# 國 立 交 通 大 學

## 資訊管理研究所

## 博士論文

# 問題解法之知識支援

# Knowledge Support for Problem-solving

研 究 生：柯 志 坤

指導教授：劉 敦 仁 博士

中 華 民 國 九 十 五 年 十 月

# 國 立 交 通 大 學

## 資訊管理研究所

## 博 士 論 文

問題解決之知識支援

Knowledge Support for Problem-solving


研 究 生：柯志坤

研究指導委員會：王朝煌 博士<br>
李瑞庭 博士<br>
羅濟群 博士<br>
楊 千 博士

指導教授：劉敦仁 博士



中 華 民 國 九 十 五 年 十 月

# 問題解決之知識支援
# Knowledge Support for Problem-solving

研 究 生：柯志坤　　　　Student：Chih-Kun Ke

指導教授：劉敦仁　　　　Advisor：Dr. Duen-Ren Liu

國 立 交 通 大 學
資 訊 管 理 研 究 所
博 士 論 文

A Dissertation
Submitted to Institute of Information Management
College of Management
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy in Information Management
October 2006

Hsinchu, Taiwan, the Republic of China

中華民國 九十五 年 十 月

# Knowledge Support for Problem-solving

Student: Chih-Kun Ke Advisor: Dr. Durn-Ren Liu

Institute of Information Management,

National Chiao Tung University

## Abstract

Problem-solving is an important process that enables corporations to create competitive business advantages. Traditionally, Case-Based Reasoning (CBR) techniques have been widely used to help workers solve problems. However, conventional approaches focus on identifying similar problems without exploring the information needs of workers and relevant context of situation during the problem-solving process. Such processes are usually knowledge intensive tasks; therefore, workers need effective knowledge support that gives them the information necessary to identify the causes of a problem and enables them to take appropriate action to resolve the situation. In this work, we propose a mining-based knowledge support system for problem-solving. Based on CBR and data mining techniques, in addition to adopting CBR techniques to identify similar situations and the action taken to solve them, the proposed system employs text mining (*Automatic Indexing*) techniques to extract the key concepts of situations and actions. These concepts form profiles that model workers' information needs when handling problems. Effective knowledge support can thus be facilitated by providing workers with situation/action-relevant information based on the profiles. Moreover, association rule mining is used to discover hidden knowledge patterns from historical problem-solving logs. The discovered patterns identify frequent associations between situations and actions, and can therefore provide decision-making knowledge, i.e., appropriate actions for handling specific situations. We develop a prototype system to demonstrate the effectiveness of providing situation/action relevant information and decision-making knowledge to help workers solve problems. Furthermore, based on CBR, data mining, and rule inference techniques, the context-based situation identified by CBR techniques provides effective context-based knowledge documents according to the context-based profile. The hidden knowledge patterns are discovered to identify inferred associations between situation features and actions, and can therefore provide context-based relevant knowledge. A prototype system is developed to demonstrate the effectiveness of providing inferred knowledge.

**Keywords**: Problem-Solving Process, Case-based Reasoning, Data Mining, Knowledge Pattern, Context, Rule Inference, Knowledge Support

# Acknowledgement

誠心地感謝口試委員王朝煌教授、李瑞庭教授、羅濟群教授以及楊千教授，謝謝您們在百忙之中，於論文口試期間，仍細審論文並給予指正與建議，使得本論文能夠更臻至完善，在此深表感激。另外，感謝論文指導老師劉敦仁教授，這六年多的指導與照顧。

花了六年多完成了這博士學程，現實的壓力讓我感覺不到任何的驕傲與光榮，不知道拿到畢業證書的那一刻，眼淚會不會從眼角流下。現在的心境充滿的是感謝還是惆悵，實在很難分的清楚，只希望往後的日子裡，社會的經歷能告訴我在交大受的訓練是值得感謝的。另外，想要感謝的親朋好友們，你們名字已經烙印在我的心中，不用在這邊強調，就讓我們將路走的更長遠吧。

志坤 2006/10

# Contents

# List of Figures

# List of Tables

# Chapter 1. Introduction

## 1.1. Motivation

Problem-solving is an important process that enables corporations to create competitive advantages, especially in the manufacturing industry. Case-based reasoning (CBR) techniques (Chang et al., 1996; Kohno et al., 1997; Park et al., 1998; Yang et al. 2004) have been widely used to help workers solve problems. For example, based on these techniques, a decision support system was developed to facilitate problem-solving in a complex production process (Park et al., 1998). CBR techniques have also been used to implement a self-improvement helpdesk service system (Chang et al., 1996), and integrated with the ART-Kohonen Neural Network (ART-KNN) to enhance fault diagnosis in electric motors (Yang et al., 2004).

Conventional CBR approaches focus on identifying similar problems without exploring the information needs of workers and relevant context of situations during problem-solving tasks. Problem-solving is a complex process that includes a series of uncertain situations and operational actions. Moreover, it is usually knowledge intensive and workers need to access relevant information in order to identify the causes of a situation and take appropriate action to solve it. Situation features are usually occurred according to the context characteristics of problem. Due to the uncertain features of situations, several causes and possible solutions may exist for a specific situation. For example, in a production process, a significant decline in performance may be due to poor materials, inexperienced workers, or faulty machinery. Thus, possible solutions would include replacing the poor materials, retraining the workers, or repairing the faulty machinery. The causes and possible solutions are usually hidden in relevant data resources and difficult to extract. In such uncertain environments, situation features collected by system are usually partial or incomplete. Workers need to use knowledge gathered and inferred from relevant context information and previous problem-solving experience to clarify the causes and take appropriate action effectively. Thus, identifying similar cases through CBR is not sufficient to solve problems. An effective knowledge support system is essential so that workers have the information necessary to identify the causes of a problem and take appropriate action to solve it.

## 1.2. Goals

According to the motivation, this work lists major goals as follows:

- Analyze collected attributes of situation/action of problem-solving;
- Based on CBR and data mining techniques, design a system framework of knowledge support for problem-solving;
- Identify similar situations/actions by CBR;
- Discovery of situation/action profile and knowledge patterns;
- Construct a knowledge support network for knowledge recommendation;
- Implement a prototype system to demonstrate the effectiveness of proposed framework;
- Analyze collected attributes and features of situation for problem-solving;
- Based on CBR, data mining, and rule inference techniques, design a system framework of context-based knowledge support for problem-solving;
- Identify similar context-based situations by CBR;
- Discovery of context-based situation profile and relevant knowledge (e.g., knowledge patterns and relevant knowledge documents);
- Implement a prototype system to demonstrate the effectiveness of proposed framework.

## 1.3. Contributions

In this work, we propose a mining-based knowledge support system for problem solving. Besides adopting CBR to identify similar situations and the action taken to solve them, we adopt text mining (*Automatic Indexing*) and rule inference techniques to compensate for the shortcomings of CBR technique. For specific situations or actions, their situation/action attributes, features, context characteristics and relevant information (documents) accessed by workers is recorded in a problem-solving log. Historical codified knowledge (textual documents), i.e., experience and know-how extracted from previous problem-solving logs, can provide valuable knowledge for solving the current problem.

The proposed system employs Information Retrieval (*Automatic Indexing*) techniques to extract the key concepts of relevant information necessary to handle a specific situation or action. The extracted key concepts form a situation/action profile that models the information needs of workers for a specific problem-solving task. The system can then uses the

***situation/action profile*** to gather existing and new relevant knowledge documents for specific situation/action. We employ association rule mining methods to discover decision-making knowledge rules about frequently adopted actions taken to handle specific situations. These rules are generated as knowledge support to help workers take the appropriate action to solve a specific situation. Furthermore, the problem-solving process includes a series of uncertain situations and operational actions, and preceding situations or actions may trigger subsequent problem situations. Therefore, workers need to gather such triggering information (chain reactions) to determine appropriate action. For example, if an unstable system causes production to decline, the solution may be to reboot the system. However, this may result in breakage of materials, which would increase production costs. The proposed approach applies sequential pattern mining methods to discover dependency knowledge which represents frequent chain-reactions. The knowledge helps workers make appropriate action plans. The discovered profiles and knowledge rules are used to construct a knowledge support network, which provides workers with relevant situation/action information, as well as decision-making and dependency knowledge. A prototype system is developed to demonstrate the effectiveness of the knowledge support network.

Moreover, we adapt system framework to provide context-based knowledge support for problem-solving. The adapted system employs constraint-based association rule mining methods to discover context-based inference rules from the problem-solving log. Context-based inference rules identify inferred associations between situation features and relevant context characteristics. Based on the discovered context-based inference rules, the system infers more situation features to assist CBR in situation identification. The proposed system employs Information Retrieval (*Automatic Indexing*) techniques to extract the key concepts of relevant information necessary to handle a specific situation. The extracted key concepts form a context-based situation profile that models the information needs of workers for handling problem situation in certain context. The system can then uses the ***context-based situation profile*** to gather existing and new relevant knowledge documents for specific situation according to the context information. Furthermore, the adapted system continually infers situation features to form the context-based knowledge patterns which provide workers with relevant inferred knowledge (inferred situation features and relevant context-based inference rules), as well as context-based decision-making and dependency knowledge.

## 1.4. Organization

The remainder of this work is organized as follows. Chapter 2 reviews related works on knowledge discovery and problem-solving. Chapter 3 introduces the knowledge requirements of knowledge support for problem-solving. Chapter 4 describes the knowledge support based on CBR and data mining techniques, including knowledge support framework for problem-solving, discovery of problem-solving knowledge, knowledge support for problem-solving, and a prototype system implementation. Chapter 5 illustrates the knowledge support based on CBR, data mining, and rule inference techniques, including context-based knowledge support framework for problem-solving, discovery of context-based problem-solving knowledge, the prototype system, discussions, and comparisons. Finally, we summarize this work and describe the future works in Chapter 6.

# Chapter 2. Related Work

The related literature covers knowledge management, problem-solving, case-based reasoning, information retrieval, data mining techniques, and context-awareness.

## 2.1. Knowledge management and knowledge retrieval

AI techniques have advanced knowledge management, including knowledge acquisition, knowledge repositories, knowledge discovery, and knowledge distribution (Liebowitz, 2001). Knowledge acquisition captures tacit and explicit knowledge from domain experts (Kohno et al, 1997; Klemettinen et al., 1997), while knowledge repositories formalize the outcomes of knowledge acquisition and integrate knowledge in distributed corporate environments (Georgalas, 1999). Taxonomy and mapping mechanisms are used to represent relevant knowledge and construct a framework for building a knowledge repository (Chakrabarti et al., 1997). Knowledge discovery and mining approaches explore relationships and trends in the knowledge repositories to create new knowledge. In addition, heuristic mechanisms, such as proactive knowledge delivery and context-aware knowledge retrieval, are used to enhance knowledge distribution (Abecker et al., 2000).

A repository of structured, explicit knowledge, especially in document form, is a codified strategy for managing knowledge (Davenport & Prusak, 1998; Gray, 2001). However, with the growing amount of information in organization memories, knowledge management systems (KMS) face the challenge of helping users find pertinent information. Accordingly, knowledge retrieval is considered a core component in accessing information in knowledge repositories (Kwan & Balasubramanian, 2003; Fenstermacher, 2002). Translating users' information needs into queries is not easy. Most systems use Information Retrieval (IR) techniques to access organizational codified knowledge. The use of Information Filtering (IF) with a profiling method to model users' information needs is an effective approach that proactively delivers relevant information to users. The technique has been widely used in the areas of Information Retrieval and Recommender Systems (Herlocker & Konstan, 2001; Middleton et al., 2004; Pazzani & Billsus, 1997). The profiling approach has also been adopted by some KMS' to enhance knowledge retrieval (Abecker et al., 2000; Agostini et al., 2003; Davies et al., 2003), whereby information is delivered to task-based business environments to support proactive delivery of task-relevant knowledge (Abecker et al., 2000; Fenstermacher, 2002; Liu 2005).

## 2.2. Problem solving and case-based reasoning

Problem solving is the thought process that resolves various difficulties and obstacles spread in the gap between the current problem and its desired solution. (Heh, 1999). Past experience or knowledge, routine problem-solving procedures, and previous decisions can be used to enhance problem-solving. Liao (2002) investigates the types of knowledge used for problem-solving and suggests the circulation of knowledge to avoid knowledge inertia. Although a knowledge-based architecture that incorporates case-based, rule-based, and heuristic-based approaches is proposed for managing problem-solving knowledge and dealing with knowledge inertia, the details of the system are not presented.

Various approaches that integrate AI techniques have been proposed to support problem solving. Case-based reasoning (CBR), which has been widely used to help workers solve problems, is the process of solving a given problem based on the knowledge gained from solving previous similar problems (Allen et al., 2002). Most CBR systems include the following steps: case representation and storage, precedent matching and retrieval, adaptation of the retrieved solution, validation of the solution, and case-base updating to include the information gained from solving the new problem. The CBR approach was used to implement a self-improvement helpdesk service system (Chang et al., 1996), and a CBR-based decision support system was developed for problem-solving in a complex production process (Park et al., 1998). More recently, Yang et al (2004) proposed integrating the CBR approach with ART-Kohonen neural networks (ART-KNN) to enhance fault diagnosis in electric motors. Moreover, RBCShell was introduced as a tool for constructing knowledge-based systems with CBR (Guardati, 1998), whereby previously solved problems are stored in the case memory to support problem-solving in new cases.

Existing studies focus on using case-based reasoning to identify similar previous cases and derive a solution for a new case from previous problem solutions. In a complex production process, problem-solving is usually knowledge intensive and requires effective knowledge support to provide workers with the necessary information to identify the causes of situations and taking appropriate action to solve them. However, identifying similar cases among previous problem cases is not sufficient to satisfy workers' information needs for solving a new problem. The required knowledge is usually hidden in various codified knowledge documents that must be proactively delivered to workers. The CBR approach does not provide such problem-relevant documents for knowledge-intensive problem solving.

## 2.3. Information retrieval in a vector space model

The key contents of a codified knowledge item (document) can be represented as a term vector (i.e., a feature vector of weighted terms) in n-dimensional space, using a term weighting approach that considers the term frequency, inverse document frequency, and normalization factors (Salton et al., 1988). The *term transformation* steps, including case folding, stemming, and stop word removal, are performed during text pre-processing (Salton et al., 1971; Poter, 1980; Witten et al., 1999). Then, *term weighting* is applied to extract the most discriminating terms (Baeza-Yates et al., 1999). Let d be a codified knowledge item (document), and let $\vec{d} = \langle w(k_1, d), w(k_2, d), \ldots, w(k_n, d) \rangle$ be the term vector of $d$, where $w(k_i, d)$ is the weight of a term $k_i$ that occurs in $d$. Note that the weight of a term represents its degree of importance in representing the document (codified knowledge). The well-known *tf-idf* approach, which is often used for *term (keyword) weighting* (Poter, 1980), assumes that terms with higher frequency in a document and lower frequency in other documents are better discriminators for representing the document. Let the term frequency $tf(k_i, d)$ be the occurrence frequency of term $k_i$ in $d$, and let the document frequency $df(k_i)$ represent the number of documents that contain $k_i$. The importance of $k_i$ is proportional to the term frequency and inversely proportional to the document frequency, which is expressed as Equation 1:

$$w(k_i, d) = \frac{1}{\sqrt{\sum_i \left(tf(k_i, d) \times \log(N/df(k_i)+1)\right)^2}} tf(k_i, d) \times (\log \frac{N}{df(k_i)} + 1), \quad (1)$$

where $N$ is the total the number of documents. Note that the denominator on the right-hand side of the equation is a normalization factor that normalizes the weight of a term.

***Similarity measure***: The cosine formula is widely used to measure the degree of similarity between two items, $x$ and $y$, by computing the cosine of the angle between their corresponding term vectors $\vec{x}$ and $\vec{y}$, which is given by Equation 2. The degree of similarity is higher if the cosine similarity is close to 1.

$$sim(x, y) = \text{cosine}(\vec{x}, \vec{y}) = \frac{\vec{x} \bullet \vec{y}}{|\vec{x}||\vec{y}|} \quad (2)$$

## 2.4. Data mining

Data mining, which has become an increasingly important research area, involves several tasks, including association rule mining, sequential pattern mining, clustering, classification, and prediction (Chen et al.,1996; Han & Kamber, 2000). We adopt association rule mining and sequential pattern mining to extract knowledge patterns from previous problem-solving instances.

***Association rules mining.*** Association rule mining tries to find an association between two sets of products in a transaction database. Agrawal et al. (1993) formalized the problem of finding association rules as follows. Let $I$ be a set of product items and $D$ be a set of transactions, each of which includes a set of products that are purchased together. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \Phi$. $X$ is the antecedent (body) and $Y$ is the consequent (head) of the rule. Two measures, support and confidence, are used to indicate the quality of an association rule. The support of a rule is the percentage of transactions that contain both $X$ and $Y$, whereas the confidence is the fraction of transactions containing $X$ that also contain $Y$.

***Sequential pattern mining.*** The input data is a set of sequences, called data-sequences. A data-sequence is a list of transactions, each of which is a set of literals, called items. Typically, a transaction-time is associated with each transaction. A sequential pattern also consists of a list of sets of items. Sequential pattern mining finds all sequential patterns from a time-based transaction database (Agrawal & Srikant, 1995; Srikant & Agrawal, 1996).

The support of an association rule or sequential pattern indicates how frequently the rule applies to the data. A high level of support corresponds to a strong correlation between the product items. The *Apriori* algorithm (Agrawal et al, 1993; 1994) is typically used to find association rules by discovering frequent itemsets (sets of items). An itemset is considered to be frequent if its support exceeds a user-specified minimum support. Association rules or sequential patterns that meet a user-specified minimum confidence can be generated from the frequent itemsets.

## 2.5. Context-awareness

According to the definitions of Schilit and Theimer (1994), *context* is the location of user, the identities of people and objects that are nearby the user, and the status of devices the user interact with. They considered that *context-awareness* is adapted to the software execution environment involving with relevant context changing. Dey et al. (2001) defined the *context* as any information that can characterize the situation of an entity, where the entity can be a user, place, service, and service relevant objects, etc. The context is categorized into location, identity, activity, and time types. A *context-aware* system uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task. Ryan et al. (1997) used "environment" to replace "activity" in the context categorization. They use context types to characterize the situation of a particular entity, and provide the information of who, what, when, and where of a particular entity. This work considers the context as any information that can characterize the status of an entity. An entity may be the staff, location, time, or object considered relevant to the interaction between the staff and problem-solving process, including the staff, resolving service provider, components which support resolving service in a problem-solving environment.

## 2.6. Rule inference with certainty factor

Shortliffe et al. (1975) has proposed the method of *Certainty Factor* (*CF*) value to derive the certainty degree during the inference, as defined in Equation 3.

$$CF(X \rightarrow Y) = \begin{cases} CF(X \rightarrow Y) = \dfrac{(Conf(X \rightarrow Y)) - S(Y)}{1 - S(Y)}, & if \quad Conf(X \rightarrow Y) > S(Y) \\ CF(X \rightarrow Y) = \dfrac{(Conf(X \rightarrow Y)) - S(Y)}{S(Y)}, & if \quad Conf(X \rightarrow Y) < S(Y) \\ 0 \quad , & otherwise \end{cases} \tag{3}$$

where the *CF* value is the certainty degree from -1 to 1; value "1" denotes complete certainty; value "-1" denotes complete uncertainty. In this work, *X* denotes the preceding set; *Y* is the set that we want to infer its certainty degree. $CF(X \rightarrow Y)$ is the *CF* value of rule $X \rightarrow Y$. $S(Y)$ is the support of *Y*. $Conf(X \rightarrow Y)$ is the confidence of rule $X \rightarrow Y$. Based on the *CF* value of items and inference rules, the inference process follows the rules defined in Equation 4.

$$\begin{aligned} &CF(X_i \wedge X_j) = MIN(CF(X_i), CF(X_j)) \\ &CF(X_i \vee X_j) = MAX(CF(X_i), CF(X_j)) \\ &CF(B \mid \{IF \quad A \quad THEN \quad B\}) = CF(A \rightarrow B) \times MAX(0, CF(A)) \\ &CF(B) = MAX(CF(B \mid \{IF \quad A \quad THEN \quad B\}), CF(B \mid \{IF \quad C \quad THEN \quad B\})) \end{aligned} \tag{4}$$

# Chapter 3. Knowledge Requirements of Knowledge Support for Problem-solving

In this chapter, we describe the knowledge requirements of knowledge support for problem-solving, including the concepts of the problem-solving process, the knowledge requirements for problem-solving, and the context-based knowledge requirements for problem-solving. A wafer manufacturing process in a semiconductor foundry is used to illustrate the proposed approach. The process comprises the following steps: crystal growing, wafer cutting, edge rounding, lapping, etching, polishing, cleaning, final inspection, packaging and shipping. The wafer cleaning step mainly uses DI (de-ionized; ultra-pure) water to remove debris left over from the mounting wax and/or polishing agent. A stable water supply system to deliver ultra-pure water for wafer cleaning is therefore vital in semiconductor manufacturing.

## 3.1. The problem-solving process

In business enterprises, especially the manufacturing industry, various problem situations may occur during the production process; for example, poor production performance, system overload, and low machine utilization. A situation denotes an evaluation point to determine the status (i.e., desirable or undesirable) of a production process. A problem may occur if there is a discrepancy between the actual situation and the desired one. For example, when the current production output is below the desired level, the production line may have some problems. Thus, a problem-solving process is often initiated to achieve the desired situation. In the process, workers take several problem-solving steps to determine what action needs to be taken to resolve the situation. Such action involves both human wisdom and enterprise knowledge. Workers may observe a problem situation, collect relevant information from the enterprise knowledge repository, explore possible causes, and identify operational conditions in order to decide appropriate action. Moreover, a problem-solving process generally consists of levels of progressive sub-problem solving, which form different stages of the process. Such stage-wise problem-solving reduces the complexity of a problem and solves it more effectively. The stages of problem-solving in a production process are usually pre-determined by experienced workers or experts according to the characteristics of the process and their experience in solving previous problems.

## 3.2. Knowledge requirements for problem-solving

***Situation and action relevant knowledge***. In a specific stage of problem-solving, a worker can access relevant documents associated with the problem situation to find the causes. For example, for the situation "crash of the water supply system", the diagnostic documents contain information about the temperature, pressure, and electric power, which may provide clues to possible causes. The expert-reports indicate that the temperature and pressure features could be the key reasons for the system's failure. The experiment-reports show that high pressure may cause an increase in temperature, which would make the system unstable and result in a crash. The know-how hidden in relevant documents can help workers discover the causes of problem situations. These relevant documents are defined as situation relevant knowledge.

After determining the cause of a problem situation, workers must decide what action to take. They do this by accessing documents related to the cause in order to identify the normal operational-conditions of the production system, and choose an appropriate course of action. Continuing with the example of the water system crash, if the cause is an anomalous temperature level, a safe temperature range is required to stabilize the system. The system's operational manual defines the normal pressure and temperature ranges. For example, when the system's output pressure is one degree of atmospheric pressure, its temperature range is 30 to 32 °C. In addition, the standard operating procedures specify the system's tuning rules: the system temperature increases 4 °C per degree of atmospheric pressure. The experiment-reports indicate a reasonable temperature range of a stable system, where, for example, 55°C is the upper limit of the range. Such relevant operational know-how is hidden in enterprise documents that must be discovered to help workers take appropriate action, i.e., tune the output pressure and temperature to keep the system stable. These documents are defined as action relevant knowledge.

***Decision-making and dependency knowledge.*** Knowing what action to take to solve problem situations is defined as decision-making knowledge, which can be discovered from previous problem-solving logs. Decision-making knowledge is expressed as association rules that represent the association of frequently adopted actions for handling specific situations. These knowledge rules are generated as knowledge support to help workers take appropriate action in handling situations. Moreover, in stage-wise problem-solving, a situation/action may trigger/affect a situation/action in a later stage. Fig. 1 illustrates the three stages of problem-solving on a production line, namely, engineering improvement, quality improvement, and maintenance management.

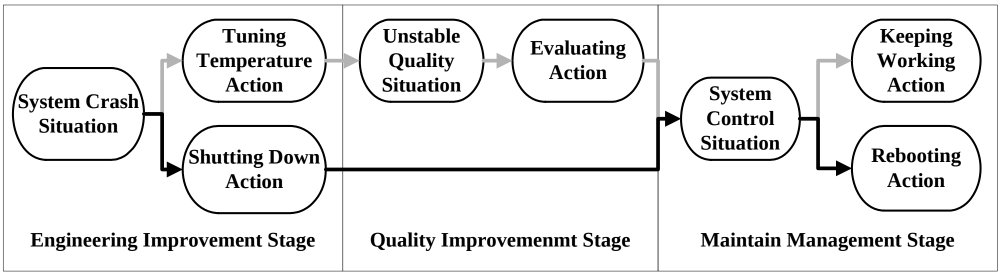


Fig. 1: A problem-solving process for a production line

In the first stage, tuning the system's temperature and shutting down the system are two appropriate ways to resolve a system crash. The shutting down action may trigger a system control situation, which requires rebooting action in the maintenance management stage. Moreover, the tuning action may cause the situation of unstable quality in the quality improvement stage. Such cause-effect relationships (chain reactions) across different stages are called dependency knowledge, which helps workers make appropriate action plans across problem-solving stages. Note that decision-making knowledge represents the intra-relationships between the situations and actions within a stage, while dependency knowledge denotes the inter-relationships between the situations and actions across different stages.

## 3.3. Context-based knowledge requirements for problem-solving

***Context-based inference.*** For a given problem, a situation may occur with various features according to the context at that time. Because situation features collected by system are usually partial or incomplete, a worker can not easily identify current situation. Accordingly, inferring more situation features according to the context characteristics is important in situation identification. For example, the water supply system in a production line provides pure water for wafer cleaning. When the system gets the situation feature "Production-quality low", the causes may be so many that a worker can not easily identify the situation. Situation feature "Parameters of water supply quantity service incorrect" is inferred from the situation feature "Production quality low" and context characteristic "Pressure of water unstable". The context characteristic "Pressure of water unstable" and inferred situation features "Parameters of water supply quantity service incorrect" provide CBR with more clues to identify current situation as "Water supply abnormal issue".

***Context-based situation profile.*** For specific situation, Information Retrieval (Automatic Indexing) techniques are used to extract key terms from situation relevant documents. The extracted key terms form a profile to represent the information needs of workers for handling the situation. Moreover, the profile can be generated according to the context of the situation and is regarded as a context-based situation profile. According to certain context, the key terms recorded in a context-based situation profile are used to locate the relevant documents. The relevant documents are recommended as knowledge support to help workers take appropriate action for handling the situation in certain context.

***Context-based decision-making and dependency knowledge.*** Knowing what action to take according to problem situation features is defined as context-based decision-making knowledge, which can be discovered from the problem-solving logs. The context-based decision-making knowledge patterns indicate the inferred associations of actions and situation features in certain context of the problem-solving process. These context-based knowledge patterns are generated as knowledge support to help workers take appropriate action in handling situations. Moreover, in stage-wise problem-solving, a situation/action may trigger/affect a situation/action in a later stage. Context-based dependency knowledge indicates the inferred relationships between situation/action features in current stage and situations/actions across different stages of the whole problem-solving process context. Context-based dependency knowledge helps workers make appropriate action plans across problem-solving stages.

# Chapter 4. Knowledge Support based on Case-based Reasoning and Data Mining

In this chapter, we describe the knowledge support based on CBR and data mining techniques, including the proposed system framework, discovery of problem-solving knowledge, knowledge support for problem-solving, and a prototype system implementation.

## 4.1. Knowledge support framework for problem-solving

The proposed knowledge support framework for problem-solving, shown in Fig. 2, employs mining techniques to discover needed knowledge. The system framework comprises a problem-solving process, knowledge discovery, and knowledge recommendation modules.

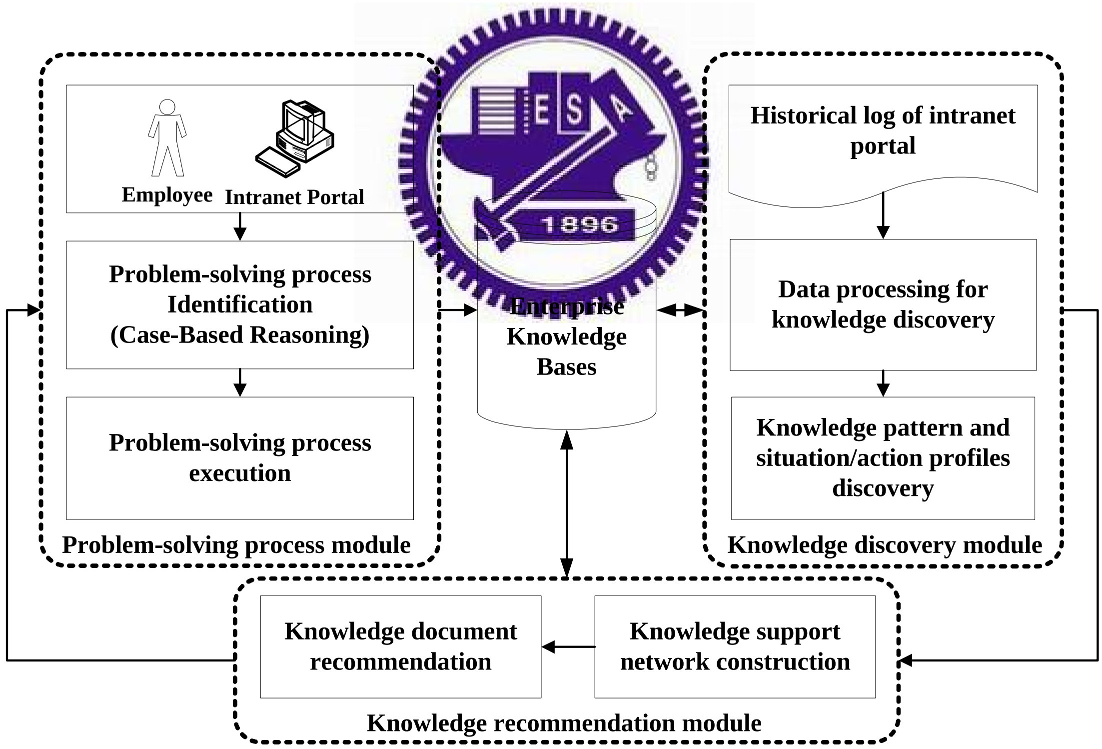


Fig. 2: Knowledge support framework for problem-solving

The proposed framework records the problem-solving steps, including the situations and actions as well as the corresponding knowledge documents accessed in the historical log. The knowledge discovery module employs mining technology to extract hidden knowledge from the historical problem solving log. The extracted knowledge, including situation/action profiles, decision-making, and dependency knowledge, is used to provide knowledge support. The knowledge base comprises historical logs, discovered knowledge patterns, situation/action profiles, and enterprise knowledge documents. This component acts as an information hub to provide knowledge support for problem-solving.

***Problem-solving process module.*** This module gathers production run-time information, such as problem situations. CBR is used to retrieve similar situation/action cases. This is described in Section 4.2. The system then suggests relevant documents and possible knowledge patterns related to the retrieved similar cases. Workers can then execute a specific problem-solving process and obtain knowledge support from the knowledge recommendation module. The problem-solving steps, including the situations, actions, and corresponding knowledge documents accessed, are recorded in the historical log.

***Knowledge discovery module.*** This module searches the historical log file to discover situation/action profiles and knowledge patterns. The following gives an overview of the knowledge discovery module. Further details are presented in Section 4.2.

- ***Discovering situation/action profiles.*** For specific situations or actions, relevant information (documents) accessed by workers is recorded in the problem-solving log. Historical codified knowledge (textual documents) can also provide valuable knowledge for solving the target problem. Information Retrieval (*Automatic Indexing*) techniques are used to extract the key terms of relevant documents for a specific situation or action. The extracted key terms form the situation/action profile, which is used to model the information needs of the workers. The knowledge support system then uses the profile to gather relevant information and help workers solve the target problem. Note that relevant information about a situation/action may vary due to a change of enterprise environment. The situation/action profiles can be used to gather existing and new relevant knowledge documents for a specific situation/action.

- ***Discovering decision-making and dependency knowledge.*** We assume that a generic problem-solving process is specified by experts to solve a problem or a set of similar problems encountered on a production line. When the production line encounters a problem, a problem-solving process is initiated. The situations occurred in a problem may vary due to the uncertainty of the constantly changing business environment. Moreover, different workers may take different actions to solve a problem according to their skills and experience. The problem-solving log records historical problem solving instances. The problem-solving process consists of several stages. *Association rule mining* is used to discover decision-making knowledge patterns (intra-relationships) hidden in a specific stage. *Sequential pattern mining* is employed to discover dependency knowledge patterns (inter-relationships) between different stages (chain reaction). This work employs the *Apriori* algorithm to find two kinds of rule patterns: association patterns of decision-making knowledge and sequential patterns of dependency knowledge. The discovered rule patterns form the basis of decision-making and dependency knowledge. When a situation or action matches a specific knowledge pattern, the associated situations or actions will be suggested as knowledge support.

***Knowledge recommendation module.*** This module constructs a knowledge support network based on the discovered knowledge patterns and situation/action profiles. A knowledge support network (KSN) is a conceptual representation of knowledge for a specific problem-solving process. It recommends situation/action relevant documents and decision-making/dependency knowledge as knowledge support. As noted previously, the situation/action profiles are used to gather existing and new relevant knowledge documents for a specific situation/action. The situation relevant documents help determine the cause of a problem, while the action-relevant documents (operating procedures and guidelines) instruct workers how to solve it. The KSN also comprises decision-making and dependency knowledge patterns extracted from the knowledge discovery module, and suggests frequently adopted actions for handling the problem situation. Dependency knowledge patterns are suggested to help workers infer possible cause-effect relationships and make appropriate action plans across problem-solving stages. The knowledge patterns and relevant documents provide practical knowledge support to help workers solve problems. Further details are presented in Section 4.3.

## 4.2. Discovery of problem-solving knowledge

This section describes the procedure of discovering knowledge from historical problem-solving logs, as shown in Fig. 3. To illustrate the proposed approach, we use data from the log file of a semiconductor foundry’s intranet portal, which contains the problem-solving log for handling problems on the production line. The company operates wafer manufacturing fabs to provide the industry with leading-edge foundry services. The log file records the encountered situation and the action taken at each problem stage. The system also contains documents accessed by workers for each situation/action during the problem-solving process. The data fields of the log include user data and problem-solving data. User data comprises factory, department, and user-role data. Problem-solving data contains the subject (text description) and attribute values of the situation/action, the stages, and the documents accessed.

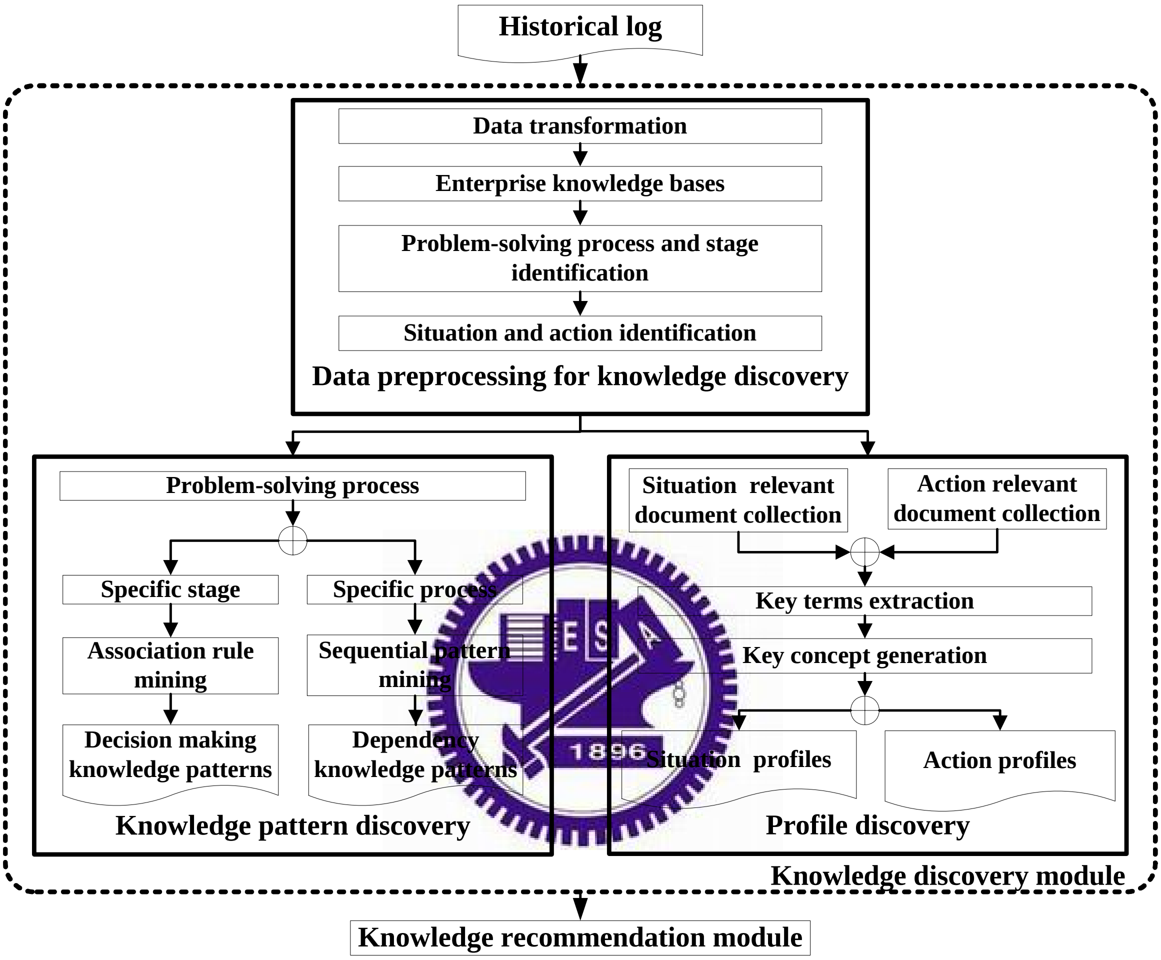


Fig. 3: The procedures of knowledge discovery process.

### 4.2.1. Data preprocessing for knowledge discovery

The data preprocessing module performs data cleaning, integration, and transformation for further knowledge discovery. The data cleaning task removes inconsistent data from the historical log. Each textual document is transformed into a term vector, i.e., a feature vector of weighted terms, using the *tf-idf* approach described in Section 2.3. The term vectors of accessed documents are then used by the profile discovery module to generate situation/action profiles. Furthermore, the data records are preprocessed to determine the problem-solving stages and the subject/attribute values of the situations/actions. The extracted values are used to identify the situations/actions for CBR. The production process, problem-solving process, and the term vectors of accessed documents are integrated into the enterprise's knowledge base.

***Problem-solving process and stage identification.*** The Stage field records the problem category, problem-solving process, and the stage. For example, "Equipment / Water-supply / Engineering - Improvement" shows that the problem category is "Equipment"; the problem solving process is "Water supply"; and the stage is "Engineering Improvement". The stage field is extracted from the data record to identify the problem-solving process and its stages.

### 4.2.2. Situation/action identification and case-based reasoning

Each situation or action is a case that is characterized by a text description and a set of attribute values. The attribute values provide additional features, such as the symptoms of a situation or the standard operating procedures of an action to identify the situation/action case. Both the text description and attribute values contribute to similarity matching and situation/action identification. For historical problem-solving instances, similar situation/action cases are transformed into the same situation/action identifier to facilitate the mining of decision-making and dependency knowledge patterns. Moreover, for the target situation/action, namely, the case workers are currently handling, the system identifies an existing case identifier or retrieves similar cases based on CBR. In the following, we describe the steps taken to transform existing cases and how to compute the similarity measures for case-based reasoning.

***Extraction of identifying term vectors.*** The data stored in the Subject field of an existing case is a text description of the situation/action. For example, Subject: "FAB8D Cu-BSC DI Water flow capacity insufficient issue" is the description of the situation - insufficient water flow capacity. The terms extracted from the subject field are used to identify the situation/action. Note that the terms are extracted using term transformation steps, including case folding, stemming, and stop word removal. We simply extract the terms without considering the term frequency, since the subject field generally contains a short text description. The extracted terms form ***identifying terms*** to identify a situation/action case. Moreover, the user needs to provide a text description for the target case, namely, the situation or action which he/she is handling. Similarly, the identifying terms of the target case are extracted from the text description using the term transformation steps. Let $T_j$ be the set of identifying terms extracted from the subject field of a situation/action case $C_j$. An identifying term vector $\vec{C}_j$ is created to represent $C_j$. The weight of a term $t_i$ in $\vec{C}_j$ is defined by Equation 5.

$$w(t_i, C_j) = \begin{cases} 1 & if\ t_i \in T_j \\ 0 & otherwise \end{cases}. \tag{5}$$

Equation. 6 defines the similarity value $sim^T(C_k, C_j)$ of two situation/action cases $C_k$ and $C_j$ based on their text descriptions. The similarity value is derived by computing the *cosine value* of the identifying term vectors of $C_k$ and $C_j$.

$$sim^T(C_k, C_j) = \text{cosine}(\vec{C}_k, \vec{C}_j) = \frac{\vec{C}_k \bullet \vec{C}_j}{\left|\vec{C}_k\right|\left|\vec{C}_j\right|} \tag{6}$$

***Similarity value by attribute.*** An attribute value may be nominal, binary, or numeric. For numeric attributes, a data discretization process is conducted to transform their values into value ranges or user-defined concept terms (such as *low*, *middle* or *high*). Equation 7 defines the similarity value $sim^A(C_k\ (attrb_x);\ C_j\ (attrb_x))$ of two situation/action cases $C_k$ and $C_j$, derived according to their values of attribute $x$; $value(C_k\ (attrb_x))$ denotes the transformed value of attribute $x$ of $C_k$, which is calculated by the discretization process.

$$sim^A(C_k(attrb_x), C_j(attb_x)) = \begin{cases} 1 & \text{if value}(C_k(attrb_x)) \text{ equals value}(C_j(attb_x)) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

***Similarity function for case-based reasoning.*** Equation 8 defines the similarity function used to compute the similarity measure between two cases $C_k$ and $C_j$. The similarity function is modified from Guardati (1998) by considering the cosine measure and attribute discretization.

$$similarity(C_k, C_j) = w_T sim^T(C_k, C_j) + \sum_{x=1}^{m} w_x sim^A(C_k(attrb_x), C_j(attb_x)), \tag{8}$$

where $sim^T(C_k, C_j)$ is the similarity value derived from the identifying term vectors of $C_k$ and $C_j$; $sim^A(C_k\ (attrb_x);\ C_j\ (attrb_x))$ is the similarity value obtained from the values of attribute $x$; $w_T$ is the weight factor for the text description, and $w_x$ is the weight given to attribute $x$. Note that the summation of $w_T$ and all $w_x$ is equal to 1.

***Transforming existing cases.*** Similar cases are transformed into the same situation/action identifier to discover decision-making and dependency knowledge patterns. The similarity measures among existing cases are computed using Equation 8. A threshold $\theta$ is defined to identify cases with high similarity measures (i.e., $similarity(C_k, C_j) > \theta$). Cases with the same or high similarity measures are transformed into the same situation/action identifier. The transformation procedure is conducted in an incremental and greedy manner. Assume that $r$ situation identifiers have been created. For each $S_i$ of $r$ situation identifiers, one or more situation cases have been transformed into $S_i$. $C_k$ is the situation case that needs to be transformed into a situation identifier. Let $minsim(C_k, S_i)$ be the *minimum* $similarity(C_k, C_j)$ over all $C_j$ that is transformed into $S_i$. The procedure finds a situation identifier $S_f$ such that $minsim(C_k , S_f)$ is the *maximum* of $minsim(C_k, S_i)$ over all $S_i$ (for $i$ = 1 to $r$). For a situation case $C_k$, $C_k$ is transformed into $S_f$, if $minsim(C_k, S_f)$ is greater than $\theta$; otherwise, $C_k$ is transformed into a new situation identifier. The transformation procedure for action cases is conducted in a similar way. Table 1 lists the situations and actions in each stage of the water supply problem-solving process.

Table 1: Situations/actions in the water supply problem-solving process

| **Water supply problem-solving process** | |
|---|---|
| **Situations** | **Actions** |
| **[S1]** Flow Capacity Abnormal Issue (Subject: Insufficient/Unstable/Overflow) | **[A1]** Testing based on SOPs |
| **[S2]** Supply Quantity Abnormal Issue (Subject: Insufficient/Unstable/Overflow) | **[A2]** Consult expert information |
| **[S3]** Power Supply Abnormal Issue (Subject: Insufficient /Unstable/Excess) | **[A3]** Modify the configuration |
| **[S4]** Water Pressure Abnormal Issue (Subject: Insufficient/Unstable/Excess) | **[A4]** Recycle the material |
| **[S5]** Cleaning Quality Abnormal Issue (Subject: Low/Unstable) | **[A5]** Monitor the output |
| **[S6]** Pipe Abnormal Issue (Subject: Broken/Clogged) | **[A6]** Discuss with workers |
| **[S7]** Controller Temperature Abnormal Issue (Subject: Excess/Unstable) | **[A7]** Report the outcome |
| … | … |

***Case-based reasoning for a target case.*** A target case is a situation or action that a worker is currently handling. After entering a target case $C_k$ of a situation/action, the system identifies an existing case identifier of $C_k$ or retrieves similar situation/action cases if $C_k$ is a new case. The similarity measures between the target case and previous cases are computed using Equation 6. The identification procedure is similar to the transformation procedure. Assume there are $r$ situation identifiers. Let $minsim(C_k, S_i)$ be the *minimum* $similarity(C_k, C_j)$ over all $C_j$ transformed into $S_i$. The procedure finds a situation identifier $S_f$ such that $minsim(C_k , S_f)$ is the *maximum* of $minsim(C_k, S_i)$ over all $S_i$ (for $i$ = 1 to $r$). An existing situation identifier $S_f$ is identified if $minsim(C_k , S_f)$ is greater than $\theta$; otherwise, the situation is a new case and the system assigns a new identifier to it. The case and its identifier are then stored in the knowledge base, and CBR is initiated to retrieve similar cases based on their similarity measures and to suggest possible knowledge related to the similar cases.

### 4.2.3. Discovery of situation/action profiles

The log records the set of documents accessed for handling a situation/action. For example, Doc_ID: "AF0001C0F25" is a planning report that describes how to deal with the abnormal water quality in the DI water system. DI (de-ionized) water is ultra-pure water used for wafer cleaning in semiconductor manufacturing. The term vectors of the documents are derived using Equation 1, i.e., the *tf-idf* approach described in Section 2.3.

A situation/action profile is also represented as a term vector (a feature vector of weighted terms), which is derived by analyzing the set of documents accessed for handling the situation/action case. Each document $d_j$ is pre-processed and represented as a term vector $\vec{d}_j$. Let $D_S$ denote the set of documents accessed to handle the situation/action $C_S$. A centroid approach is used to derive the profiling term vector $\vec{P}_S$ of $C_S$ by averaging the term vectors of documents in $D_S$. Equation 9 defines the weight of a term $k_i$ in $\vec{P}_S$.

$$w(k_i, C_S) = \frac{1}{|D_S|} \sum_{d_j \in D_S} w(k_i, d_j) \,. \qquad \textbf{(9)}$$

***Retrieval of situation/action relevant documents.*** The system recommends/retrieves relevant knowledge documents to help workers solve problems based on the situation/action profiles. The key contents of a codified knowledge document are represented as a term vector. The situation/action profile of a case $C_S$ is expressed as a profiling term vector $\vec{P}_S$. The cosine measure of term vectors, described in Section 2.3, is used to derive the similarity measure. Let $\vec{d}_j$ be the term vector of document $d_j$. The cosine measure of $\vec{P}_S$ and $\vec{d}_j$, cosine($\vec{P}_S$, $\vec{d}_j$), is the similarity measure between the situation/action and document $d_j$. Documents with the top-N similarity measures are selected as relevant documents.

### 4.2.4. Discovery of knowledge patterns

***Generic problem-solving process.*** Recall that a generic problem-solving process is specified by experts to solve a problem. The specification includes the stages and their execution order. This work focuses on the execution of a sequence of stages. For example, the generic water supply problem-solving process is "Normal Management Stage (NM Stage) → Engineering

Improvement Stage (EI Stage) → Exception Management Stage (EM Stage) → Quality Improvement Stage (QI Stage) → Maintenance Management Stage (MM Stage). For any given problem, the situations may vary; thus the follow-up actions may also vary.

***Discovery of decision-making knowledge patterns.*** Association rule mining is used to discover decision-making knowledge hidden in each problem solving stage. In this paper, we adopt the *Apriori* algorithm to find the frequent association patterns of decision-making knowledge, namely situation → action. The criteria of minimum support and confidence are used to filter out non-frequent patterns. The discovered rule patterns form the basis of decision-making knowledge. When a situation matches a specific knowledge pattern, the associated action will be suggested as knowledge support. For example, the discovered decision-making knowledge patterns in the quality improvement stage (QI) are:

- QI_S1 → QI_A3

  If the situation "Water Flow Capacity Abnormal Issue (Insufficient)" occurs, take the "Modify the configuration" action.

- QI_S6 → QI_A5

  If the situation "Pipe Abnormal Issue (Clogged)" occurs, take the "Monitor the output" action.

***Discovery of dependency knowledge patterns.*** Sequential pattern mining is adopted to discover the dependency knowledge patterns (inter-relationships) hidden between stages. We use a modified *Apriori* algorithm for *sequential pattern mining* to discover the frequency of similar situations and actions across different stages. The criterion of minimum support is used to filter out non-frequent (chain reaction) relationships. When a situation or action matches a specific knowledge pattern, the chain of situations or actions is suggested as knowledge support. Some examples of dependency knowledge patterns are:

- EI_S1 → QI_S4

  If the situation "Flow Capacity Abnormal Issue (Overflow)" occurs in the engineering improvement stage, then it is likely that the situation "Water Pressure Abnormal Issue (Excess)" will occur in the quality improvement stage.

- EI_A4 → EM_S1

  If the "Recycle the material" action is taken in the engineering improvement stage, then the situation "Flow Capacity Abnormal Issue (Unstable)" is likely to occur in the exception management stage.

- NM_A3 → QI_A5

  If the "Modify the configuration" action is taken in the normal management stage, then the "Monitor the output" action is likely to be taken in the quality improvement stage.

- EI_S2 → EM_A7 → QI_S1

  The situation "Supply Quantity Abnormal Issue (Insufficient)" in the engineering improvement stage frequently triggers the "Report the outcome" action in the exception management stage; and then triggers the "Flow Capacity Abnormal Issue (Insufficient)" situation in the quality improvement stage.

The dependency knowledge patterns denote the chain reaction across different stages. This helps workers plan appropriate actions for different problem-solving stages. The decision-making and dependency knowledge patterns are integrated into the knowledge support network.

## 4.3. Knowledge support for problem-solving

This section describes the construction of the knowledge support network, which provides knowledge recommendations for problem-solving. The procedure is showed in Fig. 4.

### 4.3.1. Knowledge support network

A knowledge support network (KSN) is constructed from the output of the knowledge discovery module. The KSN comprises the specification of the generic problem-solving process, decision-making and dependency knowledge patterns, situation/action profiles, and relevant documents.

***Specification of a generic problem-solving process.*** The specification includes the problem description, the stage names, and their execution orders.

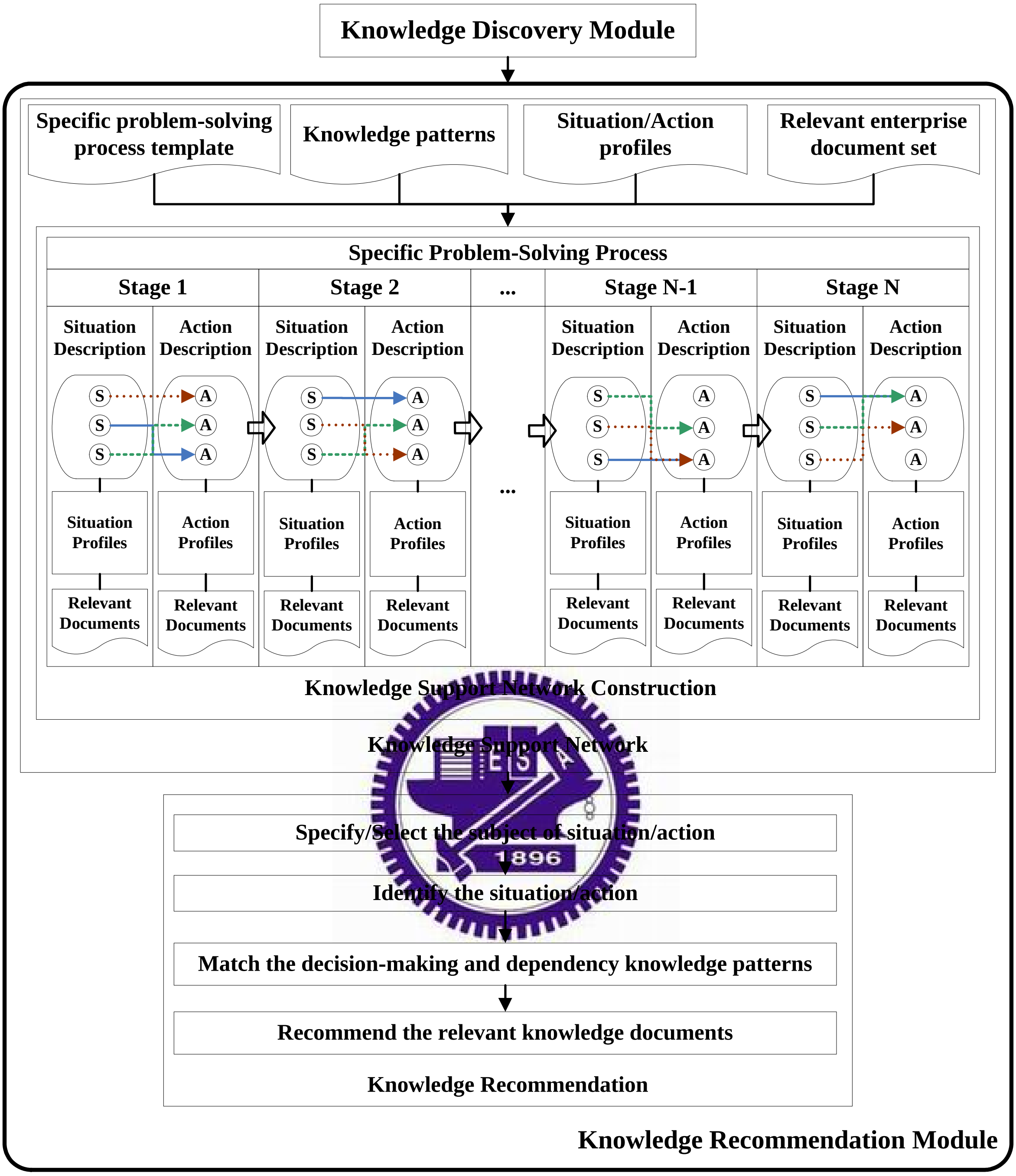


Fig. 4: The procedures of knowledge recommendation

***Decision-making knowledge patterns.*** Decision-making knowledge patterns indicate the frequent association of situations and actions in the problem-solving process. For each stage, a decision-making knowledge pattern *situation* → *action* indicates that the *action* frequently adopted to solve the encountered *problem situation*. The KSN provides frequently adopted actions for handling a specific situation based on the decision-making knowledge patterns. Fig. 5 shows the discovered decision-making knowledge patterns in the KSN of the water supply problem-solving process.

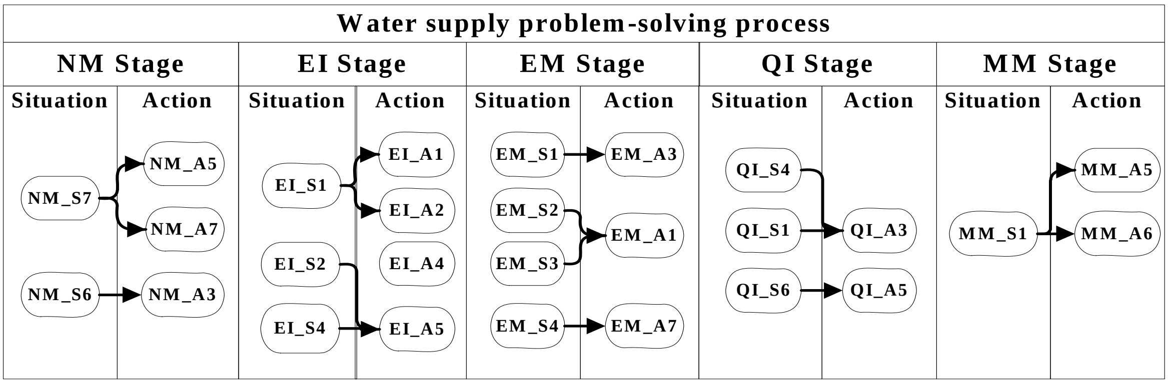


Fig. 5: Decision-making knowledge patterns in a knowledge support network

***Dependency knowledge patterns.*** For a specific problem-solving process, the dependency knowledge patterns express the relationships between situations and actions across different stages. For example, the dependency knowledge pattern “EM_S3 → MM_A5” implies that if a “Power Supply Abnormal Issue (Unstable)” situation occurs in the exception management stage, then the “Monitor the output” action is frequently taken in the maintenance management stage. A dependency knowledge pattern “EI_S4 → QI_A5→ MM_A6” implies that a “Water Pressure Abnormal Issue (Unstable)” situation in the engineering improvement stage will trigger a “Monitor the outcome” action in the quality improvement stage; and then trigger a “Discuss with the worker” action in the maintenance management stage. Based on the dependency knowledge patterns, the KSN provides triggering situations or actions across different stages, which help workers predict possible situations in later stages, and plan appropriate actions. Fig. 6 shows the dependency knowledge patterns in a KSN.

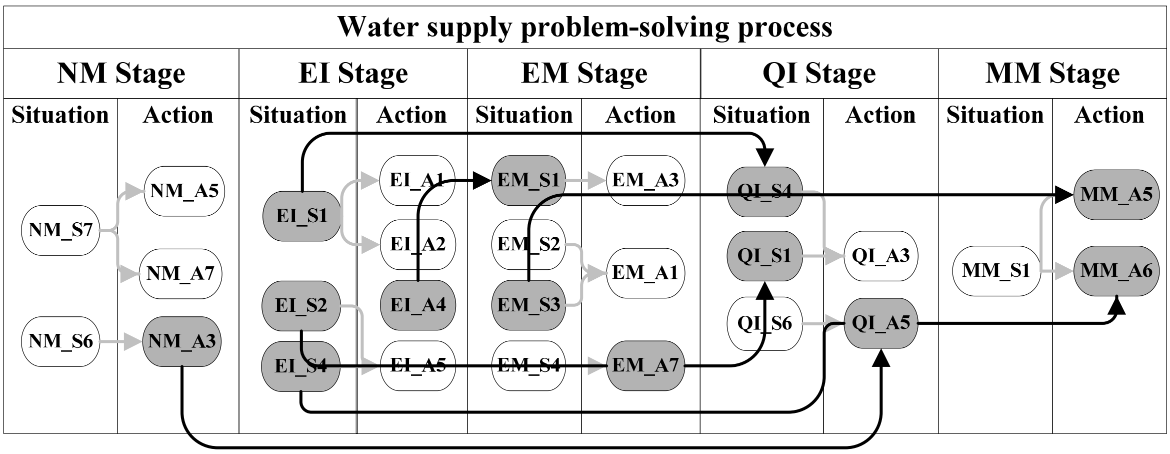


Fig. 6: Dependency knowledge pattern in a KSN

***Situation/action profiles and relevant documents.*** The situation/action profiles are generated from the accessed documents, as described in section 4.2. For example, in the situation of abnormal water quantity, the accessed documents include: “DI analytical machine water quantity recording” and “DI GCHC machine water quantity recycling.” The situation profile is generated from the accessed documents. Once a worker encounters a problem situation or decides to take a particular action, the KSN provides relevant documents as knowledge support based on the situation/action profiles. Fig. 7 illustrates a situation profile and the relevant documents for the water supply problem-solving process. Based on the situation/action profiles, the knowledge support network gathers previous and new relevant documents, such as “DI analytical machine water quantity recording” and “DI GCHC machine water quantity recycling” and new documents “8D DI system waste water quantity estimation” and “8D UF Flush water quantity recycling”.

### 4.3.2. Knowledge recommendation

The problem-solving process module employs CBR to identify the current situation or retrieve similar situation-cases according to the similarity measures. The knowledge recommendation module then suggests relevant documents according to the situation profile of the current situation or similar cases, as shown in Fig. 7. The system also recommends relevant action documents (e.g., operating procedures and guidelines) according to the action profile. Note that the top-N relevant documents are recommended according to the cosine measure of the term vectors of the documents and the situation/action profiles, as described in Section 4.2.

Moreover, the system suggests possible actions for handling the current situation according to the decision-making knowledge patterns. Note that the actions in the decision-making patterns (i.e., situation => action) whose left-hand side match the current situation are suggested and ranked according to the confidence values of the rules. Dependency knowledge patterns are also suggested to help workers predict a possible chain reaction across different stages and develop appropriate action plans.

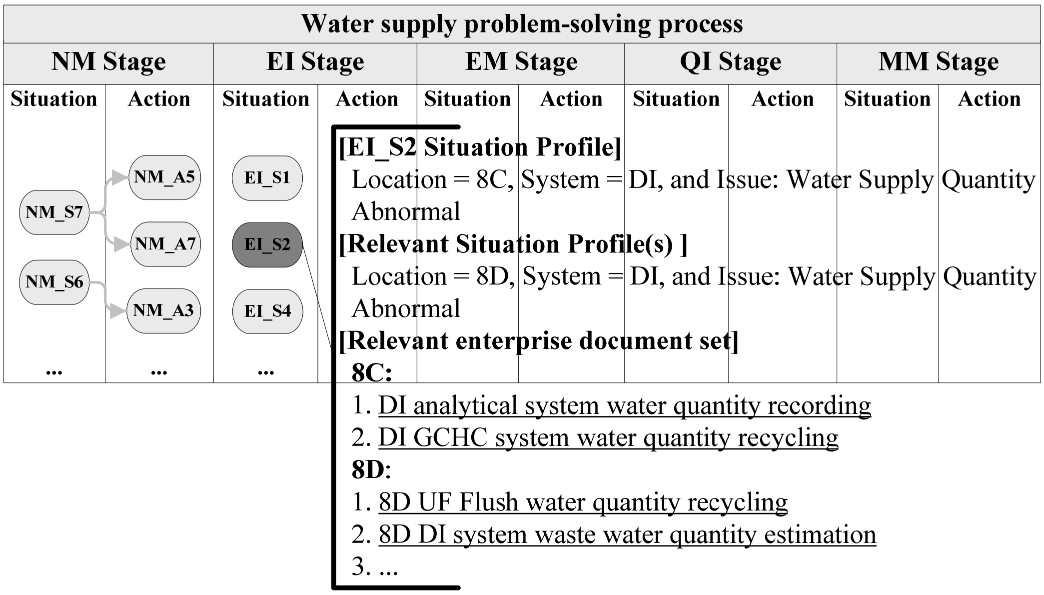


Fig. 7: Situation profile and relevant documents

## 4.4. System implementation

We developed a prototype system to demonstrate the effectiveness of the proposed knowledge support system for problem-solving. The implementation is conducted using several software tools, including the Java(TM) 2 Platform Standard Edition Runtime Environment Version 5.0, Java Server Pages, and Macromedia Dreamweaver MX. A web and application server is setup on Apache Tomcat 5.5.7, and Microsoft SQL Server 2000 is used as the database system for storing data related to the problem-solving process and codified knowledge documents. The data mining tool Weka 3.4 is used to discover knowledge patterns in the historical problem-solving log.

The generic problem-solving process, situation/action profiles, decision-making and dependency knowledge patterns form the knowledge support network. The network provides relevant knowledge documents, and suggests decision-making and dependency knowledge patterns. The problem-solving knowledge support system is integrated with the knowledge support network to provide more effective knowledge support for browsing problem-solving knowledge patterns. The interface of the problem-solving knowledge support system includes the system frames for user login, search engine, and user-guide. A worker Annie logs into the system and gets a problem list. Once she selects a generic problem-solving process to browse, the problem (e.g., water supply problem) can be browsed further in the system platform, as shown in Fig. 8.

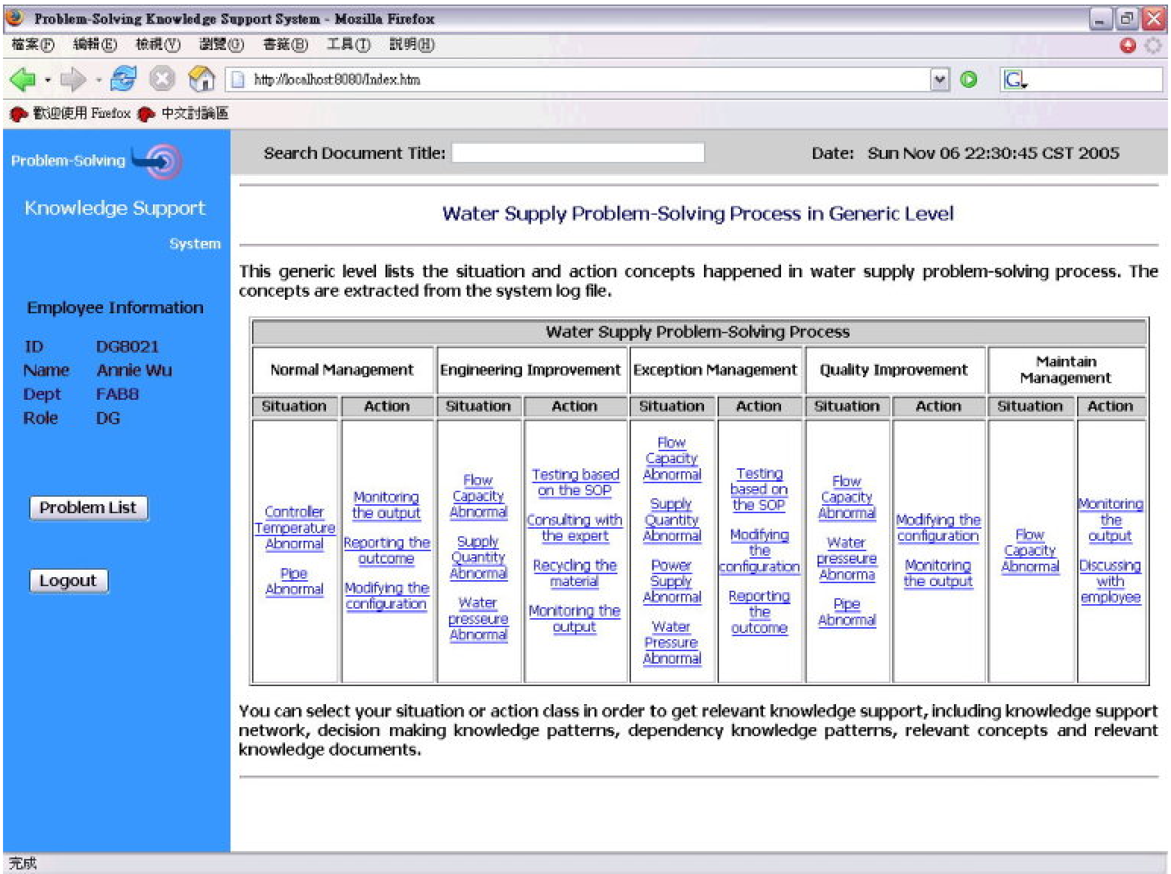


Fig. 8: A generic water supply problem-solving process

Annie can choose a situation/action to get knowledge support. Fig. 9 shows an example where Annie chooses the situation “Controller Temperature abnormal issue” in the normal management stage of the water supply problem-solving process. The system presents the decision-making knowledge patterns: “Controller Temperature abnormal situation → Monitoring the output action” in the knowledge support network. The relevant documents for the situation “Controller Temperature abnormal issue” are shown below the page. The system also displays the key terms of the profile for the situation, including Controller Temperature Abnormal and Controller operation status. The key terms give workers an overview of the current situation. By reading the relevant knowledge documents, AF0001C0F25 and A9600400762, Annie can understand the situation, identify its causes, and take appropriate action. Moreover, the suggested dependency knowledge pattern can help Annie realize a possible chain reaction across different stages. Accordingly, workers can develop appropriate action plans across different problem-solving stages.

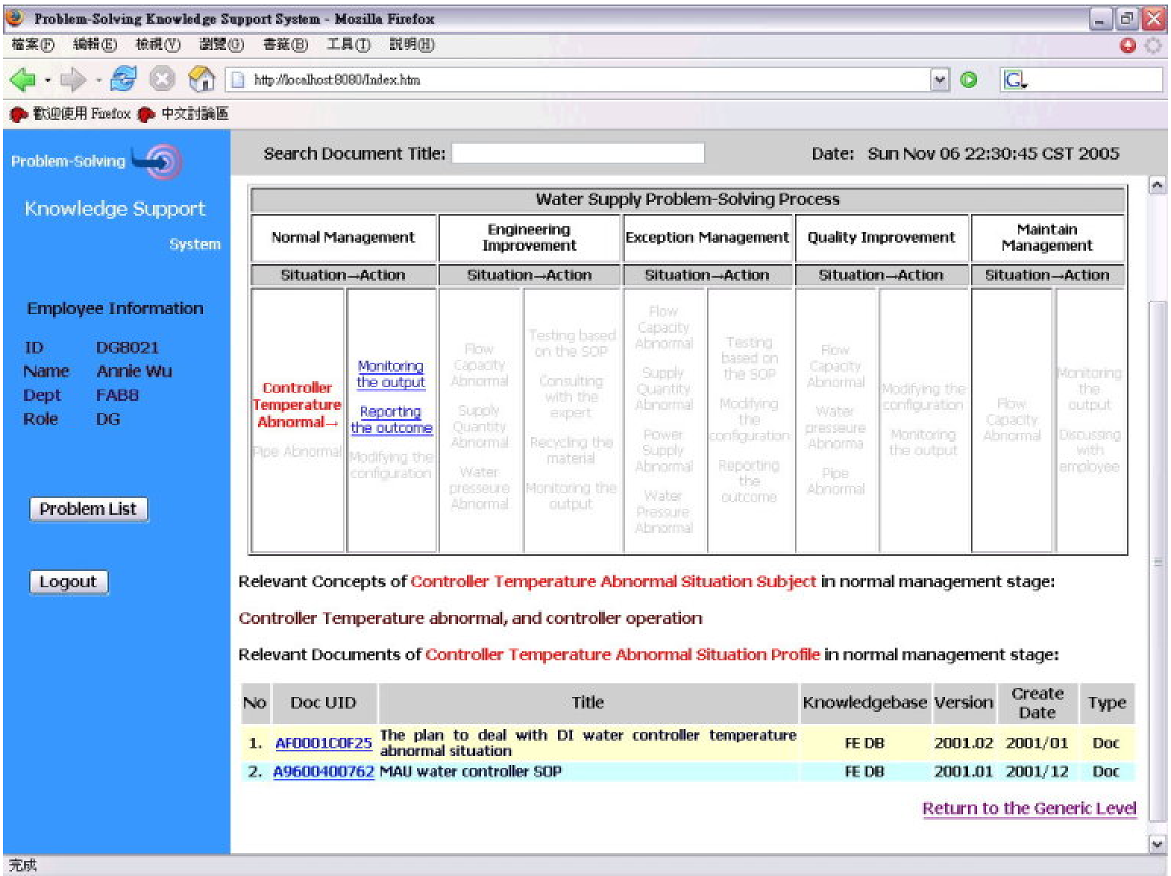


Fig. 9: Decision-making knowledge patterns for the water supply problem

# Chapter 5. Knowledge Support based on Case-based Reasoning, Data Mining, and Rule Inference

In this chapter, we describe the knowledge support based on CBR, data mining, and rule inference techniques, including context-based knowledge support framework for problem-solving, discovery of context-based problem-solving knowledge, the prototype system implementation, discussions, and comparisons.

## 5.1. Context-based knowledge support framework for problem-solving

The adapted system framework comprises a problem-solving process, context-based inference rule discovery, context-based situation profile discovery, context-based knowledge pattern discovery, and knowledge recommendation modules, as illustrated in Fig. 10.

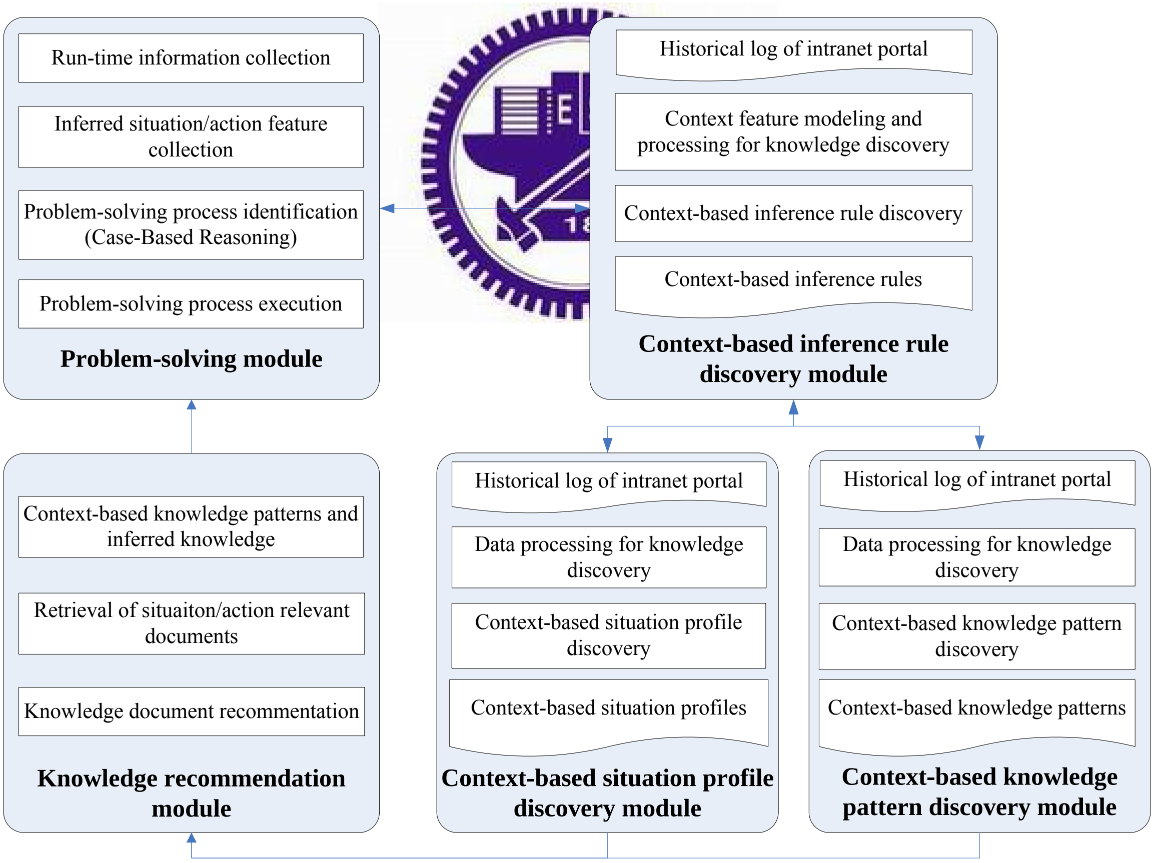


Fig. 10: The adapted system framework of context-based knowledge support

***Problem-solving module.*** This module gathers production line run-time information, such as problem situation features. Collected and inferred situation/action features help CBR to retrieve similar situation/action cases. Workers can then execute a specific problem-solving process and obtain relevant knowledge documents from the knowledge recommendation module. The problem-solving steps, including the situations, actions, and corresponding knowledge documents, are recorded in the historical log. This is described in Section 5.2.1.

***Context-based inference rule discovery module.*** This module gets situation description, attributes, features and relevant context information from problem-solving process module. The framework uses constraint-based association rule mining to discover the association between situation features and relevant context characteristics. The discovered context-based inference rules are used to infer more relevant problem features in order to assist CBR identify problem situation encountered. The module is described in Section 5.2.1.

***Context-based situation profile discovery module.*** This module analyzes the historical log file to discover context-based situation profiles. For specific situations/actions in certain context, relevant information (documents) accessed by workers is recorded in the problem-solving log. Historical codified knowledge (textual documents) can also provide valuable knowledge for solving the target problem. Information Retrieval (*Automatic Indexing*) techniques are used to extract the key terms of relevant documents of a specific situation for certain context. The extracted key terms form the context-based situation profile, which is used to model the information needs of the workers in certain context. The knowledge support system then uses the profile to gather relevant information and help workers solve the target problem. Further details are presented in Section 5.2.2.

***Context-based knowledge pattern discovery module.*** This module discovers context-based knowledge patterns for situation in certain context. Based on collected attributes, context characteristics, and relevant context-based inference rules, the system continually infers relevant situation features and actions as clues to form the context-based knowledge patterns, including decision-making and dependency knowledge. We design a scoring mechanism to represent the action importance of discovered context-based decision-making knowledge patterns. The context-based knowledge pattern with its score helps workers take reasonable actions in certain context of problem-solving process. The details of this module are described in Section 5.2.3.

***Knowledge recommendation module.*** This module recommends context-based knowledge patterns, inferred knowledge, and relevant situation documents as context-based knowledge support. The context-based knowledge patterns and inferred knowledge assist workers take appropriate actions for solving situation or realize dependency of situations/actions in certain context. As noted previously, the context-based situation profiles are used to gather existing and new relevant knowledge documents of a specific situation for certain context. The relevant documents provide practical knowledge support to help workers solve problems. Further details are presented in Section 5.2.4.

## 5.2. Discovery of context-based problem-solving knowledge

This section describes the procedure of discovering context-based knowledge from historical problem-solving logs. To illustrate the proposed approach, we use data from the log file of a semiconductor foundry's intranet portal, which contains the problem-solving log for handling problems on the production line. The company operates wafer manufacturing fabs to provide the industry with leading-edge foundry services. The log file records the encountered situations and actions taken. The system also contains profiles accessed by workers for each situation of different context during the problem-solving process. The data fields of the problem-solving log include the situation features and context information.

### 5.2.1. Context-based situation identification and case-based reasoning

Each situation or action is a case that is characterized by a text description, situation features and a set of attribute values. The attribute values provide important information such as the symptoms of a situation to identify the situation case. Situation features are analyzed from previous problem situations/actions and can be predefined in system. Such situation features may be collected in run-time by the system or selected by the user. For undefined situation causes, users need to provide a text description of the situation. The text description can be used to extract identifying terms for the situation. Moreover, situation features collected by the system are usually partial and incomplete. Context-based inference can be initiated to infer more situation features. The text descriptions, situation features, attribute values contribute to similarity matching and situation identification. For the target situation/action, namely, the case workers are currently handling, the system identifies an existing case identifier or retrieves similar cases based on CBR.

***Extraction of identifying term vectors.*** The data stored in the Subject field of an existing case is a text description of the situation. For example, Subject: "FAB8D Cu-BSC DI Water flow capacity insufficient issue" is the description of the situation - insufficient water flow capacity. The terms extracted from the subject field are used to identify the situation and attributes, e.g., situation name: insufficient water flow capacity; factory name: FAB8; department identification: D; system type: DI; system status: water flow capacity insufficient. The relevant context entity and feature include staff: Annie; role: DG; time: 20040502-PM; location: Hsinchu, service name: DI water supply service, etc. Note that the terms are extracted using term transformation steps, including case folding, stemming, and stop word removal. We simply extract the terms without considering the term frequency, since the subject field generally contains a short text description. The extracted terms form identifying terms to identify a situation case. Moreover, the user needs to provide a text description for the target case, namely, the situation or action which he/she is handling. Similarly, the *identifying terms* of the target case are extracted from the text description using the term transformation steps. Let $T_j$ be the set of identifying terms extracted from the subject field of a situation case $C_j$. An identifying term vector $\vec{C}_j$ is created to represent $C_j$. The weight of a term $t_i$ in $\vec{C}_j$ is defined by Equation 5. Equation 6 defines the similarity value $sim^T(C_k, C_j)$ of two situation cases $C_k$ and $C_j$ based on their text descriptions. The similarity value is derived by computing the cosine value of the identifying term vectors of $C_k$ and $C_j$.

***Similarity value by attribute.*** An attribute value may be nominal, binary, or numeric. For numeric attributes, a data discretization process is conducted to transform their values into value ranges or user-defined concept terms (such as low, middle or high). Equation 7 defines the similarity value $sim^A(C_k\,(attrb_x), C_j\,(attrb_x))$ of two situation cases $C_k$ and $C_j$, derived according to their values of attribute $x$; $value(C_k\,(attrb_x))$ denotes the transformed value of attribute $x$ of $C_k$ , which is calculated by the discretization process.

***Context modeling.*** The context information is any information about an entity status. An entity can be the user, physical location, service, or service relevant object, etc. Due to the variety of context information, it is not easy to represent the complete context information of an entity. Therefore, based on the problem-solving environment, this work uses a modeling mechanism which composes with three levels to formalize the context information including *Context entity level, Context feature level* and *Context association level*.

- ***Context entity level.*** This level represents the conceptual abstraction of context entity spread in a problem-solving environment, includes physical, organization, process, staff, service, and document entities, etc.
- ***Context feature level.*** The context feature may be predefined by a domain expert that shows relevant information of a specific entity. A context entity may include one or more context features, for example, a physical entity covers the identification, time and location features; an organization entity may include the factory and department features; a process entity contains stage, task, and status features; a staff entity has user, role, degree, and activity features; a service entity may involve with system, component, and parameter features; a document entity includes original, type, author, and score features, etc.
- ***Context association level.*** This level defines the association relationship between relevant features and attributes of the context entities. The association relationship is used to collect more relevant information of current problem-solving process based on context characteristics. We list some pre-defined association types as follows.
  - The *organization-staff association* describes the relationship between organization and staff entity, e.g., Annie belongs to DG role in B department of Fab8 factory.
  - The *staff-process association* describes the relationship that user-role carries out the specific process, e.g., DG-Annie carries out the water supply problem-solving process.
  - The *staff-service association* describes the relationship that user-role uses the specific system service, e.g., DG-Annie uses the DI water supply system service.
  - The *process-service association* shows the relationship between the process and service entity, e.g., the water supply process contains the DI water supply and pipe control system services.
  - The *process-document association* describes the relationship that some documents support specific process, e.g., expert or experiential reports of specific situation.
  - The *service-document association* shows the relationship that some documents belong to specific service, e.g., user guide or technical documents of specific system service.

Based on context modeling, the system proactively collects the relevant context entities and features of current situation. For example, when staff Annie suffers from the controller temperature abnormal situation, the relevant entities include physical time, location, organization, Annie, water supply problem-solving process, DI water supply system, and relevant knowledge documents, etc. The system also gathers relevant features of context entities in a controller temperature abnormal situation, such as physical time: 20040502-PM 3:24; location: Hsinchu; factory: Fab8; department: B; user-role: DG- Annie; process: water supply problem-solving process; stage: normal management stage; situation: controller temperature abnormal situation; service: DI water supply system service; document: AF0001C0F25; author: PTC; Score: 4; original: DIFF knowledge base, etc. The collected context entities and features of specific situation are stored in enterprise knowledge base for context-based inference rule discovery. Context entities and situation/action features are represented in some meta-rule format predefined by expert. The proposed system enforces the constraint-based association rule mining to discover the context-based inference rules from the problem-solving log.

***Context-based inference rule mining.*** The context-based inference rules discovered from association rule mining represent the associations of situation features and context characteristics. The rule format is shown as Equation 10:

$$[\boldsymbol{feature_p \ldots} \text{ and } \boldsymbol{context_q \ldots}] \rightarrow [\boldsymbol{feature_r}] \qquad [\boldsymbol{Support = s\%, Confidence = c\%}] \quad (10)$$

For example, for the controller temperature abnormal situation, the features of staff entity: “Annie” and service entity: “DI water supply system service” are associated with the feature of DI water supply system service entity: “Parameter incorrect”. The context-based inference rule is shown as follows.

$$[\textit{Staff(Annie)} \text{ and } \textit{DI water supply system service()}] \rightarrow$$
$$[\textit{DI water supply system service(Parameter: incorrect)}] \ [\textit{Support} = 2\%, \textit{Confidence} = 13\%]$$

For specific situation, the collected context entities and features are used to discover relevant actions. The format of context-based inference rule is represented as Equation 11:

$$[\boldsymbol{feature_p} \ldots \text{ and } \boldsymbol{context_q} \ldots] \rightarrow [\boldsymbol{Action_r}] \qquad [\boldsymbol{Support = s\%, Confidence = c\%}] \quad (11)$$

For example, for the controller temperature abnormal situation, the features of staff entity: “Annie” and service entity: “DI water supply system service” are associated with the Action: “Reporting the outcome”. The context-based inference rule is shown as follows.

$$[\textit{Staff(Annie)} \text{ and } \textit{DI water supply system service()}] \rightarrow [\textit{Reporting the outcome action()}]$$
$$[\textit{Support} = 2\%, \textit{Confidence} = 13\%]$$

For specific problem-solving process, the collected context entities and features of specific situation are used to discover relevant situation features. Equation 12 shows the format of context-based inference rule that infers relevant action feature of specific situation; the format of context-based inference rule that infers relevant situation features of specific action is represented as Equation 13:

$$[\boldsymbol{feature_p} \ldots \text{ and } \boldsymbol{context_q} \ldots]_{Si} * \ldots \text{and } [\boldsymbol{feature_u} \ldots \text{ and } \boldsymbol{context_v} \ldots]_{Aj} * \rightarrow [\boldsymbol{feature_r}]_{Ak} \qquad [\boldsymbol{Support = s\%, Confidence = c\%}] \quad (12)$$

$$[\boldsymbol{feature_p} \ldots \text{ and } \boldsymbol{context_q} \ldots]_{Si} * \ldots \text{and } [\boldsymbol{feature_u} \ldots \text{ and } \boldsymbol{context_v} \ldots]_{Aj} * \rightarrow [\boldsymbol{feature_r}]_{Sk} \qquad [\boldsymbol{Support = s\%, Confidence = c\%}] \quad (13)$$

The examples are illustrated as follows. The feature of context entity Staff: “Annie” in controller temperature abnormal situation of Normal Management stage and the feature of context entity Staff: “PTC” in consulting with the expert action of Engineering Improvement stage are associated with the feature of DI water supply system service entity: “Parameter: increasing pressure” in modifying the configuration action of Exception Management stage. The context-based inference rule is shown as follows.

$$[\textit{Staff(Annie)}]_{NM\_S7} \text{ and } [\textit{Staff(PTC)}]_{EI\_A2} \rightarrow$$
$$[\textit{DI water supply system service(Parameter: increasing water pressure)}]_{EM\_A1}$$
$$[\textit{Support} = 1\%, \textit{Confidence} = 14\%]$$

The feature of context entity DI water supply system service: "Parameter: output value" in monitoring the output action of Normal Management stage and the feature of context entity Document: "A9600400762" in testing based on the SOP action of Engineering Improvement stage are associated with the feature of DI water supply system service entity: "Parameter: water quantity" in supply quantity abnormal situation of Exception Management stage. The context-based inference rule is shown as follows.

[*DI water supply system service(Parameter: output value)*]$_{NM\_A5}$ and

[*Document(A9600400762)*]$_{EI\_A1}$ $\rightarrow$

[*DI water supply system service(Parameter: water quantity )*]$_{EM\_S2}$

[*Support* = 3%, *Confidence* = 11%]

***Certainty Factor value of context-based inference rule.*** The certainty degree of system collected situation feature is set to 1. For inferred situation features, this work employs the method of *Certainty Factor* (*CF*) value (Shortliffe et al., 1975) to derive the certainty degree during the inference, as defined in Equation 3. The preceding set denotes run-time situation features and context characteristics; the succeeding set is the situation feature that we want to infer its certainty degree. For example, The *CF* value of the context-based inference rule: [Staff(Annie)] $\rightarrow$ [DI water supply system service(Parameter: incorrect)] is 0.033. The details of calculation are shown as follows.

[*Staff(Annie)*] $\rightarrow$ [*DI water supply system service(Parameter: incorrect)*]

[*Support* = 2%, *Confidence* = 13%]

*S*([*DI water supply system service(Parameter: incorrect)*]) = 10%

*CF*([*Staff(Annie)*] $\rightarrow$ [*DI water supply system service(Parameter: incorrect)*])

= (13%-10%)/(1-10%) =0.033

***Inference for situation features.*** Based on the *CF* value of situation feature and context-based inference rule, the inference process follows the rules defined in Equation 4. An example is illustrated in Fig. 11. The details of inference process are shown as follows. The context-based inference rule: [Role(DG)] $\rightarrow$ [Staff(Annie)] indicates the feature: DG of context entity: Role inferring the feature: Annie of context entity: Staff. Its *CF* value is 0.7. The *CF* value of [Service(Water Supply)] $\rightarrow$ [DI water supply system service ()] is 0.5. Then two context entities: [Staff(Annie)] and [DI water supply system service ()] have "AND" relationship. Its output *CF* value is 0.5. The *CF* value of [Staff(Annie) and DI water supply system service ()] → [DI water supply system parameter(Incorrect)] is 0.3. The *CF* value of [Pipe system service()] → [DI water supply system parameter(Incorrect)] is 0.2. Finally, there is a "JOIN" relationship with two inference conditions. The *CF* value of [Staff(Annie) and DI water supply system service ()] → [DI water supply system parameter(Incorrect)], [Pipe system service()] → [DI water supply system parameter(Incorrect)] is 0.3. Inferred situation features with high ranking of *CF* value are considered as the Inferred knowledge to assist CBR in identifying situation encountered.

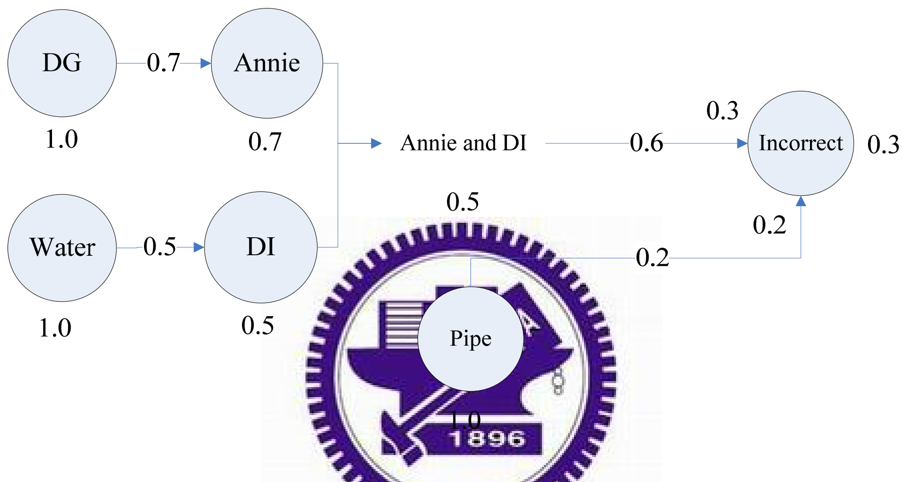


Fig. 11: An example of inference process

$CF(Annie) = CF(DG) * CF(IF\ DG\ THEN\ Annie) = 1.0 * 0.7 = 0.7$

$CF(DI) = CF(Service) * CF(IF\ Service\ THEN\ DI) = 1.0 * 0.5 = 0.5$

$CF(Annie \wedge DI) = MIN(CF(Annie), CF(DI)) = 0.5$

$$CF(Incorrect) = MAX(CF(Annie \wedge DI) * CF(IF\ Annie \wedge DI\ THEN\ Incorrect), CF(Pipe) * CF(IF\ Pipe\ THEN\ Incorrect)) = MAX(0.5 * 0.6, 1.0 * 0.2) = 0.3$$

Inferred situation features with high ranking of *CF* value are considered as the Inferred knowledge. Then the inferred knowledge assists CBR in situation identification. Let $F_j$ be the set of situation features of $C_j$ that are collected by the system or inferred by the context-based inference rules. A feature vector $\vec{C}_{Fj}$ is created to represent $C_j$. The weight of a feature $f_i$ in $\vec{C}_{Fj}$ is defined by Equation 14.

$$w(f_i, C_j) = \begin{cases} CF_i & if\ f_i \in F_j \\ 0 & otherwise \end{cases}. \tag{14}$$

If $f_i$ is inferred by the context-based inference rules, $w(f_i, C_j)$ is set to $CF_i$ - the inferred $CF$ value of $f_i$ ; if the $f_i$ is selected by the user or gathered by the system, $CF_i$ is set to value "1"; otherwise $w(f_i, C_j)$ is 0. Equation 15 defines the similarity value $sim^F(C_k, C_j)$ of two situation cases $C_k$ and $C_j$ based on their situation features. The similarity value is derived by computing the cosine value of the feature vectors of $C_k$ and $C_j$.

$$sim^F(C_k, C_j) = \text{cosine}(\vec{C}_{Fk}, \vec{C}_{Fj}) = \frac{\vec{C}_{Fk} \bullet \vec{C}_{Fj}}{\left|\vec{C}_{Fk}\right|\left|\vec{C}_{Fj}\right|} \tag{15}$$

***Similarity function for case-based reasoning.*** Equation 16 defines the similarity function used to compute the similarity measure between two cases $C_k$ and $C_j$. The similarity function is modified from Guardati (1998) by considering the combination of the similarity of text descriptions, attribute values and situation features.

$$\begin{aligned} similarity\ (C_k, C_j) = {} & w_T\, sim^T(C_k, C_j) + w_F\, sim^F(C_k, C_j) + \\ & \sum_{x=1}^{m} w_x\, sim^A(C_k(attrb_x), C_j(attb_x)) \end{aligned} \tag{16}$$

where $sim^T(C_k, C_j)$ is the similarity value derived from the identifying term vectors of $C_k$ and $C_j$; $sim^F(C_k, C_j)$ is the similarity value derived from situation features of $C_k$ and $C_j$; $sim^A(C_k (attrb_x), C_j (attrb_x))$ is the similarity value obtained from the values of attribute $x$; $w_T$ is the weight factor for the text description; $w_F$ is the weight factor for the situation feature; and $w_x$ is the weight given to attribute $x$. Note that the summation of $w_T$ , $w_F$ and all $w_x$ is equal to 1.

***Case-based reasoning for a target case.*** A target case is a situation that a worker is currently handling. After entering a target case $C_k$ of a situation, the system identifies an existing case identifier of $C_k$ or retrieves similar situation cases if $C_k$ is a new case. The similarity measures between the target case and previous cases are computed using Equation 16. The identification procedure is similar to the transformation procedure. Assume there are $r$ situation identifiers. Let $minsim(C_k, S_i)$ be the *minimum similarity*$(C_k, C_j)$ over all $C_j$ transformed into $S_i$. The procedure finds a situation identifier $S_f$ such that $minsim(C_k , S_f)$ is the *maximum* of

$minsim(C_k, S_i)$ over all $S_i$ (for $i$ = 1 to $r$). An existing situation identifier $S_f$ is identified if $minsim(C_k, S_f)$ is greater than $\theta$; otherwise, the situation is a new case and the system assigns a new identifier to it. The case and its identifier are then stored in the knowledge base, and CBR is initiated to retrieve similar cases based on their similarity measures and to suggest possible knowledge related to the similar cases.

### 5.2.2. Discovery of context-based situation profiles

A context-based situation profile is also represented as a term vector which is derived by analyzing the set of documents accessed for handling the situation case in certain context. Each document $d_j$ is pre-processed and represented as a term vector $\vec{d}_j$. Let $D_S^{CXT}$ denote the set of documents accessed to handle the situation $C_S$ in certain context $CXT$. A centroid approach is used to derive the profiling term vector $\vec{P}_S$ of $C_S$ by averaging the term vectors of documents in $D_S^{CXT}$. Equation 17 defines the weight of a term $k_i$ in $\vec{P}_S$.

$$w(k_i, C_S) = \frac{1}{\left|D_S^{CXT}\right|} \sum_{d_j \in D_S^{CXT}} w(k_i, d_j). \tag{17}$$

***Retrieval of situation relevant documents.*** The system recommends/retrieves relevant knowledge documents to help workers solve problems based on context-based situation profiles. The key contents of a codified knowledge document are represented as a term vector. The situation profile of a case $C_S$ is expressed as a profiling term vector $\vec{P}_S$. The cosine measure of term vectors, described in Section 2.3, is used to derive the similarity measure. Let $\vec{d}_j$ be the term vector of document $d_j$. The cosine measure of $\vec{P}_S$ and $\vec{d}_j$, cosine($\vec{P}_S, \vec{d}_j$), is the similarity measure between the situation and document $d_j$. Documents with the top-N similarity measures are selected as relevant documents.

### 5.2.3. Discovery of context-based knowledge patterns

Recall that a generic problem-solving process is specified by experts to solve a problem. The specification includes the stages and their execution order. This work focuses on the execution of a sequence of stages. For example, the generic water supply problem-solving process is “Normal Management Stage (NM Stage) → Engineering Improvement Stage (EI Stage) → Exception Management Stage (EM Stage) → Quality Improvement Stage (QI

Stage) → Maintenance Management Stage (MM Stage). For any given problem, the situations may vary; thus the follow-up actions may also vary.

***Discovery of context-based decision-making knowledge patterns.*** The decision-making knowledge patterns discovered from previous system framework indicate the frequent association of situations and actions in a problem-solving process. Different from decision-making knowledge patterns, context-based decision-making knowledge patterns indicate the inferred associations of actions and situation features and actions in certain context of the problem-solving process. In specific stage, based on situation features and relevant context characteristics, the system continually infers situation features to form context-based decision-making knowledge pattern in certain context of specific problem-solving stage, as described in Section 5.2. Fig. 12 illustrates an example of the context-based decision-making knowledge pattern.

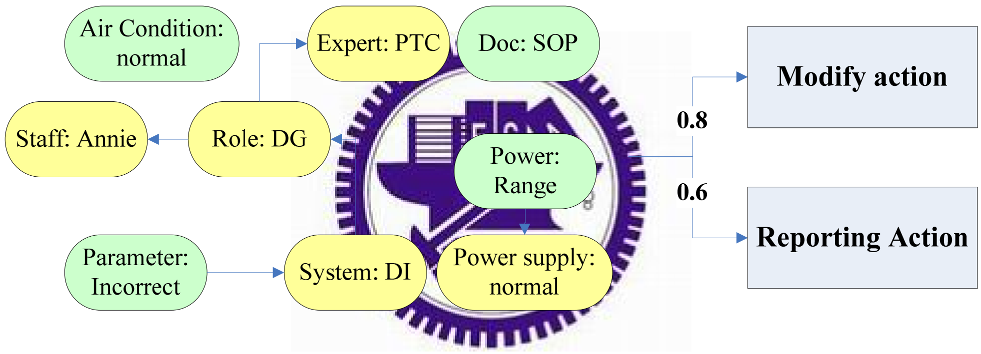


Fig. 12: An example of context-based decision-making knowledge pattern

The collected attributes: Air Condition (normal), Parameter (Incorrect), Power Status (12V), Doc(SOP) and inferred situation feature: Staff (Annie), Expert (PTC), Role (DG), System(DI), Power supply (normal) are used to discover the context-based inference rules that indicate taking modify action with *CF* value: 0.8 and taking reporting action with *CF* value: 0.6. The discovered inference rule forms the rule pattern considered as the context-based decision-making knowledge pattern.

***Deriving the score of action.*** To recommend reasonable actions for specific situation in certain context, we use the weighted linear combination of acquired context-based knowledge, including the similarity of situation cases, the confidence of context-based decision-making pattern, and the *CF* value of inferred action. Equation 18 defines the scoring method of context-based decision-making knowledge pattern of specific situation-action. Note that the actions in the context-based decision-making patterns (i.e., situation => action) whose left-hand side match the current situation are suggested and ranked according to the scoring values of the context-based decision-making knowledge patterns.

$$Socre(A_j) = w_s Sim(S_i, S_t) \times Conf(S_i \rightarrow A_j) + w_{\inf} CF(A_j^{ctx-\inf}). \tag{18}$$

where $A_j$ represents the action $j$; $S_i$ indicates the situation $i$; $S_t$ indicates the target situation $t$; the $Sim(S_i, S_t)$ represents the similarity value of situation $S_i$ and $S_t$; $Conf(S_i \rightarrow A_j)$ indicates the confidence of decision-making knowledge pattern $S_i \rightarrow A_j$; $CF(A_j^{ctx-\inf})$ indicates the certainty factor value of inferred action $A_j^{ctx-\inf}$ derived from context-based inference rules; $w_s$ is the weight factor for $Sim(S_i, S_t) \times Conf(S_i \rightarrow A_j)$; $w_{inf}$ is the weight factor for the $CF(A_j^{ctx-\inf})$. Note that the summation of $w_s$ and $w_{inf}$ is equal to 1.

***Discovery of context-based dependency knowledge patterns.*** For a specific problem-solving process, the dependency knowledge patterns discovered from previous system framework express the frequent relationships between situations and actions across different stages. Different from dependency knowledge pattern, context-based dependency knowledge patterns indicate the inferred relationships between situation/action features in current stage and situations/actions across different stages of the whole problem-solving process context. The discovered context-based inference rule may involve with several stages. The system uses its situation features as the seeds to infer situation features in relevant stages. The inferred situation features of relevant stages form the context-based dependency knowledge pattern. Fig. 13 illustrates the example of the context-based dependency knowledge patterns.

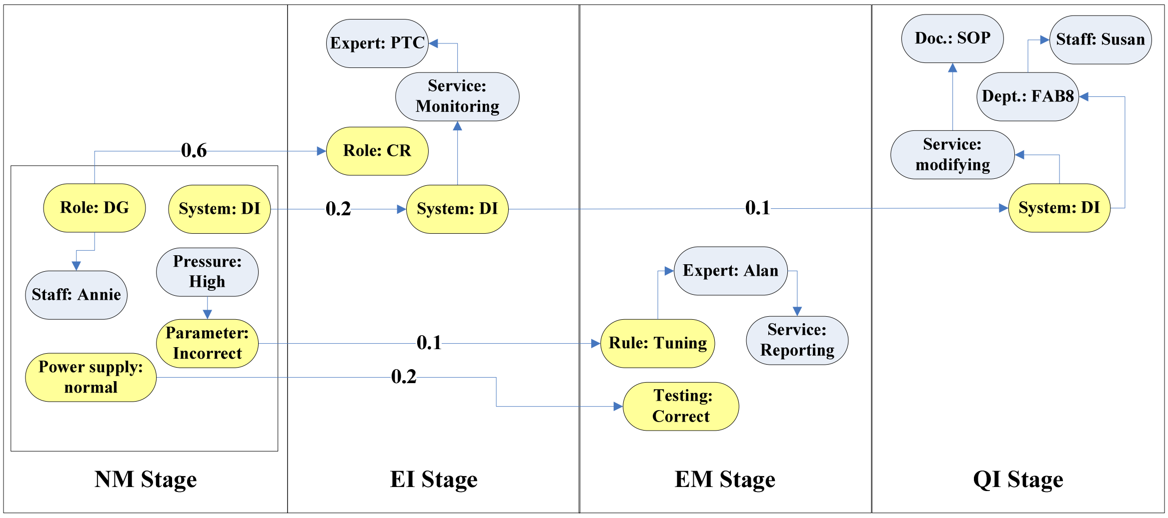


Fig. 13: An example of context-based dependency knowledge pattern

In pipe abnormal situation of Normal Management stage, the collected situation features (e.g., Role: DG; System: DI; Water supply pressure: High; Power supply service: normal) are used to infer the intra situation features (e.g., Staff: Annie; Pipe system parameter: Incorrect) based on relevant context-based inference rule (e.g., [Role(DG)] → [Staff(Annie)] and [pipe pressure(High)] → [System parameter(Incorrect)]). The collected and inferred situation features are used to infer the inter situation features in different stages (e.g., Role: CR, System: DI in Engineering Improvement stage and Rule: Tuning, Service Testing: Correct in Exception Management stage). The inferred situation features are used to continually infer intra and inter situation features in current and different stages. The inferred situation features and relevant context-based inference rules form the context-based dependency knowledge patterns.

***Context-based situation profiles and relevant documents.*** The context-based situation profiles are generated from the accessed documents in certain context, as described in section 5.2.2. For example, in the situation of abnormal water quantity, the accessed documents include: "DI analytical machine water quantity recording" and "DI GCHC machine water quantity recycling." The relevant context information includes Location: 8C; System: DI; System: DI; Service: Water Supply Service. The context-based situation profile is generated from the accessed documents in certain context. Once a worker encounters a problem situation or decides to take a particular action, the system provides relevant documents as knowledge support based on the context-based situation profiles. Fig. 14 illustrates a context-based situation profile and the relevant documents for the water supply problem-solving process.

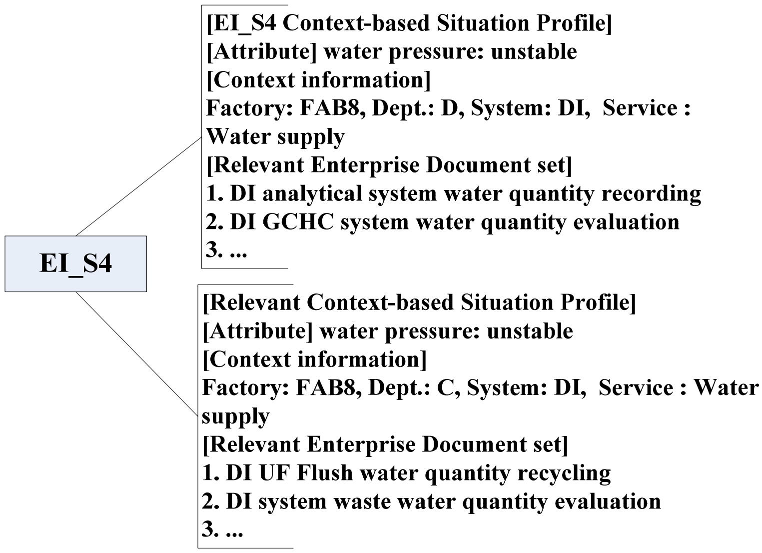


Fig. 14: Relevant context-based situation profile of situation EI_S4

Based on the context-based situation profiles, the system gathers previous and new relevant documents, such as “DI analytical machine water quantity recording” and “DI GCHC machine water quantity recycling” and new documents “DI system waste water quantity evaluation” and “DI UF Flush water quantity recycling”.

### 5.2.4. Knowledge recommendation

The proposed system suggests relevant documents according to the context-based situation profile of the current situation or similar cases, as shown in Fig. 14. The system also recommends relevant action documents (e.g., operating procedures and guidelines) according to the action profile, as shown in Fig. 15. Note that the top-N relevant documents are recommended according to the cosine measure of the term vectors of the documents and the context-based situation profiles, as described in Section 5.2.2.

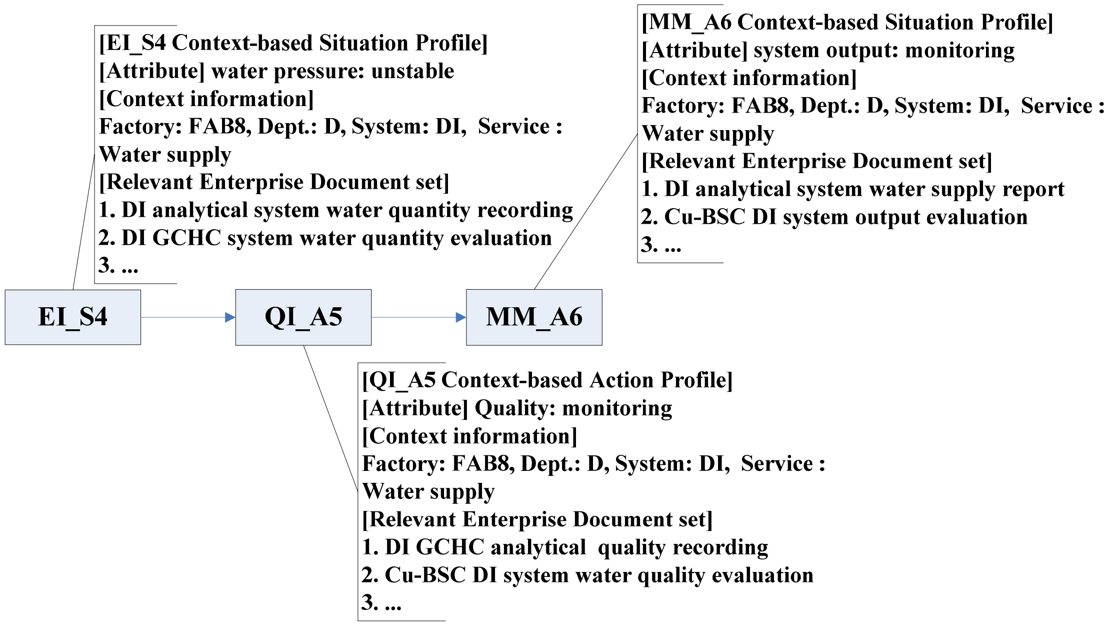


Fig. 15: Context-based situation profiles and relevant documents of the pattern

Moreover, the system suggests possible actions for handling the current situation according to the context-based decision-making knowledge patterns. When a situation matches a specific context-based knowledge pattern, the inferred situation features and relevant context-based inference rules in certain stages will be suggested as knowledge support. Furthermore, the context-based dependency knowledge patterns also denote the chain reaction across different stages. This helps workers plan appropriate actions for different problem-solving stages.

As the example of Fig. 16, the context-based knowledge patterns are in a chain reaction across different stages. In normal management stage, based on situation features collected by system, inferred situation feature, and scoring mechanism described in Section 5.2.3, the context-based decision-making knowledge patterns suggest the workers that pipe abnormal situation $\rightarrow$ Monitoring the output action with score 0.006 and pipe abnormal situation $\rightarrow$ Reporting the outcome action with score 0.0021 under the context consideration. The context-based dependency knowledge patterns also provide workers inferred situation features and relevant context-based inference rules as knowledge support to plan appropriate actions for different problem-solving stages

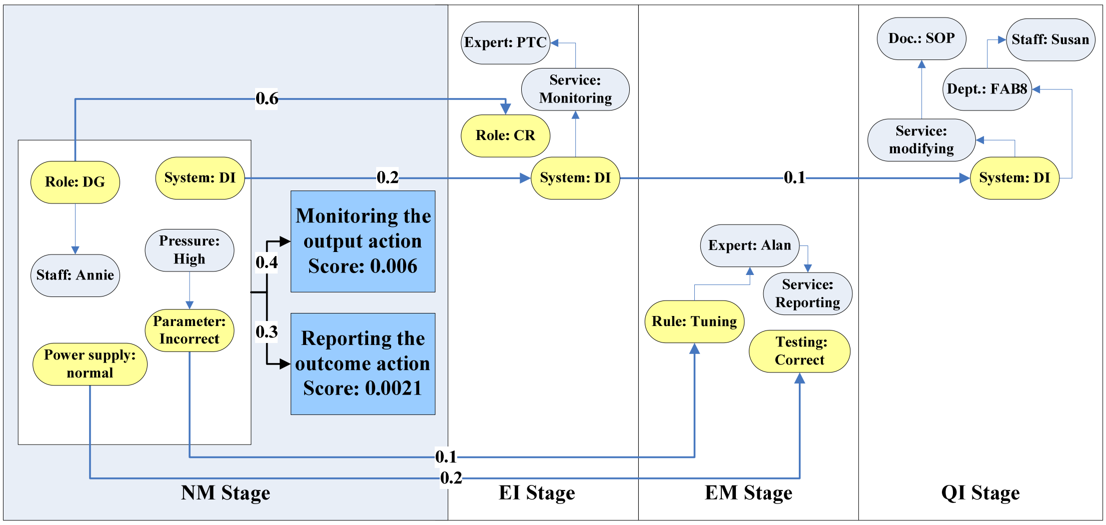


Fig. 16: Context-based knowledge patterns in a chain reaction across different stages

## 5.3. System implementation

In this section, we illustrated a system implementation to demonstrate the effectiveness of context-based rule inference. The implementation is conducted using several software tools, including the Eclipse Version 3.2 Software Development Kit (SDK) and Java(TM) 2 Platform Standard Edition Runtime Environment (J2SE) Version 5.0. The system function uses Drools of JBoss Rules Version 3.04 which is the plugin of Eclipse as the inference engine to get *Certainty Factor* (*CF*) value and infer situation feature. Microsoft SQL Server 2000 is used as the database system for storing data related to the problem-solving process and codified knowledge documents. The data mining tool Weka 3.4 is used to discover context-based inference rules in the historical problem-solving log.

The system function shows relevant problem-solving information collected in knowledge base, including problem-solving process, stage, situation/action, context-based inference rule with confidence and support value. Based on collected problem-solving information, the system function enforces the inference process and shows the inferred knowledge.

For example, the system function gathers the relevant problem-solving information, including current problem-solving process: Water Supply Problem-solving Process; current stage: Normal Management Stage; current situation: Controller Temperature Abnormal Situation; and context-based inference rule: Staff(Annie) → DI Water Supply System Service (Parameter: incorrect) with confidence 0.13 and support 0.02. The system interface and relevant problem-solving information are illustrated in Fig. 17.

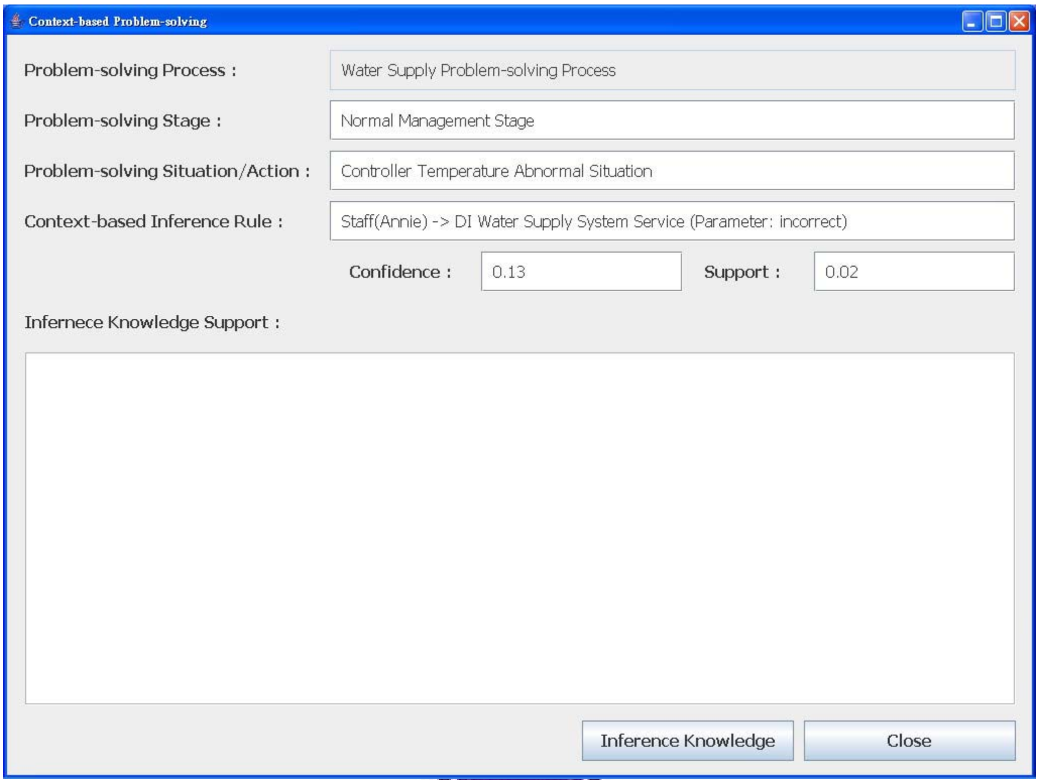


Fig. 17: A prototype system of context-based problem-solving knowledge support

After the inference process, the *CF* value of situation features are provided to worker, as shown in Fig. 18. According to the relevant information stored in enterprise knowledge base, the *CF* value of Staff(Annie) is "1", support of "DI Water Supply System Service(Parameter: incorrect)" is 0.1, the system enforces inference process (the details mentioned in Section 5.2.1) and gets the *CF* value of "DI Water Supply System Service(Parameter: incorrect)" is 0.033. The system function assists worker get the *CF* values of context-based inference rules and situation features in order to infer more situation features continually.

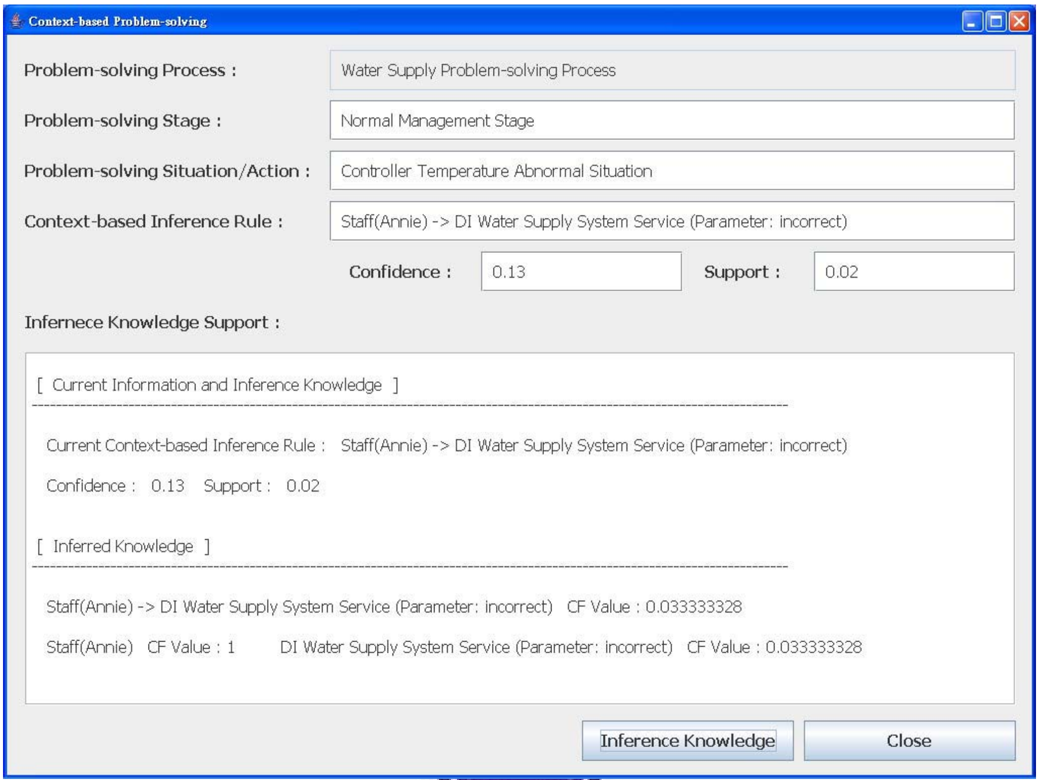


Fig. 18: Inference knowledge for the water supply problem

## 5.4. Discussions and comparisons

In this section, we discuss and compare the proposal knowledge support framework and context-based knowledge support.

### 5.4.1. Discussions

This work proposes a knowledge support system for problem-solving on a production-line. The descriptions of situation/action, attributes collected by system, and inferred situation features assist case-based reasoning in situation identification. Information Retrieval (*Automatic Indexing*) techniques are applied to discover the key terms of a situation. The terms form relevant situation profiles that model the information needs of workers to handle a problem. *Association rule mining* and *sequential pattern mining* techniques are used to discover decision-making and dependency knowledge patterns, and context-based inference rules. The context-based inference rules are used to infer more relevant situation features. This system discovers context-based decision-making and dependency knowledge based on context-based inference rules and inferred situation features. The situation profiles, discovered knowledge patterns, context-based inference rules, inferred features, and context-based situation profiles forms the basis to support problem-solving on a production line. Some issues or shortcomings of this framework are discussed as follows.

(1) The knowledge support system discovers relevant knowledge rules in order to provide knowledge support in a problem-solving process. However, the items of knowledge rules may involve with various types of data. For example, an attribute value may be nominal, binary, or numeric; the numeric attributes, a data discretization process is conducted to transform their values into value ranges or user-defined concept terms (such as *low*, *middle* or *high*). Therefore, in rule processing, the rule matching is an important issue that needs to be addressed.

(2) Based on CBR, data mining, and rule inference techniques, the context-based knowledge support system enforces context modeling to formalize the relevant situation features and context characteristics of a problem-solving process. The situation features and context characteristics are considers as the items of transaction in order to discover context-based inference rules. However, situation features and context characteristics in different context may have different importance. Therefore, the importance of various situation features and context characteristics should be considered in different levels of context modeling.

## 5.4.2. Comparisons

***Comparison to related work.*** We compare the proposed knowledge support system with related work, the details are illustrated as follows.

(1) Liao (2002) investigates the types of knowledge used for problem-solving and suggests the circulation of knowledge to avoid knowledge inertia. Although a knowledge-based architecture that incorporates case-based, rule-based, and heuristic-based approaches is proposed for managing problem-solving knowledge and dealing with knowledge inertia, the details of the system are not presented. In this work, the proposed system framework presents the procedures of knowledge discovery and recommendation processes. Moreover, the details of system implementation and real scenario are also illustrated clearly.

(2) Existing studies focus on using case-based reasoning to identify similar previous cases and derive a solution for a new case from previous problem solutions (Chang et al., 1996; Park et al., 1998; Guardati, 1998; Yang et al., 2004). In a complex production

process, problem-solving is usually knowledge intensive and requires effective knowledge support to provide workers with the necessary information to identify the causes of situations and taking appropriate action to solve them. However, identifying similar cases among previous problem cases is not sufficient to satisfy workers' information needs for solving a new problem. The required knowledge is usually hidden in various codified knowledge documents that must be proactively delivered to workers. The CBR approach does not provide such problem-relevant documents for knowledge-intensive problem solving. In this work, we adopt text mining (*Automatic Indexing*) techniques to compensate for the shortcomings of CBR technique.

(3) Problem-solving is the thought process that resolves various difficulties and obstacles spread in the gap between the current problem and its desired solution. (Heh, 1999). Problem-solving process includes a series of uncertain situations and operational actions. Moreover, situation features are usually occurred according to the context characteristics of problem. Due to the uncertain features of situations, several causes and possible solutions may exist for a specific situation. The causes and possible solutions are usually hidden in relevant data resources and difficult to extract. In such uncertain environments, situation features collected by system are usually partial or incomplete. Workers need to use knowledge gathered and inferred from relevant context information and previous problem-solving experience to clarify the causes and take appropriate action effectively. This work adopts rule inference techniques to consolidate the knowledge support for problem-solving.

***Comparison of two kinds of knowledge supports.*** The knowledge supports for problem-solving are separated into two parts. One uses the CBR and data mining techniques to provide knowledge support for problem-solving, which does not consider context characteristics; the other uses the CBR, data mining, and rule inference technique to provide knowledge support for problem-solving considering context characteristics. There are some comparisons illustrated as follows.

(1) The problem-solving process is a complex process that involves with wide scope of enterprise information and knowledge. In the knowledge support framework without the context consideration, using CBR and data mining techniques to process user descriptions and collected attributes may not be enough to support the problem-solving process. Based on the consideration of context, the context-based knowledge support

enforces CBR, data mining, and rule inference techniques to discover and infer more relevant knowledge, thus can help worker identify the certain situation and obtain relevant knowledge support effectively.

(2) The setting of minimum support and minimum confidence criteria may filter out some non-frequent but important clues of problem-solving process. Without context consideration, the system may derive very few decision-making and dependency knowledge rule patterns. Accordingly, worker may not obtain relevant knowledge support for certain situation/action. Based on collected context characteristics and inferred knowledge, the context-based knowledge support can infer more context-based knowledge to compensate the shortcoming of incomplete information and sparsity of rule patterns.

# Chapter 6. Conclusions and Future works

## 6.1. Summary

In this work, a novel knowledge support system has developed for problem-solving on a production-line. The description of situation/action and collected attributes assist case-based reasoning to identify similar situations/actions. Information Retrieval (*Automatic Indexing*) techniques are then applied to discover the key terms of a situation/action. The terms form situation/action profiles that model the information needed of workers to handle a problem. Association rule mining and sequential pattern mining are used to discover decision-making and dependency knowledge patterns, respectively. The situation/action profiles and discovered knowledge patterns are used to construct a knowledge support network, which forms the basis of support for solving problems on a production line. The proposed system provides integrated browsing and suggestions about problem-solving knowledge. Relevant documents are recommended to help users identify the root cause of a problem situation and the appropriate action to take. Workers can also use the knowledge support network to navigate the knowledge patterns and obtain decision-making and dependency knowledge. The proposed knowledge support network, enhanced with suggestions about problem-solving knowledge, provides workers with the necessary knowledge to effectively solve problems. A prototype system is implemented using a data set from a company's intranet portal, in which the log file contains a log of information for handling problems on the company's production line.

Moreover, based on context modeling, context-based inference rules are discovered to infer more relevant situation features. The description of situation, collected attributes, and inferred situation features assist case-based reasoning in situation identification. Information Retrieval (*Automatic Indexing*) techniques are then applied to discover the key terms of a situation. The terms form context-based situation profiles that model the information needs of workers to handle a problem. The system uses the ***context-based situation profile*** to gather existing and new relevant knowledge documents for situation in certain context. Furthermore, the system continually infers situation features to form context-based knowledge patterns which provide workers with relevant inferred knowledge, as well as decision-making and dependency knowledge.

## 6.2. Future works

In our future work, we will apply our proposed method to different data resources or other application domains. This work has focused on solving problems in stages in different situations with different actions. The stages need to be predefined by experts, which is the case with the company's production line. For other application domains, the stages may not be easy to define. Moreover, the stages investigated in this work are limited to a sequential order, rather than a combination of AND/OR parallelisms and sequences, as in a workflow system. Accordingly, a more flexible approach to address these issues would be worthy of further study.

# References


[1] Abecker, A., Bernardi, A., Maus, H., Sintek, M., & Wenzel, C. (2000). Information supply for business processes: coupling workflow with document analysis and information retrieval. Knowledge-Based Systems, 13(1), 271-284.

[2] Agostini, A., Albolino, S., De Michelis, G., De Paoli, F., & Dondi, R. (2003). Stimulating knowledge discovery and sharing. Proceedings of the International ACM Conference on Supporting Group Work, Sanibel Island, Florida, USA, November, 248-257.

[3] Agrawal, R., Imielinski, T., & Swami, A. N. (1993). Mining Association Rules between Sets of Items in Large Databases. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., USA, May, 207-216.

[4] Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. Proceedings of the VLDB conference, 407-419.

[5] Agrawal, R., & Srikant, R. (1995). Mining Sequential Patterns. Proceedings of the 11th International Conference on Data Engineering, Taipei, Taiwan, March, 3-14.

[6] Allen, J., Blaylock, N., & Ferguson, G. (2002). A Problem Solving Model for Collaborative Agents. Proceedings of the first international joint conference on Autonomous agents and multiagent systems, Bologna, Italy, July, 774-781.

[7] Baeza-Yates, R., & Ribeiro-Neto, B. (1999). Modern Information Retrieval, New York: The ACM Press.

[8] Chakrabarti, S., Dom, B., Agrawal, R. & Raghavan, P. (1997). Using taxonomy, discriminants, and signatures for navigating in text databases. Proceedings of the 23rd VLDB conference Athens, Greece.

[9] Chang, K. H., Raman, P., Carlisle, W. H., & Cross, J. H. (1996). A Self-improving helpdesk service system using case-based reasoning techniques. Computer in Industry, 30, 113-125.

[10] Chen, G. & Kotz, D. (2000). A Survey of Context-Aware Mobile Computing Research, tech. report TR2000-381, Dept. of Computer Science, Dartmouth College, Hanover, N.H.

[11] Chen, M.S., Park J. S., & Yu, P. S. (1996). Data Mining: An Overview from Database Perspective, IEEE Transactions on Knowledge and Data Engineering, 8(6), December.

[12] Davenport, T.H., & Prusak, L. (1998). Working knowledge: How organizations manage what they know, Boston MA: Harvard Business School Press.

[13] Davies, N.J., Duke, A., & Stonkus, A. (2003). OntoShare: Evolving ontologies in a knowledge sharing system, In Davies, NJ, Fensel, D. van Harmelen, F. (Eds.): Towards the semantic Web – Ontology-based knowledge management. Wiley, London UK, 161-176.

[14] Dey, A.K. (2001). Understanding and Using Context, Journal of Personal and Ubiquitous Computing, 5(1), 4–7.

[15] Fenstermacher, Kurt D. (2002). Process-Aware knowledge retrieval. Proceedings of the 35th Hawaii International Conference on System Sciences, Big Island, Hawaii, USA, 209-217.

[16] Georgalas, N. (1999). A Framework that uses Repositories for Information Systems and Knowledge Integration. Proceedings of the IEEE Symposium on Application-Specific Systems and Software Engineering and Technology, 128-135.

[17] Gray, P.H. (2001). The impact of knowledge repositories on power and control in the workplace. Information Technology & People, 14(4), 368-384.

[18] Guardati, S. (1998) RBCShell: A tool for the construction of systems with case-based reasoning, Expert Systems with Applications. 14, 63-70.

[19] Han, J. & Kamber, M. (2000). Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, USA.

[20] Heh, J. S. (1999). Evaluation Model of Problem Solving. Mathematical and Computer Modelling. 30, 197-211.

[21] Herlocker, J.L. & Konstan, J.A. (2001). Content-independent, task-focused recommendation. IEEE Internet Computing, 5(6), 40-47.

[22] Klemettinen, M., Mannila, H. & Toivonen, H. (1997). A Data Mining Methodology and Its Application to Semi-Automatic Knowledge Acquisition. Proceedings of the 8th International Workshop on Database and Expert Systems Applications, Washington, DC, USA, 670-681.

[23] Kohno, T., Hamada, S., Araki, D., Kojima, S., & Tanaka, T. (1997). Error repair and knowledge acquisition via case-based reasoning, Artificial Intelligence, 91, 85-101.

[24] Kwan, M.M., & Balasubramanian, P. (2003). KnowledgeScope: Managing knowledge in context, Decision Support Systems, 35, 467-486.

[25] Liao, S. H. (2002). Problem solving and knowledge inertia. Expert Systems with Applications, 22, 21-31.

[26] Liebowitz, J. (2001). Knowledge management and its link to artificial intelligence. Expert Systems with Applications, 20, 1-6.

[27] Liu D.-R., Wu I.-C., & Yang K.-S., (2005). Task-based K-Support System: Disseminating and Sharing Task-relevant Knowledge, Expert Systems with Applications, 29(2), August, 408-423.

[28] Middleton, S. E., Shadbolt, N. R., & De Roure, D. C. (2004). Ontological user profiling in recommender system. ACM Transactions on Information Systems, 22(1), 54-88.

[29] Park, M. K., Lee, I., & Shon, K. M. (1998). Using case reasoning for problem solving in a complex production process. Expert System with Application, 15, 69-75.

[30] Pazzani, M., & Billsus, D. (1997). Learning and revising user profiles: The identification of interesting Web sites. Machine Learning, 27, 313-331.

[31] Porter, M.F. (1980). An algorithm for suffix stripping. Program 14(3) 130-137.

[32] Ryan, N. (1997). Mobile Computing in a Fieldwork Environment: Metadata Elements,

Project working document, version 0.2.

[33] Salton, G., & Buckley, C. (1988). Term weighting approaches in automatic text retrieval. Information Processing & Management, 24(5), 513-523.

[34] Salton, G., & Lesk, M.E. (1971) Computer evaluation of indexing and text processing. in: G. Salton (Eds), The SMART Retrieval System: Experiments in Automatic Document Processing, PrenticeHall, Englewood Cliffs, NJ, 143-180.

[35] Schilit, B. & Theimer, M. (1994). Disseminating Active Map Information to Mobile Hosts, IEEE Network, 8(5), 22–32.

[36] Shortliffe, E. & Buchanan, B. (1975). A model of inexact reasoning in medicine, Mathematical Biosciences, 23, 351-379.

[37] Srikant, R., & Agrawal, R. (1996). Mining Sequential Patterns: Generalizations and Performance Improvements. Proceedings of the Fifth International Conference on Extending Database Technology (EDBT), Avignon, France, 3-17.

[38] Witten, I.H., Moffat, A., & Bell, T.C. (1999). Managing Gigabytes: Compressing and Indexing Documents and Images, 2nd edn., Morgan Kaufmann Publishers, Los Alto, USA.

[39] Yang, B. S., Han, T., & Kim, Y. S. (2004). Integration of ART-Kohonen neural network and case-based reasoning for intelligent fault diagnosis, Expert System with Applications, 26, 387-395.