.* (2000) in which an insurance company has to determine whether an incident that occurred in a hydroelectric power plant was indeed an explosion.

Having said that, the proposed methodology provides assistance for ontology development but it does not release the ontology developer from conducting a literature survey, and unbiased analyses of the textual definitions. The latter is a particularly challenging task due to the fact that many textbook definitions include contradictory statements. In addition, some definitions also tend to evolve with time as new knowledge is generated. For example, in the past, engineers considered vapor cloud explosions

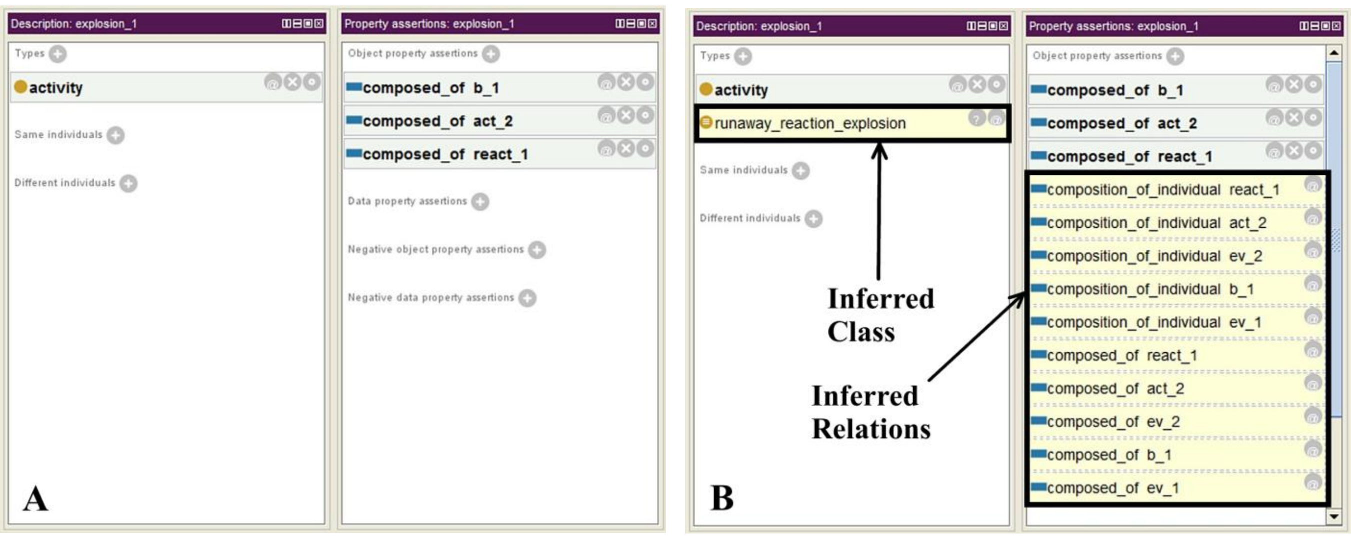

Figure 6. Classification of explosion_1. *A* is the classification before running the reasoner and *B* is the classification after running the reasoner

as unconfined explosions until research showed that a certain degree of confinement is necessary for the explosion to occur.

Finally, the need for unbiased analysis suggests that a mechanism for collaborative development of chemical engineering ontologies is needed. We believe that a framework similar to that of Wikipedia or open source software can be such mechanism.

Acknowledgements The study described in this paper was supported by a grant from the Japan Society for the Promotion of Science (Grant No. 21310112). The authors are grateful to the reviewers and the editor for their valuable comments.

Literature Cited

Abbasi, T. and S. A. Abbasi; "The Boiling Liquid Expanding Vapour Explosion (BLEVE): Mechanism, Consequence Assessment, management," *J. Hazard. Materi.*, **141**, 489-519 (2007)

Batres, R., M. West, D. Leal, D. Price, K. Masaki, Y. Shimada, T. Fuchino and Y. Naka; "An Upper Ontology based on ISO 15926," *Comput. Chem. Eng.*, **31**, 519-534 (2007)

Batres, R., T. Fuchino and Y. Shimada; "A Semantic Approach for Incident Database Development," IChemE Symposium Series, **155**, 477-484 (2009)

Bechhofer, S., F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider and L. A. Stein; "OWL Web Ontology Language Reference," February 10 2004, http://www.w3.org/TR/owl-ref/ (2004)

Carson, P. A. and C. J. Mumford; Hazardous Chemicals Handbook, Elsevier, Oxford, U.K (1994)

CCPSa; "Guidelines for Safe Storage and Handling of Reactive Materials," Center for Chemical Process Safety/AIChE, New York, USA (1995)

CCPSb; "Guidelines for Chemical Reactivity Evaluation and Application to Process Design," Center for Chemical Process Safety/AIChE, New York, U.S.A (1995)

Chung, P. W. H. and M. Jefferson; "A Fuzzy Approach to Accessing Accident Databases," Applied Intelligence, **9**, 129–137 (1998)

Crowl, D. A.; "Understanding Explosions," Center for Chemical Process Safety/AIChE, New York, USA (2003)

Davey, B. A. and H. A. Priestly; Introduction to Lattices and Orders, Cambridge University Press, Oxford, U.K (1990)

Davletshina, T. A. and N. P. Cheremisinoff; Fire and Explosion Hazards Handbook of Industrial Chemicals, William Andrew Publishing/Noyes, New Jersey, U.S.A (1998)

Eckhoff, R. K.; Dust Explosions in the Process Industries, 2nd Edition, Elsevier, U.S.A (1997)

Felder, R. M. and R. W. Rousseau; Elementary Principles of Chemical Processes, John Wiley & Sons, Inc., U.S.A (2000)

Findler, N. and T. Bickmore; "On the Concept of Causality and a Causal Modeling System for Scientific and Engineering Domains, CAMUS," *Applied Artificial Intelligence*, **10**, 5, 455-487 (1996)

Finin, T. and L. Ding; Search Engines for Semantic Web Knowledge, Proceedings of XTech 2006: Building Web 2.0, Amsterdam, The Netherlands (2006)

Grüninger, M. and M. Fox; "Methodology for the Design and Evaluation of Ontologies," Proceedings of IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada (1995)

Hailemariam, L., P. Suresh, V. P. K. Akkisetty, G. Joglekar, V. Hsu, A. Jaine, K. Morris, G. Reklaitis, P. Basu and V. Venkatasubramanian; "Excipient Interaction Prediction: application of the Purdue Ontology for Pharmaceutical Engineering (POPE)," Proceedings of the 18th European Symposium on Computer Aided Process Engineering, 25, 85-90, Amsterdam, The Netherlands (2008)

Horridge, M. and P. F. Patel-Schneider; "OWL 2 Web Ontology Language Manchester Syntax," W3C Working Group Note. http://www.w3.org/TR/owl2-manchester-syntax/ (2012)

Horrocks, I., P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof and M. Dean; "SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission," Available at http://www.w3.org/Submission/SWRL/ (2004)

ISO 15926-2; "ISO-15926:2003 Integration of lifecycle data for process plant including oil and gas production facilities: Part 2 – Data model," International Organization for Standardization, Switzerland (2003)

Lees, F. P.; Loss Prevention in the Process Industries, Elsevier, Oxford, U.K (1996)

Lenoir, E. M. and J. A. Davenport; "A Survey of Vapor Cloud Explosions: Second Update," Process Saf. Prog., **12**, 12-33 (1993)

Marca, D. L. and C. L. McGowan; IDEF0 and SADT: A Modeler's Guide," OpenProcess, Inc., U.S.A (2005)

Martin, R., J. A. Reza and L. W. Anderson; "What is an Explosion? A Case History of an Investigation for the Insurance Industry," *J. Loss Prev. Process Ind.*, **13**, 491-497 (2000)

Morbach, J., A. Wiesner and W. Marquardt; "OntoCAPE—A (re)usable Ontology for Computer-aided Process Engineering," *Comput. Chem. Eng.*, **33**, 1546–1556 (2009)

Motik, B., P. F. Patel-Schneider and B. Parsia; "OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax," W3C Recommendation. http://www.w3.org/TR/owl2-syntax/ (2009)

OSHA; "Combustible Dust in Industry: Preventing and Mitigating the Effects of Fire and Explosion," U. S. Department of Labor, Occupational Safety and Health Administration. http://www.osha.gov/dts/shib/shib073105.html (2005)

Pease, A., I Niles and J Li; "The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications," Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web, Edmonton, Canada (2012)

Priss, U.; "Formal Concept Analysis in Information Science," *Annu. Rev. Information Science and Technology*, **40**, 521-543 (2006)

Shearer, R., B. Motik and I. Horrocks; "HermiT: A Highly-Efficient OWL Reasoner," 5th Int. Workshop on OWL: Experiences and Directions (OWLED 2008 EU), Karlsruhe, Germany (2008)

Shoham, Y.; "Reasoning About Change: Time and Causation from the Standpoint of Artificial Intelligence," MIT Press, Cambridge, U.S.A (1988)

Sirin, E., M. Smith and E. Wallace; “Opening, Closing Worlds - on Integrity Constraints,” OWL: Experiences and Directions 2008, C. Dolbear, A. Ruttenberg, and U. Sattler, eds., OWLED (2008)

Tudorache, T., N. F. Noy, S. W. Tu and M. A. Musen; “Supporting Collaborative Ontology Development in Protégé,” 7th International Semantic Web Conference. Springer, Karlsruhe, Germany, pp. 17-32 (2008)

Wille, R.; “Restructuring lattice theory: an Approach based on Hierarchies of Concepts,” Ordered Sets, I. Rival, eds., Reidel, Dordrecht-Boston, U.S.A, pp. 445–470 (1982)

Yevtushenko, S.; “Computing and Visualizing Concept Lattices,” Ph.D. Thesis, TU Darmstadt, Germany (2004)