20th European Symposium on Computer Aided Process Engineering – ESCAPE20
S. Pierucci and G. Buzzi Ferraris (Editors)

# A Knowledge-based Framework for Incident Management of Pharmaceutical Processes

Jyunpei Sugiura,[a] Hirofumi Kawai,[b] Yukiyasu Shimada,[c] Tetsuo Fuchino,[d] and Rafael Batres[a*]

*[a]Toyohashi University of Technology, Toyohashi 441-8580, Japan*
*[b]Process Systems Engineering Division, Tokyo Institute of Technology*
*[c]National Institute of Occupational Safety and Health, Tokyo 204-0024, Japan*
*[d]Tokyo Institute of Technology, Tokyo 152-8552, Japan*
*[*]Corresponding author: rbp@tut.jp*

## Abstract

Current GMP regulations require pharmaceutical, medical device, and food manufacturers to keep incident records that document incipient faults, near-misses, and incidents which have a potential impact on the quality and safety of their products. This paper presents a framework for storing, maintaining, and retrieving information about such past incidents and their solutions. Extracted information can be used to identify what went wrong and what solutions were effective in order to avoid similar incidents. This paper focuses on one of the components of the framework, namely the identification of association rules.

**Keywords**: FCA, ontologies, CBR, GMP, incident logs, product safety

## 1. Introduction

Despite the emphasis on R&D in the pharmaceutical industries, quality-related deficiencies contribute to more than 25% of the total revenues (Winkle, 2007). Good Manufacturing Practice (GMP) regulations apply to pharmaceutical, medical device, and food manufacturers to ensure that their products are processed reliably, repeatedly, consistently, safely and to a high quality. Current GMP regulations (CGMPs) require manufacturers to investigate incipient faults, near-misses, and incidents that have an impact on the product quality and safety. These data are used to produce incident reports that are maintained with the intent of extracting information that can be used to avoid similar incidents while improving product quality and productivity. Incident reports contain data such as materials and equipment involved in the incident (e.g. process lines, products, batches and raw materials), possible or actual consequences, possible causes, products made prior to and during the event and corrective or preventive actions. Incident reports are written the form of textual natural language descriptions which limit the ability to use past data in an efficient way. Some research has been done on mining maintenance logs of discrete manufacturing plants, but no research has been reported on the mining of incident logs of batch or continuous plants. Specifically, Devaney et al. (2005) discuss a project for mining maintenance logs using text processing, text clustering and case-based reasoning but no specific results are reported. Anand et al (2006) use association rules for mining a subset of incidents stored in the National Response Center incident database. The association rule extraction is performed by exploring all the combinations resulting from different kinds of equipment and 12 chemicals, an approach that works well for small datasets but requires a computational effort that increases exponentially. This computational effort can be alleviated by Formal Concept Analysis which includes a number of algorithms for a more efficient mining (Estacio-Moreno, 2008) (Lakhal and Stumme, 2005).

This paper presents a framework for storing, maintaining and retrieving information about past decisions on incident resolution based on case-based reasoning, ontologies, and formal concept analysis. This paper focuses on one of the components of the framework, namely the identification of association rules.

## 2. Methodology

The core of the methodology is based on case-based reasoning (Fig. 1). In case-based reasoning problems are solved "by using or adapting solutions to old problems" (Riesbeck and Schank, 1989). A case is a representation of the problem and a solution to that problem. In this paper, a case is made up with information contained in an incident report. Incident reports are stored in a case base where incident information follows a predefined structure which includes definitions from domain ontologies. Ontologies define a set of classes and a set of relations between these classes for things such as equipment, processing activities, and causality. When a new problem arises (for example when new incident occurs) the information that is available about this target incident is entered to retrieve similar cases from which the best matching case is selected. Subsequently, domain knowledge is used to complete missing information or adapted by using domain-specific knowledge. The resulting case is also adapted by means of association rules generated by analyzing the case base using a technique called Formal Concept Analysis. The retrieved case and the adapted case are shown to the user who uses this information to take preventive or corrective actions. Subsequently, the user confirms the case with on-site investigation and stored as a new case.

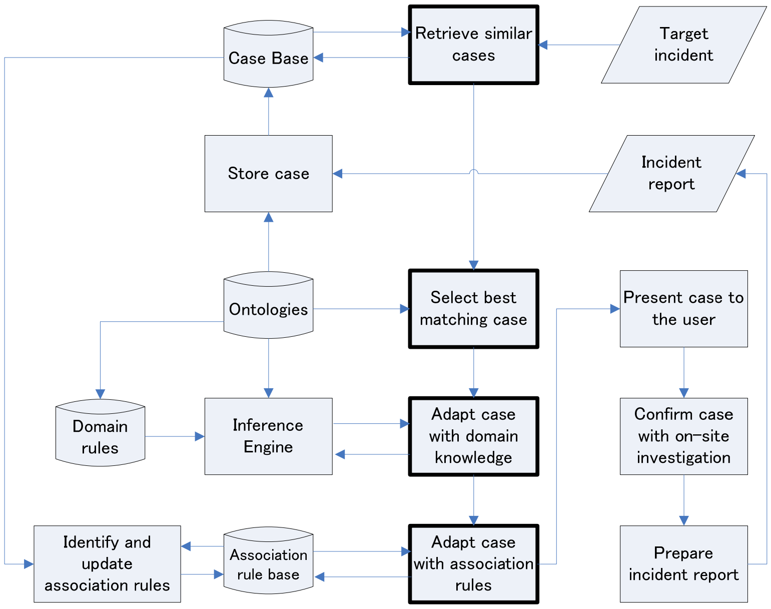

Figure 1. Methodology of the proposed framework

## 3. Ontologies

Ontologies are used in the "Select best matching case" step to find similarities between parts of the previous cases and the target problem. An ontology describes a shared and common understanding of a domain that can be communicated between people and heterogeneous software tools. An ontology is constructed by defining classes of things, their taxonomy, the possible relations between things and axioms for those relations. A class represents a category of things that share a set of properties. The classes in the ontology can be represented as a tuple $\langle C, \leq c \rangle$ where $C$ is a set in which each element is a class with partial order $\leq c$ on $C$. A relation is a function that maps its arguments to a Boolean value of true or false. Examples of relations are *connected_to*, and *part_of*. Class taxonomies are defined with the use of the *subclass* relation. A class is a subclass of another class if every member of the subclass is also a member of the superclass. ISO 15926 describes classes and relations that can be used to represent things such as processing activities, personnel, plant equipment, chemical processes, batch recipes and engineering diagrams (Batres *et al.*, 2007).

## 4. Best case selection

The best case is selected using a similarity function that evaluates the differences between the target problem and a case stored in the case base. A number of different similarity functions for text, quantities, and ontology objects are investigated. A number of well-known similarity functions exist for text and quantities. For similarity between two ontology objects $a \in C_1$ and $b \in C_2$ is measured by the distance between classes $C_1$ and $C_2$ in regards to their lowest common ancestor.

Let $C$ denote the set of all classes in the ontology. Given two classes $C_1$, $C_2 \in C$, let lca($C_1$, $C_2$) be the common ancestor of $C_1$ and $C_2$. We define similarity between $C_1$ and $C_1$ as:

$$\mathrm{sim}(C_1, C_2) = f\left(\mathrm{dist}(C_1, \mathrm{lca}(C_1, C_2)), \mathrm{dist}(C_2, \mathrm{lca}(C_1, C_2))\right)$$

where dist($C_1$, lca($C_1, C_2$)) and dist($C_2$, lca($C_1, C_2$)) are the number of classes + 1 in the shortest path from the class $C_1$ to lca($C_1, C_2$) and from $C_2$ to lca($C_1, C_2$).

## 5. Adaptation with domain knowledge

Domain knowledge can be used to identify and modify parts of the best case that are not applicable to the target problem. If the framework attempts to re-apply a corrective action and discovers that one part of the corrective action is not applicable for the current target problem, the corrective action will modified by finding a different alternative for that part. For example, suppose that the framework finds that by "numbering the trays when more than one sample is analyzed" help to prevent an incident that is caused by "mistakenly switching samples" when carrying out a near-infrared analysis. The solution presented by the framework suggests to "number the cuvettes when more than one sample is analyzed," when carrying out a similar analysis using UV. However, the current target problem uses sampler trays but not cuvettes. In this case, the solution would be an adapted transformation by replacing cuvettes by trays.

## 6. Adaptation with of association rules

The identification of association rules refers to the extraction of hidden relations in the logs that permit the discovery of knowledge that includes the identification of patterns in the records, the prediction of the probability that events will occur, and the identification of strong relations between causes and effects. An association rule is a relation between two sets of items *A*, *B*, that indicates that cases involving *A* tend also to involve *B*.

Identification of association rules is done by processing the incident data stored in the case base using Formal Concept Analysis (FCA). FCA is an analysis technique for knowledge processing based on applied lattice and order theory (Wille, 1982). FCA assumes that data is represented in as a tuple $K := \langle O, A, Y \rangle$ where $O$ is a set of objects, $A$ is a set of attributes, $Y$ a set of binary relations $Y \subseteq O \times A$ containing all pairs $\langle o, a \rangle \in Y$ such that the object $o$ has the attribute $a$ such as in as in (incident1, caused by step started too late). The initial step in FCA is to find all pairs $\langle O_i, A_i \rangle$ that satisfy $O_i \subseteq O$ , $A_i \subseteq A$ , $O' = A_i$ and $A' = O_i$ where $A'$ is the set of attributes common to all objects in $O_i$, and $O'$ represents the set that has all attributes in $A_i$. Each pair $\langle O_i, A_i \rangle$ is called a formal concept. $O_i$ and $A_i$ are respectively the *extent* and the *intent* of the formal concept. The hidden relations become apparent by analyzing the so-called concept lattice. A concept lattice is a partially ordered set in which a $\langle O_i, A_i \rangle \subseteq \langle O_j, A_j \rangle$ iff $O_i \le O_j$ . Several algorithms for lattice-construction are available.

Typically, the set $K := \langle O, A, Y \rangle$ is represented by a cross table. In this paper, $O$ represents the set of incidents and $A$ denotes the set of attributes that include specific causes and consequences, equipment categories, product names, raw materials used, products made prior to the event, products made during the event, and impacts to product.

Preliminary observations indicate that it is possible to identify mutually exclusive classes of events, direct causality (when the intent includes only one cause and only one consequence), and potential multiple causality (when the intent includes two or more causes and one consequence).

In FCA, association rules are expressed as $A_1 \Rightarrow A_2$ where $A_1$ and $A_2$ are sets of attributes that are disjoint ( $A_1 \cap A_2 = \varnothing$ ). Unlike the domain rules (explained in section 5) association rules are probabilistic in nature. The level of support of $A_1 \Rightarrow A_2$ is defined as the proportion of cases that include all the attributes in $A_1$ and $A_2$ . The level of confidence is the proportion of cases that include all the attributes in $A_2$ within the subset of those cases that include all the attributes in $A_1$ . Here we define $f(O_i) := \{ a \in A \mid \langle o, a \rangle \in Y \, \forall o \in O_i \}$ and $g(A_i) := \{ o \in O \mid \langle o, a \rangle \in Y \, \forall a \in A_i \}$.

Formally, the level of support of rule $A_1 \Rightarrow A_2$ is defined as $supp(\ A_1 \Rightarrow A_2\ ) = \dfrac{|g(A_1 \cup A_2)|}{|O|}$. Based on this definition, the confidence of rule $A_1 \Rightarrow A_2$ is defined as $conf(\ A_1 \Rightarrow A_2\ ) = \dfrac{supp(\ A_1 \cup A_2\ )}{supp(\ A_1\ )}$.

*6.1. Example*

Fig. 2 shows an illustrative example of past cases D1,…,D11 organized as a context table. Incidents are reported to occur when carrying out recipes R1 and R2 with cases C1,…,C5 and consequences E1,…,E3. It is immediately apparent that cases labeled D1, D2, D3 constitute a formal concept because they share exactly the same attributes {R1, C1, C2, E1} not shared by any other object. Similarly, cases labeled D11, D12 constitute another formal concept with attributes {R1, C2, E1}. From the lattice it can be seen that {R1, C1, C2, E1} $\subseteq$ {R1, C2, E1}.

| | R1 | R2 | C1 | C2 | C3 | C4 | C5 | E1 | E2 | E3 |
|---|---|---|---|---|---|---|---|---|---|---|
| Obj1 | × | | × | × | | | | × | | |
| Obj2 | × | | × | × | | | | × | | |
| Obj3 | × | | × | × | | | | × | | |
| Obj4 | | × | | × | | | | × | | |
| Obj5 | | × | | | | × | × | | × | |
| Obj6 | | × | | | | × | × | | × | |
| Obj7 | | × | | | | × | × | | × | |
| Obj8 | | × | | | × | | | × | × | |
| Obj9 | × | × | | | | | × | | | × |
| Obj10 | × | × | | | | | × | | | × |
| Obj11 | × | | | × | | | | × | | |
| Obj12 | × | | | × | | | | × | | |

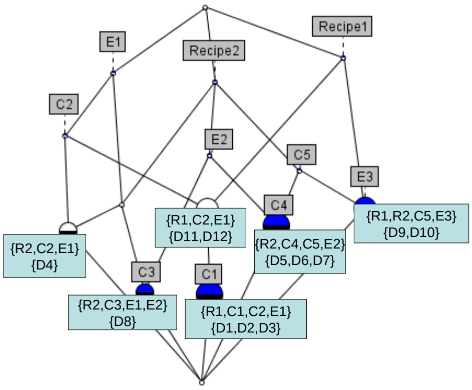

Figure 2. A context table incident cases and its corresponding lattice

There are three cases related to recipe R1 in which C1, C2 are the cause of E1. From a simple look at these individual cases, three causality alternatives are possible: {R1, C1, E1}, {R1, C2, E1} or {R1, C1, C2, E1}. However, by taking into account other cases, FCA concludes that C1 alone cannot be considered a cause of E1. Note that {R1, C1, E1} is not included in the lattice, while keeping {R1, C2, E1} and {R1, C1, C2, E1}. This observation is also supported by the association rules in Fig. 3.

Notice that rules 3, 5 support the causality of {R1, C2, E1} and rule 8 supports {R1, C1, C2, E1}. Specifically, rule 8 suggests that C1 and C2 are probably interrelated.

*6.1.1. Direct causality*

Rule 1 indicates that there is a strong support to conclude that independently of which recipe is carried out C2 causes E1.

*6.1.2. Relations between recipes and events*

From rule 6 and from the lattice it can be seen that E2 occurs only when the production is carried out with recipe R2.

| | |
|---|---|
| 1 < 6 > C2 =[100%]=> < 6 > E1; | 8 < 3 > C1 =[100%]=> < 3 > R1 C2 E1; |
| 2 < 7 > E1 =[86%]=> < 6 > C2; | 9 < 3 > C4 =[100%]=> < 3 > R2 C5 E2; |
| 3 < 5 > R1 E1 =[100%]=> < 5 > C2; | 10 < 2 > R1 R2 =[100%]=> < 2 > C5 E3; |
| 4 < 5 > C5 =[100%]=> < 5 > R2; | 11 < 2 > E3 =[100%]=> < 2 > R1 R2 C5; |
| 5 < 6 > C2 E1 =[83%]=> < 5 > R1; | 12 < 1 > R2 E1 E2 =[100%]=> < 1 > C3; |
| 6 < 4 > E2 =[100%]=> < 4 > R2; | 13 < 1 > C3 =[100%]=> < 1 > R2 E1 E2; |
| 7 < 3 > R2 C5 E2 =[100%]=> < 3 > C4; | |

Figure 3. Association rules

### *6.1.3. Mutually exclusive events*

From the lattice it can be noticed that consequences E1 and E2 can occur simultaneously. However, rule 12 indicates that there is a small probability that E1 and E2 are related. Contrasting with this, neither the lattice nor the association rules indicates that E3 can take place along with E1 and E2. The latter situation is an example of exclusive events which can be used to adapt a case.

## 7. Conclusions

This paper presents a framework for storing, maintaining and retrieving information about past corrective actions of incidents by combining case-based reasoning, ontologies, and formal concept analysis. Preliminary observations indicate that it is possible to identify mutually exclusive classes of events, direct causality (when the intent includes only one cause and only one consequence), potential multiple causality (when the intent includes two or more causes and one consequence) and the relations between events and other entities. However, much work is needed in several components of the framework.

## 8. 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).

## References

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

M. Devaney, A. Ram, H. Qiu, J. Lee. Preventing Failures by Mining Maintenance Logs with Case-based reasoning. 59th Meeting of the Society for Machinery Failure Prevention Technology (2005)

A. Estacio-Moreno, Y. Toussaint, C. Bousquet. Mining for Adverse Drug Events with Formal Analysis. 21st Int. Congress of the European Federation for Medical Informatics (2008)

L. Lakhal and G. Stumme. Efficient Mining of Association Rules Based on Formal Concept Analysis. In B. Ganter, G. Stumme, R. Wille (Eds.) Formal Concept Analysis, Foundations and Applications. Lecture Notes in Computer Science 3626 Springer 2005

J.F. van Leeuwen, M. J. Nauta, D. de Kaste, Y.M.C.F. Odekerken-Rombouts, M.T. ldenhof, M.J. Vredenbregt and D.M. Barends. Risk analysisnext term by previous termFMEA as an element of analytical validationnext term. Journal of Pharmaceutical and Biomedical Analysis, 50(5), (2009) 1085-1087

C. Riesbeck, and R. Schank. Inside Case-based Reasoning. Northvale, NJ, Erlbaum (1989)

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

H. N. Winkle. Implementing Quality by Design. Evolution of the Global Regulatory Environment: A Practical Approach to Change, http://www.fda.gov/downloads/AboutFDA/CentersOffices/CDER/ucm103453.pdf (2007)