Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 33 (2007) 147–161
Expert Systems with Applications
www.elsevier.com/locate/eswa

# Knowledge support for problem-solving in a production process: A hybrid of knowledge discovery and case-based reasoning

Duen-Ren Liu *, Chih-Kun Ke

*Institute of Information Management, National Chiao Tung University, 1001 Ta Hseuh Rd, Hsinchu 300, Taiwan*

**Abstract**

Problem-solving is an important process that enables corporations to create competitive business advantages. Traditionally, case-based reasoning 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 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 paper, we propose a mining-based knowledge support system for problem-solving. In addition to adopting case-based reasoning to identify similar situations and the action taken to solve them, the proposed system employs text mining (information retrieval) 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.


*Keywords:* Case-based reasoning; Data mining; Knowledge support; Problem-solving; Text mining

## 1. Introduction

Problem-solving is an important process that enables corporations to create competitive advantages, especially in the manufacturing industry. Case-based reasoning (CBR) techniques (Chang, Raman, Carlisle, & Cross, 1996; Kohno, Hamada, Araki, Kojima, & Tanaka, 1997; Park, Lee, & Shon, 1998; Yang, Han, & Kim, 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 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. Due to the uncertain characteristics 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

* Corresponding author. Tel.: +886 3 5712121x57405.
*E-mail address:* dliu@iim.nctu.edu.tw (D.-R. Liu).

0957-4174/$ - see front matter © 2006 Published by Elsevier Ltd.
doi:10.1016/j.eswa.2006.04.026

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, workers need to use knowledge gathered from relevant information and previous problem-solving experience to clarify the causes and take appropriate action. Thus, identifying similar cases through CBR is not sufficient to solve problems effectively. 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.

In this work, we propose a mining-based knowledge support system for problem-solving. Besides adopting case-based reasoning to identify similar situations and the action taken to solve them, we adopt text mining methods to compensate for the shortcomings of CBR. For specific situations or actions, 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 (IR) 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.

Moreover, 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. Finally, a prototype system is developed to demonstrate the effectiveness of the knowledge support network.

The remainder of this paper is organized as follows. Section 2 reviews related works on knowledge discovery and problem-solving. Section 3 introduces the proposed framework of knowledge support for problem-solving. Section 4 describes the discovery of knowledge patterns, including situation/action profiles and knowledge rules. The knowledge support network and its usage are discussed in Section 5. Section 6 presents an implementation of the prototype system. Finally, in Section 7, we present our conclusions and indicate the direction of future work.

## 2. Related work

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

### 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 (Klemettinen, Mannila, & Toivonen, 1997; Kohno 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, Dom, Agrawal, & Raghavan, 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, Bernardi, Maus, Sintek, & Wenzel, 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 (Fenstermacher, 2002; Kwan & Balasubramanian, 2003). 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, Shadbolt, & De Roure, 2004; Pazzani & Billsus, 1997). The profiling approach has also been adopted by some KMS' to enhance knowledge retrieval (Abecker et al., 2000; Agostini, Albolino, De Michelis, De Paoli, & Dondi, 2003; Davies, Duke, & Stonkus, 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, Wu, & Yang, 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, Blaylock, & Ferguson, 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 & Buckley, 1988). The *term transformation* steps, including case folding, stemming, and stop word removal, are performed during text pre-processing (Porter, 1980; Salton & Lesk, 1971; Witten, Moffat, & Bell, 1999). Then, *term weighting* is applied to extract the most discriminating terms (Baeza-Yates & Ribeiro-Neto, 1999). Let $d$ be a codified knowledge item (document), and let $\vec{d} =< w(k_1,d), w(k_2,d), \ldots, w(k_n,d) >$ 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* (Porter, 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 Eq. (1):

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

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.

#### 2.3.1. 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 Eq. (2). The degree of similarity is higher if the cosine similarity is close to 1.0.

$$sim(x,y) = \text{cosine}(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{|\vec{x}||\vec{y}|}. \tag{2}$$

### 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, Park, & Yu, 1996; Han & Kamber, 2000). We adopt association rule mining and sequential pattern mining to extract knowledge patterns from previous problem-solving instances.

#### 2.4.1. Association rules mining

Association rule mining tries to find an association between two sets of products in a transaction database. Agrawal, Imielinski, and Swami (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 $Y$.

#### 2.4.2. 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; Agrawal & Srikant, 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.

## 3. The system framework of knowledge support for problem-solving

In this section, we describe the proposed system framework, including the concepts of the problem-solving process, the knowledge required for problem-solving, and the proposed knowledge support framework. 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 predetermined 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

#### 3.2.1. 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 30–32 °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.

#### 3.2.2. 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.

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. 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.

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.

#### 3.3.1. 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.

#### 3.3.2. 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.

- *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

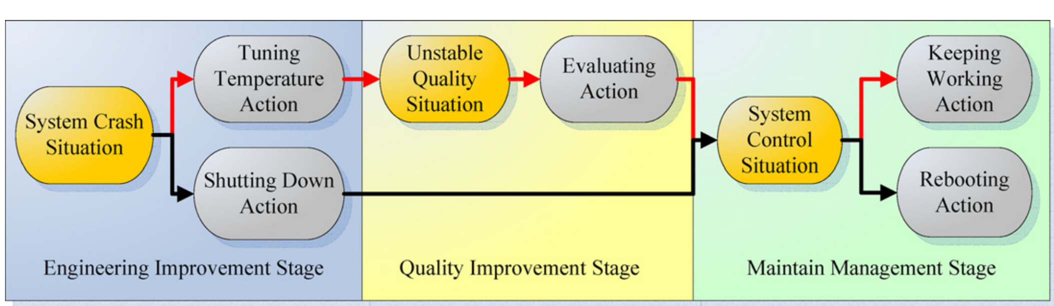


Fig. 1. A problem-solving process for a production line.

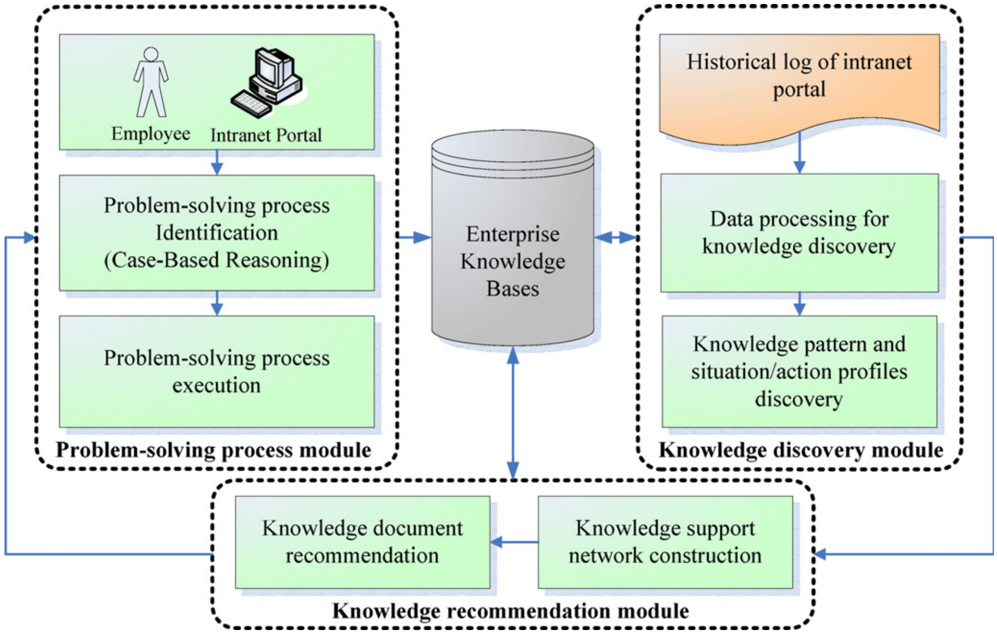


Fig. 2. Knowledge support framework for problem-solving.

target problem. Information retrieval (IR) and text mining 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.

#### 3.3.3. 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 5.

## 4. 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.

### 4.1. 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 case-based reasoning. The production process, problem-solving process, and the term vectors of accessed documents are integrated into the enterprise's knowledge base.

#### 4.1.1. 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. 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

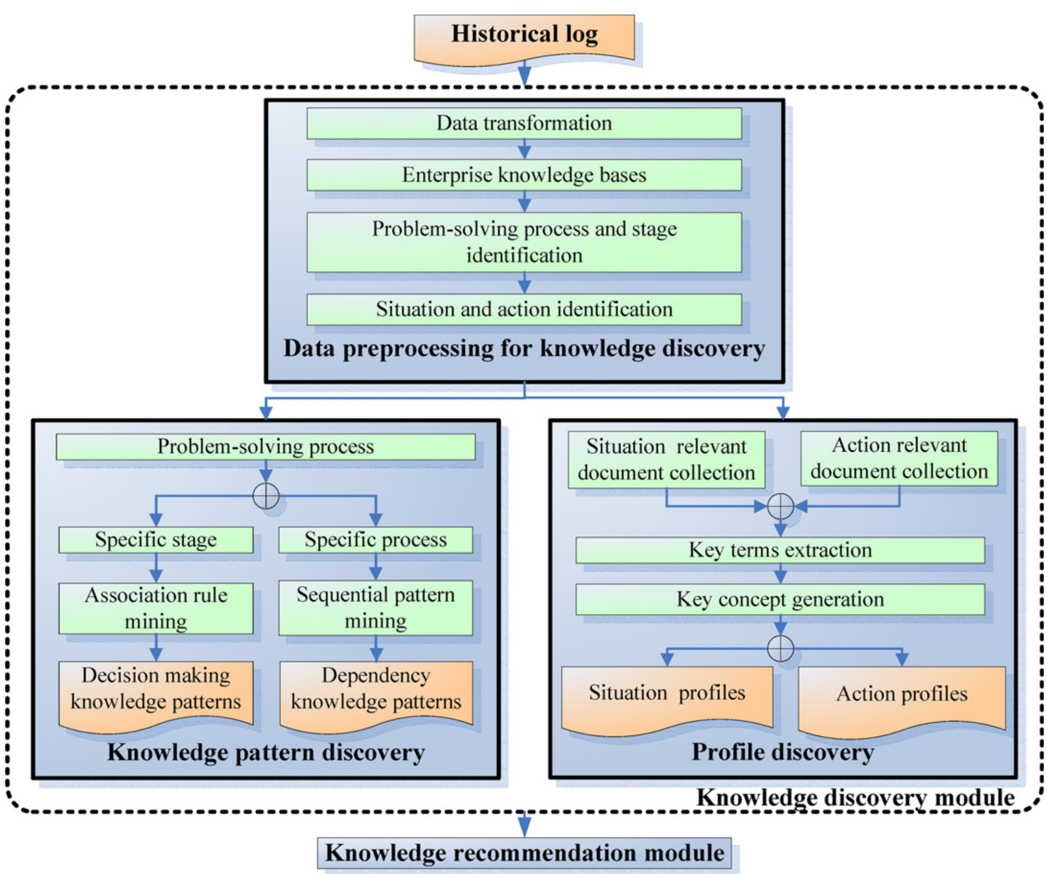


Fig. 3. The procedure of the knowledge discovery process.

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.

#### 4.2.1. 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 Eq. (3).

$$w(t_i, C_j) = \begin{cases} 1 & \text{if } t_i \in T_j, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Eq. (4) 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 $C_j$.

$$sim^T(C_k, C_j) = \text{cosine}(\vec{C}_k, \vec{C}_j) = \frac{\vec{C}_k \cdot \vec{C}_j}{|\vec{C}_k||\vec{C}_j|}. \tag{4}$$

#### 4.2.2. 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*). Eq. (5) 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.

$$sim^A(C_k(attrb_x), C_j(attb_x)) = \begin{cases} 1 & \text{if value } (C_k(attrb_x))\text{equals value}(C_j(attb_x)), \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

#### 4.2.3. Similarity function for case-based reasoning

Eq. (6) 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.

$$similarity(C_k, C_j) = w_T sim^T(C_k, C_j) + \sum_{x=1}^{m} w_x sim^A(C_k(attrb_x), C_j(attb_x)), \tag{6}$$

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.

#### 4.2.4. 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 Eq. (6). 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.

#### 4.2.5. 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 Eq. (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

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 |
| ... | ... |

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.

### 4.3. 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 Eq. (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 $\vec{\boldsymbol{P}}_S$ of $C_S$ by averaging the term vectors of documents in $D_S$. Eq. (7) defines the weight of a term $k_i$ in $\vec{\boldsymbol{P}}_S$.

$$\boldsymbol{w}(\boldsymbol{k}_i, \boldsymbol{C}_S) = \frac{\boldsymbol{1}}{|\boldsymbol{D}_S|} \sum_{\boldsymbol{d}_j \in \boldsymbol{D}_S} \boldsymbol{w}(\boldsymbol{k}_i, \boldsymbol{d}_j). \tag{7}$$

#### 4.3.1. 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 $\vec{\boldsymbol{P}}_S$. 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 $\vec{\boldsymbol{P}}_S$ and $\vec{d}_j$, cosine($\vec{\boldsymbol{P}}_S, \vec{d}_j$), is the similarity measure between the situation/action and document $d_j$. Documents with the top-N similarity measures are selected as relevant documents.

### 4.4. Discovery of knowledge patterns

#### 4.4.1. 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. We note that the instances of the water supply problem-solving process can be extracted from the log file.

#### 4.4.2. 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.

#### 4.4.3. 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.

## 5. 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 illustrated in Fig. 4.

### 5.1. 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.

#### 5.1.1. Specification of a generic problem-solving process

The specification includes the problem description, the stage names, and their execution orders.

#### 5.1.2. 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.

#### 5.1.3. 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.

#### 5.1.4. Situation/action profiles and relevant documents

The situation/action profiles are generated from the accessed documents, as described in Section 4.3. 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".

### 5.2. 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.

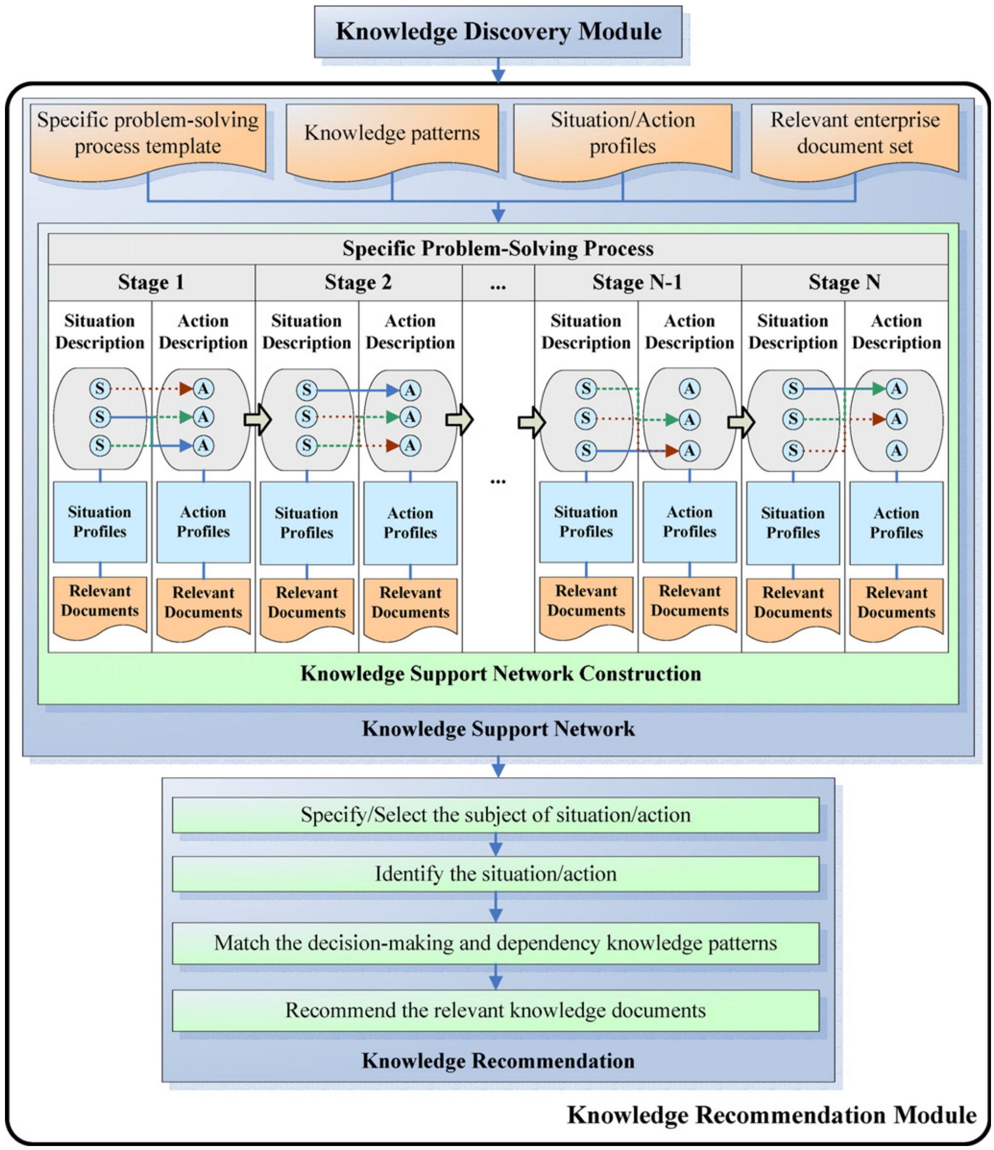


Fig. 4. The procedures of knowledge recommendation.

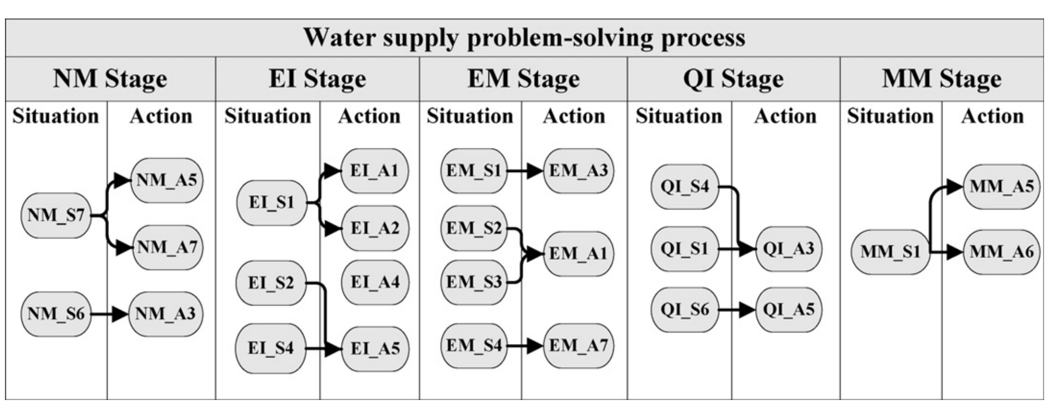


Fig. 5. Decision-making knowledge patterns in a knowledge support network.

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 4.3.

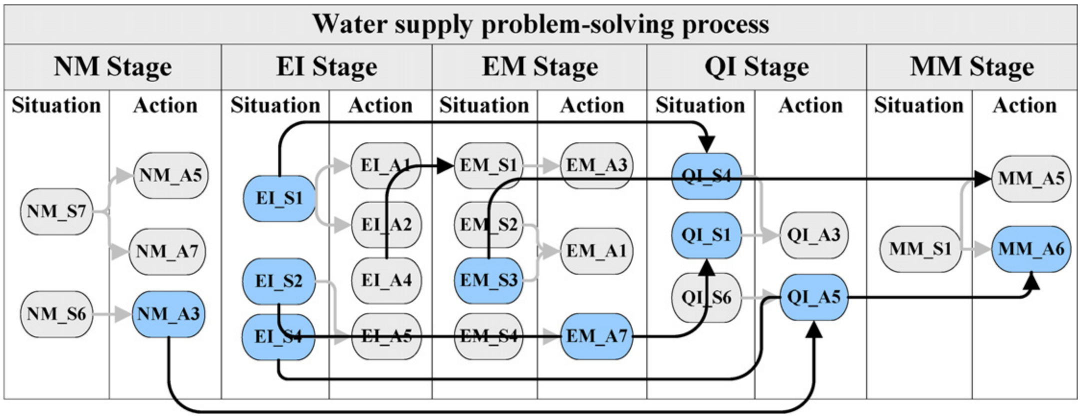


Fig. 6. Dependency knowledge pattern in a KSN.

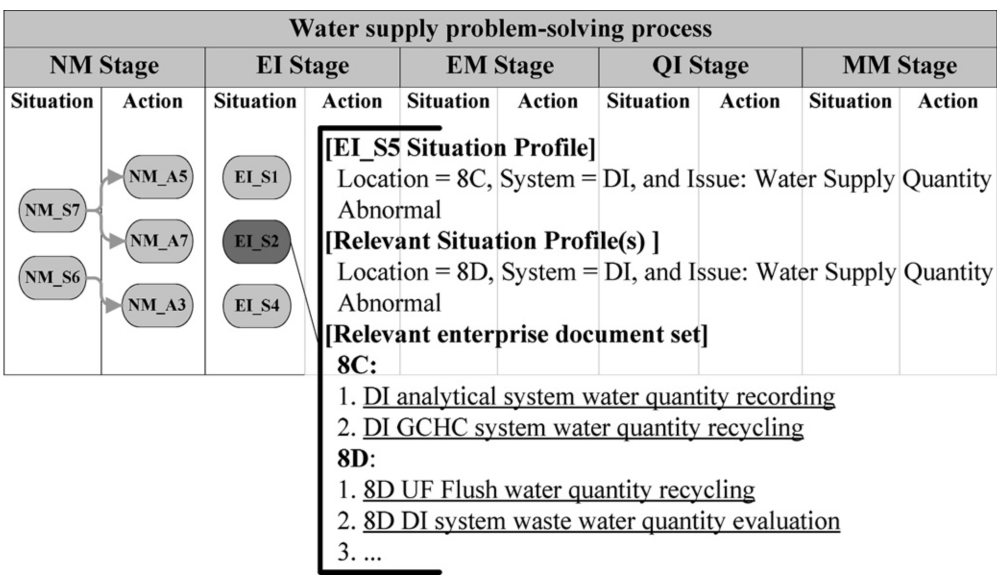


Fig. 7. Situation profile and relevant documents.

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.

## 6. 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 Page, 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.

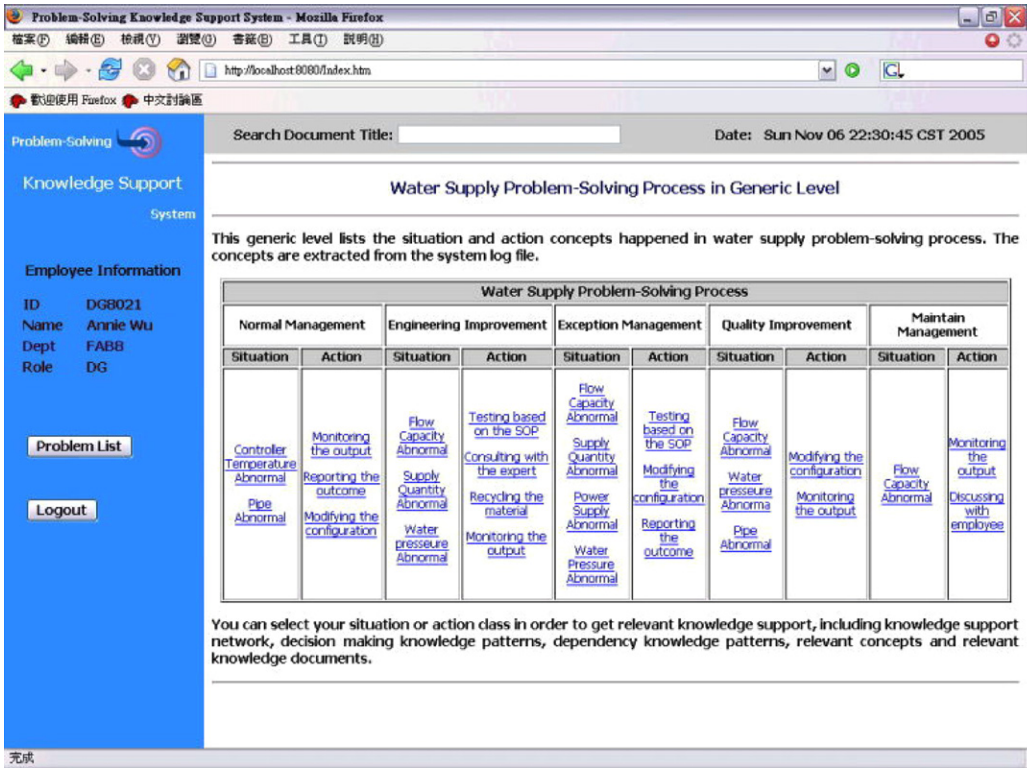


Fig. 8. A generic water supply problem-solving process.

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

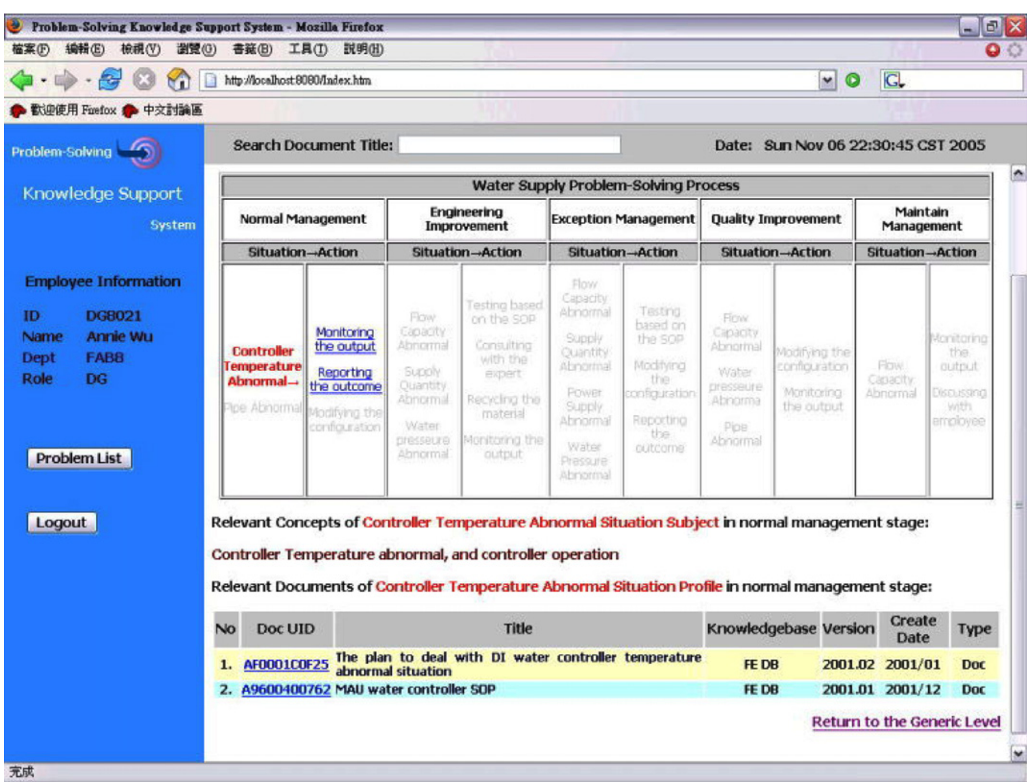


Fig. 9. Decision-making knowledge patterns for the water supply problem.

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.

## 7. Conclusion

In this work, we have developed a novel knowledge support system for problem-solving on a production-line. Case-based reasoning is used to identify similar situations/actions. Text mining techniques are then applied to discover the key terms of a situation/action. The terms form situation/action profiles that model the information needed to handle a problem. Association rule mining and sequential pattern mining are used to discover decision-making and dependency knowledge patterns, respectively. The discovered situation/action profiles and 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.

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.

## Acknowledgement

This research was supported in part by the National Science Council of the Taiwan (Republic of China) under the grant NSC 94-2416-H-009-015.

## References

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.

Agostini, A., Albolino, S., De Michelis, G., De Paoli, R., & Dondi, R. (2003). Stimulating knowledge discovery and sharing. In *Proceedings of the international ACM conference on supporting group work*, Sanibel Island, Florida, USA, November (pp. 248–257).

Agrawal, R., Imielinski, T., & Swami, A.N. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on management of data*, Washington, DC, USA, May (pp. 207–216).

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the VLDB conference* (pp. 407–419).

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the 11th international conference on data engineering*, Taipei, Taiwan, March (pp. 3–14).

Allen, J., Blaylock, N., & Ferguson, G. (2002). A problem solving model for collaborative agents. In *Proceedings of the first international joint conference on autonomous agents and multiagent systems*, Bologna, Italy, July (pp. 774–781).

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. New York: The ACM Press.

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

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.

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).

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

Davies, N. J., Duke, A., & Stonkus, A. (2003). OntoShare: evolving ontologies in a knowledge sharing system. In N. J. Davies, D. Fensel, & F. van Harmelen (Eds.), *Towards the semantic Web—ontology-based knowledge management* (pp. 161–176). London UK: Wiley.

Fenstermacher, K. D. (2002). Process-Aware knowledge retrieval. In *Proceedings of the 35th Hawaii international conference on system sciences*, Big Island, Hawaii, USA (pp. 209–217).

Georgalas, N. (1999). A framework that uses repositories for information systems and knowledge integration. In *Proceedings of the IEEE symposium on application-specific systems and software engineering and technology* (pp. 128–135).

Gray, P. H. (2001). The impact of knowledge repositories on power and control in the workplace. *Information Technology and People, 14*(4), 368–384.

Guardati, S. (1998). RBCShell: a tool for the construction of systems with case-based reasoning. *Expert Systems with Applications, 14*, 63–70.

Han, J., & Kamber, M. (2000). *Data mining: Concepts and techniques*. USA: Morgan Kaufmann Publishers.

Heh, J. S. (1999). Evaluation model of problem solving. *Mathematical and Computer Modelling, 30*, 197–211.

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

Klemettinen, M., Mannila, H., & Toivonen, H. (1997). A data mining methodology and its application to semi-automatic knowledge acquisition. In *Proceedings of the 8th international workshop on database and expert systems applications*, Washington, DC, USA (pp. 670–681).

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.

Kwan, M. M., & Balasubramanian, P. (2003). KnowledgeScope: managing knowledge in context. *Decision Support Systems, 35*, 467–486.

Liao, S. H. (2002). Problem solving and knowledge inertia. *Expert Systems with Applications, 22*, 21–31.

Liebowitz, J. (2001). Knowledge management and its link to artificial intelligence. *Expert Systems with Applications, 20*, 1–6.

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), 408–423.

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.

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.

Pazzani, M., & Billsus, D. (1997). Learning and revising user profiles: the identification of interesting Web sites. *Machine Learning, 27*, 313–331.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–137.

Salton, G., & Buckley, C. (1988). Term weighting approaches in automatic text retrieval. *Information Processing and Management, 24*(5), 513–523.

Salton, G., & Lesk, M. E. (1971). Computer evaluation of indexing and text processing. In G. Salton (Ed.), *The SMART retrieval system: Experiments in automatic document processing* (pp. 143–180). Englewood Cliffs, NJ: Prentice-Hall.

Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: generalizations and performance improvements. In *Proceedings of the fifth international conference on extending database technology (EDBT)*, Avignon, France (pp. 3–17).

Witten, I. H., Moffat, A., & Bell, T. C. (1999). *Managing gigabytes: Compressing and indexing documents and images* (2nd ed.). Los Alto, USA: Morgan Kaufmann Publishers.

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.