Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers in Industry 56 (2005) 428–441

COMPUTERS IN INDUSTRY

www.elsevier.com/locate/compind

# Application of ART neural network to development of technology for functional feature-based reference design retrieval

Chin-Bin Wang [a], Yuh-Jen Chen [b], Yuh-Min Chen [b,*], Hui-Chuan Chu [c]

[a] *Department of Information Management, Nan-Hua University, Chia-Yi, Taiwan, ROC*
[b] *Institute of Manufacturing Engineering, National Cheng Kung University, Tainan, Taiwan, ROC*
[c] *National University of Tainan, Tainan, Taiwan, ROC*

Received 22 December 2003; accepted 23 December 2004
Available online 14 March 2005

**Abstract**

Engineering design is a knowledge intensive process. The execution of each task in the process requires various aspects of knowledge and experience. Therefore, organizing, storing and retrieving product design information, design intents and underlining design knowledge is one of the most important tasks in engineering knowledge management.

This study develops a novel scheme for functional feature-based reference design retrieval using adaptive resonance theory (ART1) neural network to provide engineering designers with easy access to relevant design and other knowledge. This retrieval process includes the steps of functional feature-based query, case searching, and case ranking. The technology involves a binary code-based representation for functional features, ART1 neural network for functional feature-based case clustering, functional feature-based case similarity ranking, and a case-based representation for designed entities.

The objective of this study can be achieved by performing the following tasks: (i) designing a functional feature-based reference design retrieval process, (ii) developing a functional feature representation, (iii) investigating ART1 neural network, (iv) implementing a functional feature-based reference design retrieval mechanism, and (v) experimenting with functional feature-based case clustering.


*Keywords:* ART1 neural network; Knowledge retrieval; Knowledge management; Engineering design

## 1. Introduction

With the advent of the knowledge economy, knowledge has become the asset for enterprises in the 21st century. Whether enterprise knowledge can be effectively organized, stored and shared is a key factor

* Corresponding author. Tel.: +886 6 2757575x63922; fax: +886 6 2085334.
*E-mail address:* ymchen@mail.ncku.edu.tw (Y.-M. Chen).

0166-3615/$ – see front matter 
doi:10.1016/j.compind.2004.12.004

for enterprise success. Consequently, effectively organizing, storing and sharing knowledge to boost business intelligence (BI) is crucial for enterprises in the knowledge economy age.

Engineering design [12,21] is the process of establishing requirements based on customer needs, transforming them into performance specifications and functions, and then mapping those specifications and functions and converting them into design solutions that can be economically manufactured and produced based on creativity, scientific principles, and technical knowledge.

Engineering design, a knowledge intensive process, includes the tasks of conceptual design, detailed design, engineering analysis, assembly design, process design, and performance evaluation. Each task is conducted using various aspects of knowledge and experience. Whether “the knowledge and experience can be organized, stored and effectively retrieved is a major determining factor in increasing product development capability and quality and reducing development cycle time and cost. Therefore, organizing, storing, and retrieving product design information, design intents and underlining design knowledge are the basis of and also one of the most important tasks in engineering knowledge management.

Recently, information retrieval approach/system development has focused on retrieving documents related to a user query while retrieving as few irrelevant documents as possible. To pursue the above goal, numerous studies on information retrieval have been developed from various aspects, including modeling [18,25,26], document classification and categorization [1,3,6–9], system architecture [22], user interface [19,24], data visualization [16,20], filtering [2,13–15,23], and language [18]. However, these approaches in the studies were unsuitable to solve the functional feature-based reference design retrieval since these two structural types of document and functional feature of the part are different. Moreover, we also discovered no effective and practical method/approach for retrieving related engineering knowledge in engineering design based on querying the levels of customer requirements, functional requirements, functional features, and engineering specifications. Therefore, this circumstance causes a bottleneck for sharing valuable product information and engineering knowledge in engineering design.

This study applies the ART1 neural network to realize a scheme for functional feature-based reference design retrieval to provide engineering designers with easy access to relevant reference information and knowledge. This objective can be achieved by performing the following tasks: (i) designing a functional feature-based reference design retrieval process, (ii) developing a functional feature representation, (iii) investigating ART1 neural network, (iv) implementing a functional feature-based reference design retrieval mechanism, and (v) experimenting with functional feature-based case clustering.

## 2. Functional feature-based reference design retrieval

This section first briefly presents a proposed engineering knowledge management framework. Subsequently, the process of functional feature-based reference design retrieval is described. Three main areas of the process of functional feature-based reference design retrieval then are explained, namely: (i) functional feature representation, (ii) ART1 neural network, and (iii) functional feature-based case retrieval by ART1. Each portion involves several important techniques. Techniques related to the representation of functional feature include the definition of functional features and binary code-based representation for functional features. For developing the ART1 neural network model, ART1 characteristics are identified first, followed by the ART1 architecture and algorithm. The development of a method for similar case retrieval involves techniques for case-based representation of a designed entity, case clustering, and case similarity ranking. These techniques pave the way for implementation a mechanism for functional feature-based reference design retrieval.

### *2.1. Engineering knowledge management framework*

This subsection presents an overview of a proposed engineering knowledge management framework for supporting knowledge intensive activities in engineering design [4]. From Fig. 1, the framework is illustrated by the knowledge management life cycle, which consists of engineering knowledge creation,

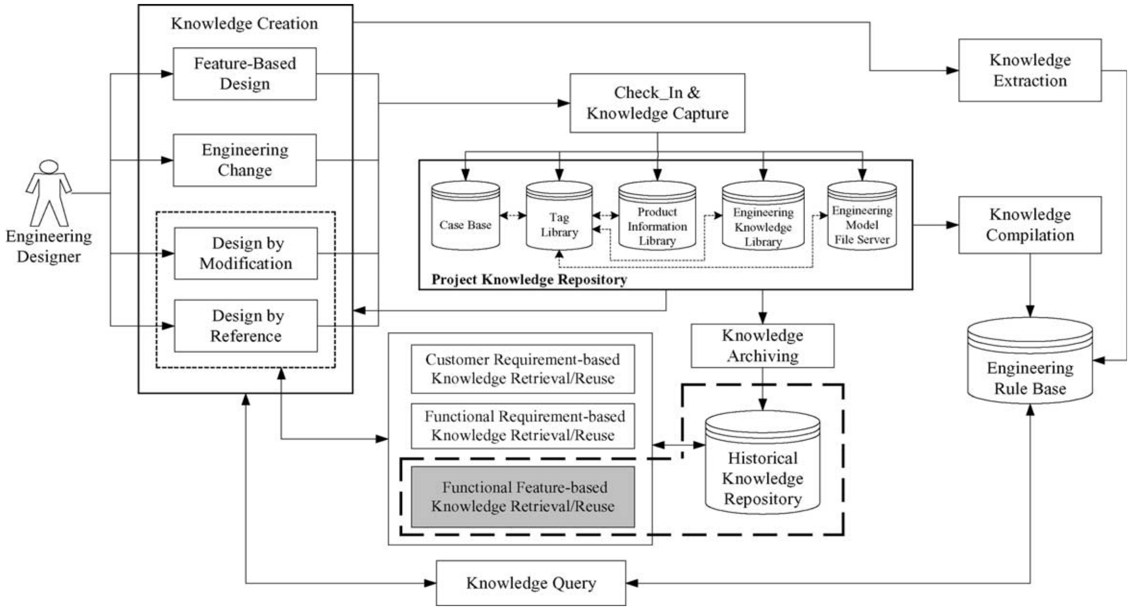


Fig. 1. Engineering knowledge management framework.

capture, compilation and storage, and retrieval/reuse/query.

In the proposed engineering knowledge management framework (Fig. 1), the knowledge retrieval part can be differentiated into three types, namely (1) customer requirement-based knowledge retrieval, (2) functional requirement-based knowledge retrieval, and (3) functional feature-based knowledge retrieval. This study primarily focuses on the functional feature-based knowledge retrieval, which is displayed as the shaded portion and surrounded by the broken line, as illustrated in Fig. 1.

## 2.2. Functional feature-based reference design retrieval process

This section details the process of functional feature-based reference design retrieval, which aims to retrieve the most similar cases as references from the historical knowledge repository according to the query of the users for functional features. To describe this retrieval process, the functional feature-based reference design retrieval process is designed using the simple and generic software architecture, as shown in Fig. 2. First, before the retrieval process can be initiated, the ART1 neural network must be trained and tested by training and testing historical samples of functional features. During the learning process of the ART1 neural network, the functional features in a historical case are represented as binary code. Once the learning process is completed, cases in the historical knowledge repository are clustered based on their functional features. The fact that the cases are clustered allows the retrieval process to be initiated. The engineering designer first specifies a set of functional features that are treated as binary variables. ART1 neural network for functional feature-based case clustering then must be applied before the actual cases acquisition, providing a case classifier for the functional feature query. This query is then processed through the ART1 neural network to acquire similar cases. Before being sent to the engineering designer, these similar cases are ranked based on the calculation of similarity coefficients. The engineering designer then examines the set of ranked cases to obtain useful information and knowledge.

## 2.3. Representation of functional features

Successfully utilizing the ART1 neural network in the functional feature-based case clustering requires first defining the functional features of parts. Subse-

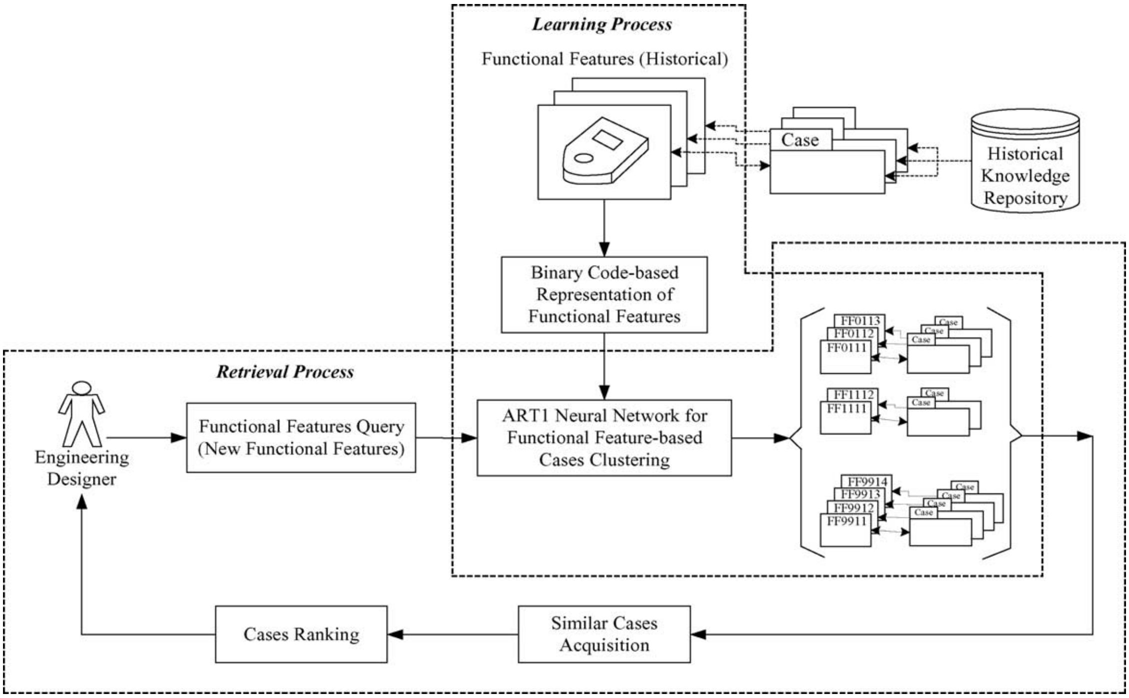


Fig. 2. Functional feature-based reference design retrieval process.

quently, a binary code-based representation is used to represent the defined functional features. The following subsections discuss the details.

### 2.3.1. Functional features definition

Functional feature identification is designed to define the functional features of a part and thus facilitate functional feature-based case clustering. By investigating studies on functional features in feature-based design, most functional features of a part are formed based on the feature interactions that depend on the spatial relationships between features [5]. Fig. 3 specifies the functional features formed by the feature interactions. The features can be classified into positive and negative features. Consequently, the feature interactions can be divided into three types, namely: (i) a positive feature and a positive feature, (ii) a positive feature and a negative feature, and (iii) a negative feature and a negative feature. The first type includes the relationships "adjacent_to" and "intersect". Meanwhile, the second type includes the relationships "add_on" and "is_in". Finally, the third type includes the relationships "adjacent_to", "is_in", and "intersect". Each type of feature interactions creates several functional features based on the specific relationship between features. For example, a positive feature and a negative feature may create the "hole",

| Interaction Type of Features | Relationship between Features | Functional Features |
|---|---|---|
| Positive Features and Positive Features | Adjacent_to | Convex |
| | Intersect | Square |
| | | Cylinder |
| Positive Features and Negative Features | Add_on | Protrusion |
| | Is_in | Hole |
| | | Groove/Slot |
| | | Step |
| Negative Features and Negative Features | Adjacent_to | Convex |
| | Is_in | Convex |
| | Intersect | Square |
| | | Cylinder |

Fig. 3. Typical functional features for a part.

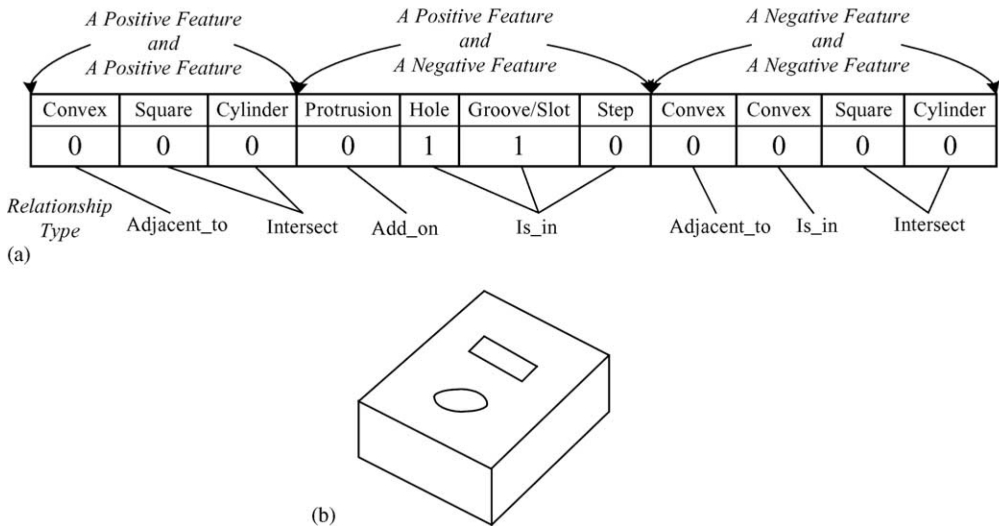


Fig. 4. (a) Binary code-based representation of functional features for the part shown below and (b) sample part.

‘‘groove/slot’’ or ‘‘step’’ function based on the relationship ‘‘is_in’’.

### 2.3.2. *Binary code-based functional features representation*

Parts are characterized using a list of functional features, which are treated as binary variables. From the previous subsection, these eleven functional features are required to record the specific part features. When coding a part based on the list, ‘‘one’’ means that the part has a given functional feature, while ‘‘zero’’ means that it does not.

Fig. 4(a) shows the binary code-based representation of functional features of the sample part displayed in Fig. 4(b). The part is represented by an 11-component vector ($X_1$, $X_2$, . . ., $X_{11}$). The components $X_1$–$X_3$ in the vector indicate the ‘‘convex’’, ‘‘square’’, and ‘‘cylinder’’ functions, which are formed by the first type of feature interactions. Furthermore, the components $X_4$–$X_7$ denote the ‘‘protrusion’’, ‘‘hole’’, ‘‘groove/slot’’, and ‘‘step’’ in an ordered sequence. They are generated through the second type of feature interactions. Finally, the last four components $X_8$–$X_{11}$ express the functional features of the third type of feature interactions, namely ‘‘convex’’, ‘‘convex’’, ‘‘square’’, and ‘‘cylinder’’. Therefore, the functional features involved in the sample part include the ‘‘hole’’ and ‘‘slot’’ functions, and are represented as ‘‘one’’.

## 2.4. *Adaptive resonance theory (ART1) neural network*

This study adopts the adaptive resonance theory (ART1) neural network to solve the problem of functional feature-based case clustering. The ART1 neural network can be defined in terms of ART1 characteristics, ART1 architecture, and ART1 algorithm, respectively. The details are presented below.

### 2.4.1. *ART1 characteristics*

Examination of neural networks reveals that most can either be plastic (during the learning phase) or stable (during recall, when the weights are frozen), but not both. To overcome the stability-plasticity dilemma faced by every learning system, the Adaptive resonance theory (ART1) neural network was proposed by Carpenter and Grossberg [10,11] and serves the purpose of cluster discovery through unsupervised learning. This theory includes the following characteristics:

- *Binary-based input vector*: ART1 is designed for binary 0/1 inputs, where each input vector may have more 0/1 elements.
- *Stability and plasticity*: The ART1 network is sufficiently stable to preserve significant past

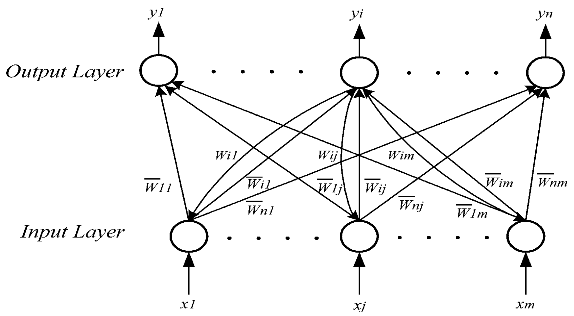


Fig. 5. Basic ART1 architecture.

learning but still remain sufficiently adaptable to incorporate new information (clusters) as necessary.

- *Unsupervised learning*: In unsupervised learning, no external teacher or critic oversees the learning process and provides feedback information. Moreover, no environmental feedback exists to indicate the nature or correctness of the outputs. The network must discover its own patterns, features, regularities, correlations, or categories in the input data and code for them in the output.
- *Quick learning capability*: When an input pattern is not sufficiently similar to any existing prototype, and a new node is created to represent a new category involving the input pattern as the prototype.
- *Concept of vigilance parameter*: The above-mentioned "sufficiently similar" depends on a vigilance parameter $\rho$, with $0 < \rho < 1$. The similarity condition is easier to meet if $\rho$ is small, leading to coarse categorization. On the other hand, if $\rho$ is 1, numerous finely divided categories are formed. The vigilance parameter value can be adjusted during learning such that increasing it can lead to subdivision of existing categories.

### 2.4.2. ART1 architecture

Fig. 5 illustrates the architecture of the ART1 neural network. Each input vector $X$ has $m$ binary 0 or 1 elements. Let the weights on the bottom-up links, $x_j$ to $y_i$, be denoted by $\bar{w}_{ij}$, and let the weights on the top-down links, $y_i$ to $x_j$, be denoted by $w_{ij}$. Notably, the first subscript of a top-down weight indicates the source node, while the second subscript indicates the destination node. The weight vectors $w_i = (w_{i1}, w_{i2}, \ldots, w_{im})^{\mathrm{T}}$, $i = 1, 2, \ldots, n$, represent stored prototype vectors and thus are also binary 0 or 1 vectors, where $i$ indicates the output nodes or categories, each of which can be enabled or disabled.

### 2.4.3. ART1 algorithm

To clearly describe the operation of the ART1 neural network, the algorithm characterizing ART1 is as follows. Before detailing the algorithm, the variables used in the ART1 neural network are summarized as follows: $m$ is the number of input vector elements, $n$ the number of output nodes, $w_{ij}$ the weights on the top-down links, where $i$ denotes the index of the output node ($i = 1, 2, \ldots, n$) and $j$ represents the index of the input vector elements ($j = 1, 2, \ldots$, m), $\bar{w}_{ij}$ the weights on the bottom-up links, $x$ the input vector, $y_i$ the net value, $r$ the similarity value, and $\rho$ is the vigilance parameter.

The algorithm of the ART1 neural network is displayed as follows:

*Input:* A set of input vector $x$ to be clustered, where $x \in \{0,1\}^m$.

*Output:* A set of weight vectors $w_i = (w_{i1}, w_{i2}, \ldots, w_{im})^{\mathrm{T}}$, $i = 1, 2, \ldots, n$, representing the prototype vectors of the discovered clusters, where $n$ is the number of clusters identified.

**Step 0.** Set $w_{ij}(0)=1$, $\bar{w}_{ij}(0) = 1/(1+m)$, for $0 < \rho \leq 1$.

**Step 1.** Feed a new sample $x$ to the input nodes.

**Step 2.** Enable all the output nodes.

**Step 3.** Use bottom-up processing to obtain a weighted sum

$$y_i = (\bar{w}_i)^{\mathrm{T}} X = \sum_{j=1}^{m} \bar{w}_{ij} x_j \tag{1}$$

where $\bar{w}_{ij}$ is the normalization of $w_{ij}$ given by

$$\bar{w}_{ij} = \frac{w_{ij}}{0.5 + \sum_j w_{ij}}, \quad j = 1, 2, \ldots, m. \tag{2}$$

**Step 4.** Use the **max net** procedure to identify the output node $i$ with the largest $y_i$ value.

**Step 5.** Verify that $x$ belongs to the $i$th cluster by performing top-down processing and forming the

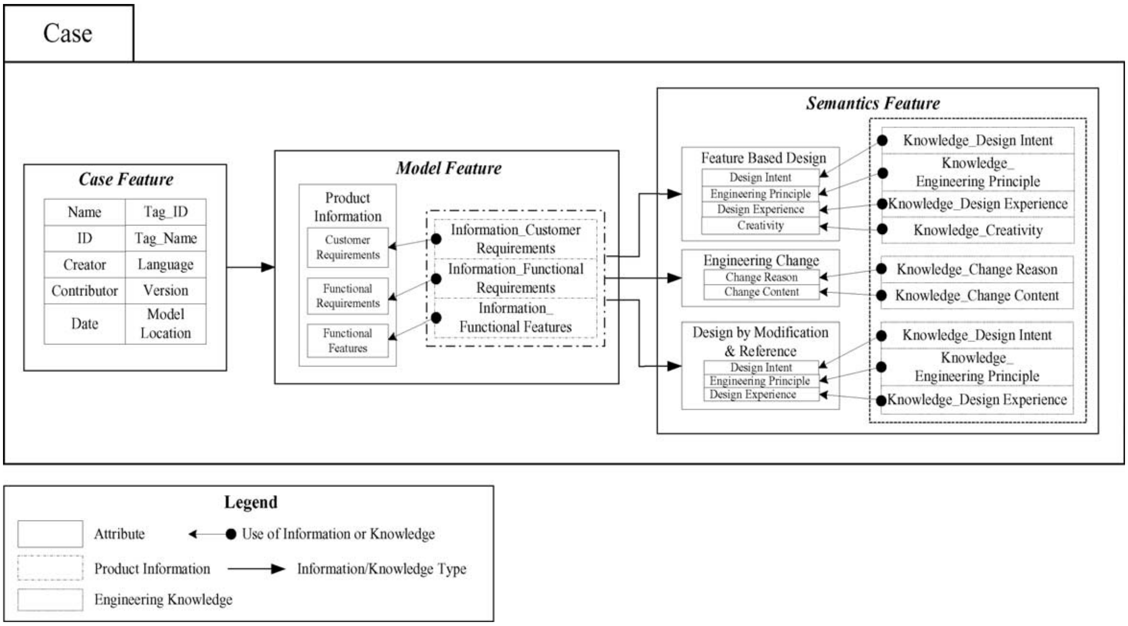


Fig. 6. Case model.

weighted sum $\sum_j w_{ij}x_j$. Then perform the following check:

$$\textbf{If } r = \frac{\sum_{j=1}^{m} w_{ij}x_j}{\|x\|} > \rho, \text{ where } \|x\| = \sum_{j=1}^{m}|x_j|. \qquad (3)$$

**Then** $x$ belongs to the $i$th cluster, proceed to Step 6,

**Otherwise**, **if** the top layer has more than a single enabled node remaining, then go to Step 7,

Alternatively, create a new output node $i$ with its initial weights set as in Step 0 and proceed to Step 6.

**Step 6.** Update the weights as follows:

$$w_{ij}(t+1) = w_{ij}(t)x_j, \quad j = 1, 2, \ldots, m \qquad (4)$$

which updates the weights of the $i$th cluster (either created or existing). Then go to Step 1.

**Step 7.** The output node $i$ is disabled by clamping $y_i$ to 0. This node thus does not participate in the current cluster search. The algorithm returns to Step 3, and will attempt to establish a new cluster different from $i$ for pattern $x$.

Table 1
The binary code for functional features of five cases

| Input samples | Positive feature and positive feature | | | Positive feature and negative feature | | | | Negative feature and negative feature | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Adjacent_to, convex | Intersect | | Add_on, protrusion | Is_in | | | Adjacent_to, convex | Is_in, convex | Intersect | |
| | | Square | Cylinder | | Hole | Groove | Step | | | Square | Cylinder |
| $x_1$ (Case_1) | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| $x_2$ (Case_2) | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| $x_3$ (Case_3) | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $x_4$ (Case_4) | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| $x_5$ (Case_5) | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |

$x_i$ indicates the vector for functional features in Case_$i$.

From the above algorithm, the ART1 algorithm includes both a learning mode and a performance model. For a given input sample $x$, the algorithm can terminate in two ways. First, if a matching prototype vector $w_i$ is found, it is adjusted in Step 6 based on Eq. (4) and the category $i$ is outputted. Second, if the stored categories contain no suitable prototype vector, a new output node $i^*$ is created, which represents a new category with a prototype vector $w_i^*$ that equals input $x$ in Step 6 by Eq. (4), and finally the new category $i^*$ is created.

### 2.5. Functional feature-based cases retrieval by ART: an example

This section uses a hypothetical part to illustrate functional feature-based case retrieval by applying the ART1 neural network. First, a case-based representation of a designed entity is introduced to record related product information and engineering knowledge [4]. Then the functional feature-based case clustering using the ART1 neural network is interpreted using an example. Finally, the vector model is applied to deal with the functional feature-based case ranking.

#### 2.5.1. Case-based representation of a designed entity

From Fig. 6(a) "Case" is viewed as a box that contains related tags and links the product information and engineering knowledge of a design entity (that is, an engineering model). The scheme of a case consists of three features: case feature, model feature, and semantic feature. Case feature defines the contents of case data, such as case name, ID, tag ID, name, model creator, contributor, date, language, version, and location. Meanwhile, model feature indicates the tag for product information that records the detailed information of a design entity, including customer requirements, functional requirements, and functional features. Finally, semantic feature represents the tags for engineering knowledge that also record the design knowledge and experience of engineering designers. These tags for engineering knowledge are classified into three categories: (1) the tag for feature-based design, (2) the tag for engineering change, and (3) the tag for design by modification/reference. Each of these tags points to relevant production information or engineering knowledge.

#### 2.5.2. Functional features-based cases clustering

To validate the aforementioned ART1 technique (as discussed in Sections 2.4.2 and 2.4.3) for functional feature-based case clustering, the functional features of five cases (Case_1, Case_2, Case_3, Case_4 and Case_5) are chosen. Table 1 shows the binary code for these functional features.

From the initialization process of ART1, the initial weights are $w_{ij} = 1$ and $\bar{w}_{ij} = \frac{1}{12}$, $j = 1, 2, \ldots, 11$; $i = 1$–5. Meanwhile, the vigilance parameter $\rho$ is set to 0.5. The input samples are then fed to the ART1 algorithm individually. Additionally, five output nodes are assumed to be available.

- Sample $x_1$ (Case_1): when sample $x_1$ is fed, the one among the five output nodes with the largest output is denoted as number 1. Since $w_{ij} = 1$ for all $i$, $j$ at this time, $r = 1$ in Eq. (3) and the vigilance test is passed unconditionally. Consequently, the first cluster is defined unconditionally. The weights are then changed based on Eqs. (4) and (2):

$$w_{1,3} = w_{1,6} = w_{1,9} = w_{1,10} = w_{1,11} = 1, \quad w_{1,j} = 0, \quad j = 1, 2, 4, 5, 7, 8$$

$$\bar{w}_{1,3} = \bar{w}_{1,6} = \bar{w}_{1,9} = \overline{w}_{1,10} = \overline{w}_{1,11} = \frac{2}{11} \quad \bar{w}_{1,j} = 0, \quad j = 1, 2, 4, 5, 7, 8$$

- Sample $x_2$ (Case_2): when sample $x_2$ is fed, no top-layer node is competing for clustering since only one active node exists; that is, node 1 is the unconditional winner. The vigilance test indicates that

$$r = \frac{\sum_{j=1}^{11} w_1 x_j}{\|x\|} = \frac{1}{3} = 0.33 < \rho = 0.5$$

Hence, it fails the test. Since output node $i$ now is the single enabled node, further searching is unnecessary, and sample $x_2$ is considered a new cluster represented by another output node, number 2. The corresponding weights $w_2$ and $\bar{w}_2$ then are computed as:

$$w_{2,3} = w_{2,4} = w_{2,8} = 1, \quad w_{2,j} = 0, \quad j = 1, 2, 5, 6, 7, 9, 10, 11$$

$$\overline{w}_{2,3} = \overline{w}_{2,4} = \overline{w}_{2,8} = \frac{2}{7}, \quad \overline{w}_{2,j} = 0, \quad j = 1, 2, 5, 6, 7, 9, 10, 11$$

- Sample $x_3$ (Case_3): when sample $x_3$ is fed, the following output values are computed based on Eq. (1):

$$y_1 = \frac{2}{11} = 0.18, \quad y_2 = \frac{4}{7} = 0.57$$

Since $y_1 < y_2$, therefore, output node 2 is a winner. Moreover, the vigilance test succeeds since

$$r = \frac{\sum_{j=1}^{11} w_{2j} x_j}{\|x\|} = \frac{2}{3} = 0.67 > \rho = 0.5$$

Accordingly, weights $w_2$ and $\bar{w}_2$ must be changed based on Eqs. (4) and (2), as follows:

$$w_{2,3} = w_{2,4} = 1, \quad w_{2,j} = 0, \quad j = 1, 2, 5, 6, 7, 8, 9, 10, 11$$

$$\overline{w}_{2,3} = \overline{w}_{2,4} = \frac{2}{5}, \quad \overline{w}_{2,j} = 0, \quad j = 1, 2, 5, 6, 7, 8, 9, 10, 11$$

- Sample $x_4$ (Case_4): when sample $x_4$ is fed, the following output values are computed based on Eq. (1):

$$y_1 = \frac{8}{11} = 0.73, \quad y_2 = 0$$

Since $y_1 > y_2$, therefore, output node 1 is a winner. Moreover, the vigilance test succeeds since

$$r = \frac{\sum_{j=1}^{11} w_{1j} x_j}{\|x\|} = \frac{4}{6} = 0.67 > \rho = 0.5$$

Therefore, weights $w_1$ and $\bar{w}_1$ must be changed based on Eqs. (2) and (4), as follows:

$$w_{1,6} = w_{1,9} = w_{1,10} = w_{1,11} = 1, \quad w_{1,j} = 0, \quad j = 1, 2, 3, 4, 5, 7, 8$$

$$\overline{w}_{1,6} = \overline{w}_{1,9} = \overline{w}_{1,10} = \overline{w}_{1,11} = \frac{2}{9}, \quad \overline{w}_{1,j} = 0, \quad j = 1, 2, 3, 4, 5, 7, 8$$

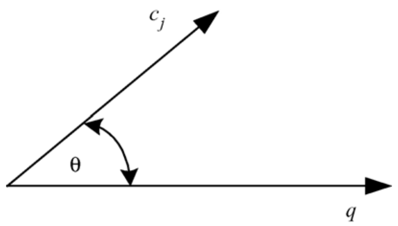


Fig. 7. Cosine of $\theta$ is adopted as sim($c_j$,$q$).

- Sample $x_5$ (Case_5): when sample $x_5$ is fed, the following output values are calculated based on Eq. (1):

$$y_1 = \frac{2}{3} = 0.67, \quad y_2 = \frac{2}{5} = 0.4$$

Since $y_1 > y_2$, therefore, output node 1 is a winner. Moreover, the vigilance test succeeds since

$$r = \frac{\sum_{j=1}^{11} w_1 x_j}{\|x\|} = \frac{3}{5} = 0.6 > \rho = 0.5$$

Therefore, weights $w_1$ and $\bar{w}_1$ must be altered based on Eq. (4) and (2), as follows:

$$w_{1,6} = w_{1,10} = w_{1,11} = 1, \quad w_{1,j} = 0, \quad j = 1, 2, 3, 4, 5, 7, 8, 9$$

$$\overline{w}_{1,6} = \overline{w}_{1,10} = \overline{w}_{1,11} = \frac{2}{7}, \quad \overline{w}_{1,j} = 0, \quad j = 1, 2, 3, 4, 5, 7, 8, 9$$

Proceeding the above learning simulation of the ART1 neural network identifies two categories: one contains samples $x_1$ (Case_1), $x_4$ (Case_4), and $x_5$ (Case_5), and other contains samples $x_2$ (Case_2) and $x_3$ (Case_3). In this example, if $x_5$ (Case_5) is a query pattern, then $x_1$ (Case_1) and $x_4$ (Case_4) are similar cases to the $x_5$ (Case_5) in functional features.

### 2.5.3. *Functional feature-based cases ranking*

The vector model defines the similarity between two terms by the cosine of the angle between their two vectors. Therefore, this vector model is adopted and slightly modified to calculate the degree of similarity between similar cases acquired through functional feature-based case clustering based on the query of functional features. Meanwhile, the query vector $\vec{q}$ can be defined as $\vec{q} = (x_{1,q}, x_{2,q}, \ldots, x_{11,q})$, while the vector for similar cases $\vec{c}_j$ is represented by $\vec{c}_j = (x_{1,j}, x_{2,j}, \ldots, x_{11,j})$. Therefore, a similar case $c_j$ and user query $q$ are represented as 11-dimensional vectors, as shown in Fig. 7. The correlation between vectors $\vec{c}_j$ and $\vec{q}$ is quantified as follows:

$$\mathrm{sim}(c_j, q) = \frac{\vec{c}_j \bullet \vec{q}}{|\vec{c}_j| \times |\vec{q}|} = \frac{\sum_{i=1}^{11} x_{i,j} x_{i,q}}{\sqrt{\sum_{i=1}^{11} x_{i,j}^2}\sqrt{\sum_{j=1}^{11} x_{i,q}^2}}$$

Using the example discussed at the end of the previous subsection, the correlation coefficient among

$x_1$ (Case_1) and $x_5$ (Case_5), $x_4$ (Case_4) and $x_5$ (Case_5) is calculated as:

$$\mathrm{sim}(c_1, c_5) = \frac{\sum_{i=1}^{11} x_{i,1}x_{i,5}}{\sqrt{\sum_{i=1}^{11} x_{i,1}^2}\sqrt{\sum_{j=1}^{11} x_{i,5}^2}} = \frac{4}{\sqrt{5}\sqrt{5}} = 0.8$$

$$\mathrm{sim}(c_4, c_5) = \frac{\sum_{i=1}^{11} x_{i,4}x_{i,5}}{\sqrt{\sum_{i=1}^{11} x_{i,4}^2}\sqrt{\sum_{j=1}^{11} x_{i,5}^2}} = \frac{4}{\sqrt{6}\sqrt{5}} = 0.73$$

According to the above calculation results, $x_1$ (Case_1) resembles $x_5$ (Case_5) than $x_4$ (Case_4). Therefore, the degree of similarity to query pattern $x_5$ (Case_5) follows the order $x_1$ (Case_1) and $x_4$ (Case_4).

## 3. Mechanism implementation and experiment

Based on the proposed techniques for functional feature-based reference design retrieval, this study implemented a prototype functional feature-based reference design retrieval mechanism at the Enter-

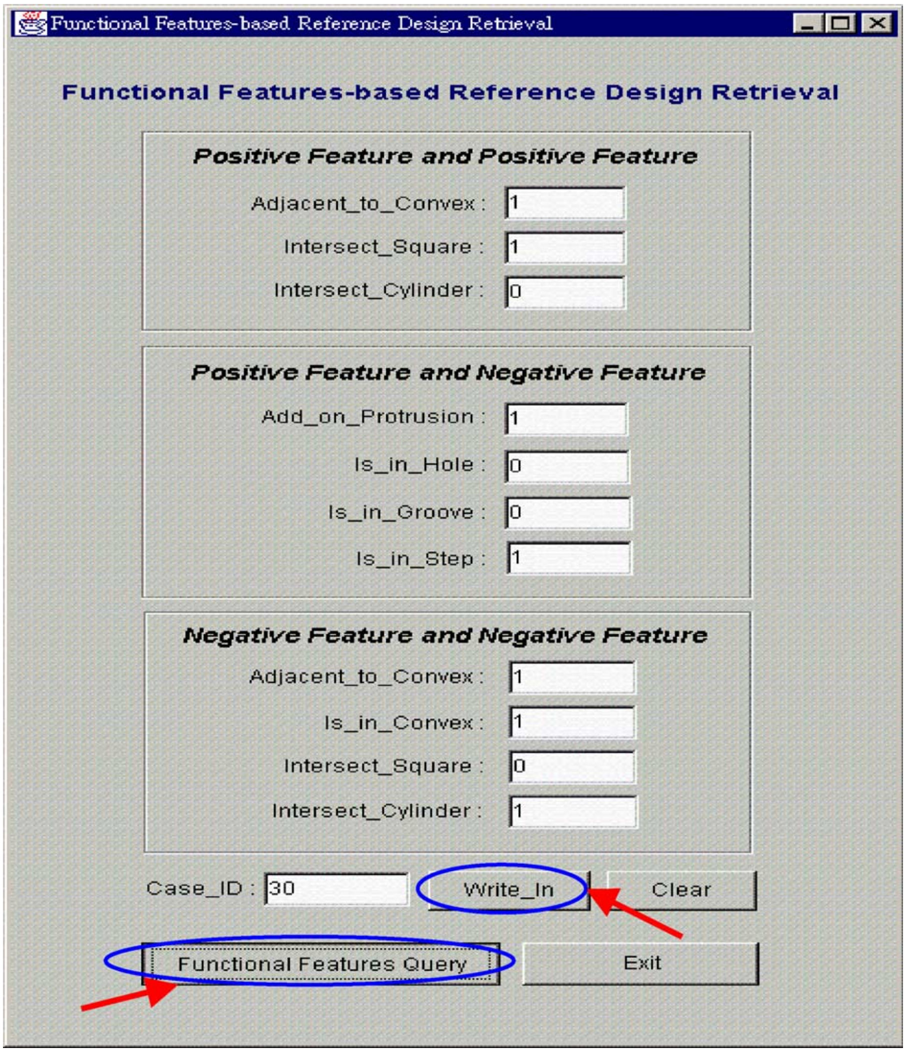


Fig. 8. User interface—functional features query.

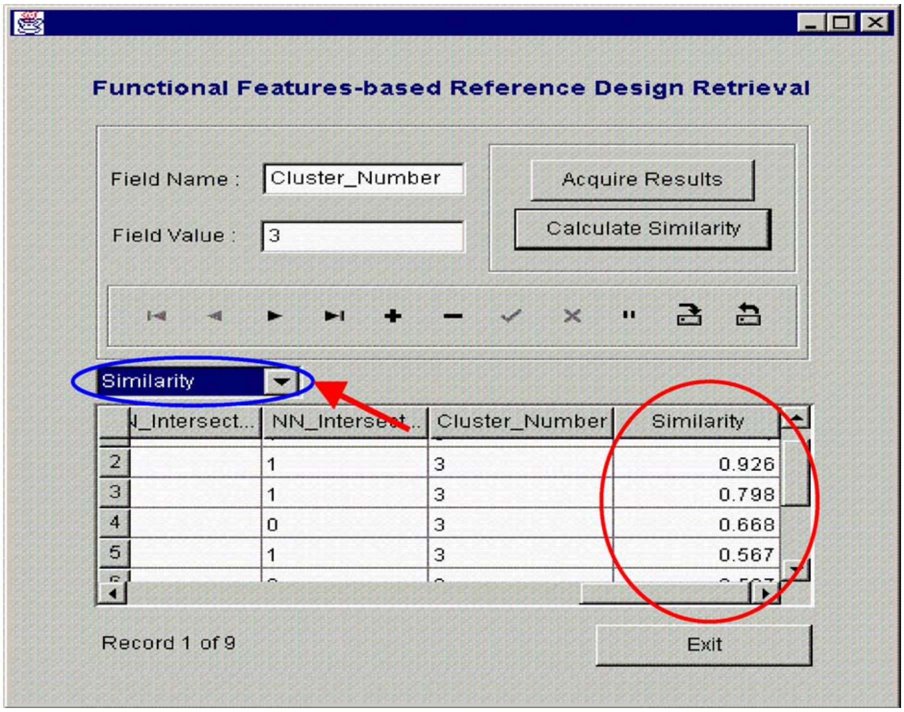


Fig. 9. User interface—similarity ranking for acquired results.

prise Engineering and Integration Research Lab (EE&IRL) of National Cheng Kung University, Taiwan, ROC. This section presents the implementation environment and the results of a functional feature-based reference design retrieval mechanism, as well as the experiment for functional feature-based case clustering. Furthermore, the experiment involves portions of the experimental results and error measurements.

### 3.1. Functional feature-based reference design retrieval mechanism implementation environment and results

Based on the proposed techniques on functional feature-based reference design retrieval, a prototype functional feature-based reference design retrieval mechanism is developed using Java in an environment equipped with the following computer hardware: Acer Veriton 7100 PC and software—Windows 2000 Server, Borland JBuilder 4.0, and Microsoft SQL Server 2000.

Figs. 9 and 10 show two of the user interfaces of a functional feature-based reference design retrieval mechanism. Meanwhile, Fig. 8 shows the screen of functional feature query for the users, while Fig. 9 shows the screen of similarity calculation and ranking for the acquired results.

### 3.2. Experiment for functional feature-based cases clustering

Based on the implemented mechanism, an experiment is performed involving functional feature-based case clustering. This subsection first presents the experimental results on functional feature-based case clustering. Subsequently, error measurements for functional feature-based case clustering are analyzed.

#### 3.2.1. Experimental results

Fig. 10 shows an illustrative example. These experimental results are obtained for 30 input samples using the ART1 neural network. Meanwhile, Fig. 10(a) displays that the 30 samples are classified into six different categories with the lower vigilance value ($\rho = 0.2$), while Fig. 10(b) shows that the 30 input samples are organized into 15 categories with

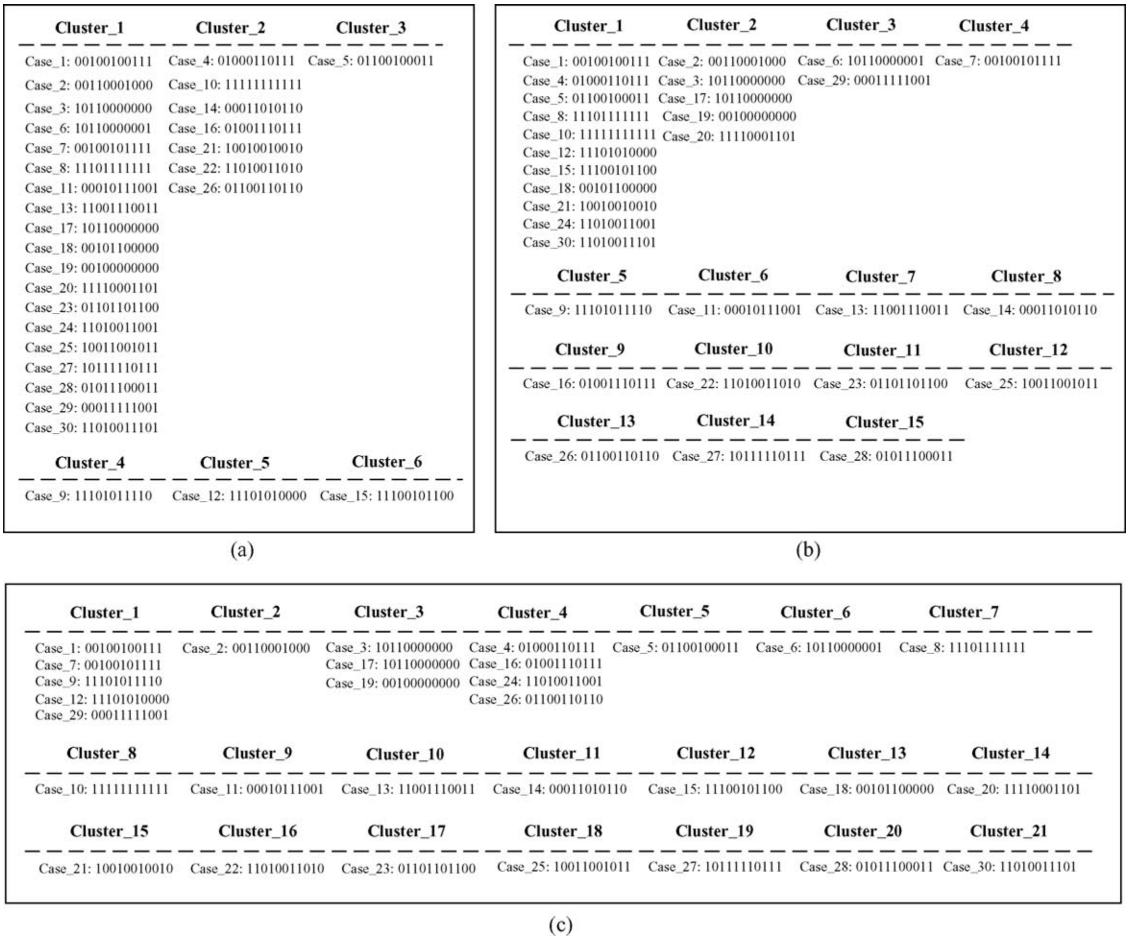


Fig. 10. Category grouping of ART1 neural network with various vigilance parameters (a) $\rho = 0.2$, (b) $\rho = 0.5$ and (c) $\rho = 0.8$.

the middle value of the vigilance parameter ($\rho = 0.5$). With the higher vigilance value ($\rho = 0.8$), the 30 input samples are grouped into 21 recognition categories, as shown in Fig. 10(c).

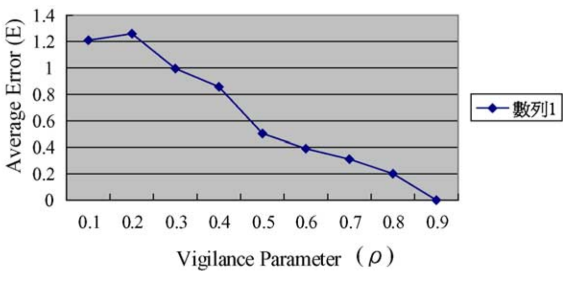


Fig. 11. Average error with variable vigilance parameter.

### 3.2.2. Error measurements

For the error measurements of functional features-based cases clustering, this study applies the concept of "total distance" to measure the error status of functional feature-based case clustering under the conditions of various vigilance parameters (i.e., $\rho = 0.1, 0.2, 0.3, \ldots, 0.9$). The formula for average error is defined as

average error = total distance/cluster number

Furthermore, the total distance is given by $\sum_p(\min_i d_i^p)$,where $d_i^p$ denotes the distance between the $p$th input sample and $i$th output layer.

Fig. 11 illustrates the average error for functional feature-based case clustering with variable vigilance parameters ($\rho$). Clearly, the average error gradually decreases with increasing vigilance parameter of incremental values. Meanwhile, Lippmann [17] indicated the vigilance value ($\rho = 0.9$) is set will be optimum for generating the best result of ART1 clustering.

## 4. Conclusions and discussions

### 4.1. General remarks

This study first presents an engineering knowledge management framework, and then focuses on developing technology for functional feature-based reference design retrieval. The crucial techniques involved in functional feature-based reference design retrieval include: (1) a binary code-based representation for functional features, (2) a case-based representation for organizing information and engineering knowledge of a designed entity, (3) ART1 neural network for functional feature-based case clustering, and (4) similarity calculation for functional feature-based case ranking. The functional feature-based reference design retrieval mechanism is implemented based on the above-mentioned techniques. This mechanism is used to perform an experiment for functional feature-based case clustering and the experimental results are discussed.

The results of this study can facilitate the practice of engineering knowledge sharing for engineering knowledge management in engineering design environments, and subsequently can increase product development capability, reduce development cycle time and cost, and ultimately enhance product marketability.

### 4.2. Future research

Future research could examine the following areas to improve the practice of engineering knowledge sharing in engineering design.

- *Customer need-based reference design retrieval mechanism*: the establishment of customer needs is the first task in the engineering design process. In this task, knowledge workers refer to historical product information and engineering knowledge for performing their work. Therefore, effectively retrieving theses references from the historical knowledge repository based on customer needs represents an important issue in future research.
- *Functional requirement-based reference design retrieval mechanism*: for functional requirement establishment, the functional requirements are obtained by analyzing customer needs. Similarly, historical product information and engineering knowledge are retrieved as references based on the perspective of functional requirements. Therefore, a reference design retrieval should be developed based on the description of functional requirements.
- *Engineering specification-based reference design retrieval mechanism*: this issue represents an extension of this study. Based on the search results of the functional feature-based reference design retrieval mechanism developed in this study, the engineering specification-based reference design retrieval mechanism can filter more precise results by using engineering specifications.

## Acknowledgement

The authors would like to thank the National Science Council of the Republic of China for financially supporting this research under Contract No. NSC 92-2212-E-006-065.

## References

[1] R.B. Allen, P. Obry, M. Littman, in: Proceedings of the ACM COCS, Conference on Organizational Computing Systems, An interface for navigating clustered document sets returned by queries (1993) 66–171.
[2] N. Belkin, B. Croft, Information filtering and information retrieval, Communications of the ACM 35 (12) (1992) 29–38.
[3] D.R. Cutting, J.O. Pedersen, D. Karger, J.W. Tukey, in: Proceedings of the 15th Annual International ACM/SIGIR Conference, Scatter/gather: a cluster-based approach to browsing large document collections (1992) 318–329.
[4] C.B. Wang, Y.J. Chen, Y.M. Chen, Knowledge refinement for engineering knowledge management, Concurrent Engineering: Research and Applications 13 (1) (2005) 43–56.
[5] Y.M. Chen, C.C. Wen, C.T. Ho, Extraction of geometric characteristics for manufacturability assessment, Robotics and Computer-Integrated Manufacturing 19 (4) (2003) 371–385.
[6] S.M. Chen, J.Y. Wang, Document retrieval using knowledge-based fuzzy information retrieval techniques, IEEE Transactions on Systems, Man and Cybernetics 25 (5) (1995) 793–803.
[7] S.M. Chen, Y.J. Horng, C.H. Lee, Document retrieval using fuzzy-valued concept networks, IEEE Transactions on Systems, Man and Cybernetics 31 (1) (2001) 111–118.

[8] W.B. Croft, Knowledge-based and statistical approaches to text retrieval, IEEE Intelligent Systems 8 (2) (1993) 8–12.
[9] C. Colombo, A. Del Bimbo, P. Pala, Semantics in visual information retrieval, Multimedia IEEE 6 (3) (1999) 38–53.
[10] G.A. Carpenter, S. Grossberg, Neural dynamics of category learning and recognition: attention, memory consolidation, and amnesia, in J. Davis, R. Newburgh, E. Wegman (Eds.), Brain Structure, Learning, and Memory, AAAS Symposium Series, 1986.
[11] G.A. Carpenter, S. Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine, Computer Vision, Graphics, and Image Process 37 (1987) 54–115.
[12] A. Ertas, J.C. Jones, The Engineering Design Process, Wiley, NY, 1993.
[13] V.N. Gudivada, V.V. Raghavan, W.I. Grosky, R. Kasanagottu, Information retrieval on the World Wide Web, Internet Computing, IEEE 1 (5) (1997) 58–68.
[14] P.E. Hart, J. Graham, Query-free information retrieval, IEEE Intelligent Systems 12 (5) (1997) 32–37.
[15] D.A. Hull, J.O. Pedersen, H. Schutze, Method combination for document filtering, in: Proceedings of the 19th Annual International ACM SIGIR Conference, 1996, pp. 279–287.
[16] P.R. Keller, M.M. Keller, Visual Cues: Practical Data Visualization, IEEE Computer Society Press, 1993.
[17] R.P. Lappmann, An introduction to computing with neural nets, IEEE ASSP Managize 4 (1987) 4–22.
[18] W.S. Li, J. Shim, K.S. Candan, Y. Hara, WebDB: a web query system and its modeling, language, and implementation, in: Proceedings of Advances in Digital Libraries, Santa Barbara, CA, USA, 1998.
[19] J.D. Mackinlay, R. Rao, S.K. Card, An organic user interface for searching citation links, in: Proceedings of ACM Conference on Human Factors in Computing Systems, vol. 1, 1995, pp. 67–73.
[20] K. Olsen, R. Korfhage, K. Sochats, M. Spring, J. Williams, Visualization of a document collection with implicit and explicit links: the vibe system, Scandinavian Journal of Information Systems (1993).
[21] G. Pahl, W. Beitz, K. Wallace, Engineering Design, Design Council, London, 1984.
[22] L. Rosenfeld, P. Morville, Information architecture for the world wide web, O'Reilly & Associates, 1998.
[23] P. Resnick, H. Varian, Introduction: special issue on collaborative filtering, Communications of the ACM 40 (3) (1997) 56–58.
[24] B. Shneiderman, D. Byrd, W.B. Croft, Sorting out searching: a user interface framework for text searches, Communications of the ACM 41 (4) (1998) 95–98.
[25] S.K.M. Wong, Y.Y. Yao, On modeling information retrieval with probabilistic inference, ACM Transactions on Information Systems 13 (1) (1995) 39–68.
[26] S.K.M. Wong, W. Ziarko, V.V. Raghavan, P.C.N. Wong, On modeling of information retrieval concepts in vector spaces, ACM Transactions on Database Systems 12 (2) (1987) 299–321.

**Dr. Chin-Bin Wang** is currently an associated professor and the Chairman of Electronic Commerce Management Department, Nan Hua University in Taiwan, ROC. He received his PhD degree in computer science from the City University of New York in 1995, and gained his MS degree from University of Southern California and BS degrees from National Tsing Hua University, Taiwan, ROC, in 1985 and 1981, respectively. His research interests include data mining, network management, engineering data and knowledge management and system integration.

**Yuh-Jen Chen** is currently a PhD candidate in the Institute of Manufacturing Engineering, College of Electrical Engineering and Computer Science, National Cheng Kung University, Taiwan, ROC. He received his BS degree from the Department of Mathematics, Chung Yuan Christian University, Taiwan, ROC, in 1999 and MS degree from the Institute of Manufacturing Engineering, National Cheng Kung University, Taiwan, ROC, in 2001. His current research interests include manufacturing information systems, engineering data and knowledge management and supply chain management.

**Dr. Yuh-Min Chen** is currently a professor and the director of Institute of Manufacturing Engineering, College of Electrical Engineering and Computer Science, National Cheng Kung University, Taiwan, ROC. He graduated from the Ohio State University with a PhD degree in industrial and systems engineering in 1991 and received his MS and BS degrees from National Tsing Hua University, Taiwan, ROC, in 1981 and 1983, respectively. Before joining the faculty of Institute of Manufacturing Engineering in 1994, he worked as a research engineer in Structural Dynamics Research Corporation, USA for 3 years. His current research interests include enterprise integration, engineering data and knowledge management, computer-aided concurrent engineering and manufacturing information systems.

**Dr. Hui-Chuan Chu** is an associate professor of Department of Special Education, National University of Tainan in Taiwan, ROC. She received her PhD degree from Columbia University in 1998. Her research interests are knowledge management, teacher knowledge and integration of information technology in teacher education.