San Francisco, California
July 5 - 7, 2006

SEKE

2006

Proceedings of the
Eighteenth International
Conference on
Software Engineering &
Knowledge Engineering
PROCEEDINGS

SEKE 2006

The 18th International Conference on Software Engineering & Knowledge Engineering

Sponsored by
Knowledge Systems Institute Graduate School, USA

Technical Program
July 5 - 7, 2006
Hotel Sofitel, San Francisco Bay, California, USA

Organized by
Knowledge Systems Institute Graduate School, USA
SEKE’2006 Foreword

On behalf of the Program Committee of the Eighteenth International Conference on Software Engineering and Knowledge Engineering (SEKE’2006), we would like to welcome you to San Francisco, USA. The conference aims at bringing together experts in software engineering and knowledge engineering to discuss on relevant results in either software engineering or knowledge engineering or both. Special emphasis is on the transference of methods between both domains.

It is our pleasure to announce that by the deadline of 15 March 2006, the main conference and three workshops received 219 submissions from 29 countries. All the papers have been rigorously reviewed by at least 3 members of the international Program Committee. Based on the review results, 99 papers have been accepted as regular papers, with an acceptance rate of 45%, and 46 accepted as short papers, with an acceptance rate of 21%. We would like to thank all the authors for their contributions.

This year, we have a rich collection of activities in the technical program, including three keynote speeches, one panel, three workshops, and 21 technical sessions. The keynotes, workshops, and technical sessions cover a wide range of topics in software engineering and knowledge engineering, including:

AI for Energy Production, 
Aspect-Oriented Development,
Collaborative Systems,
Component-Based Systems,
Data Mining,
Databases,
Decision Support,
Empirical Software Engineering,
Formal Methods,
Knowledge Acquisition,
Multi-Agent Models,

Ontologies,
Programming Languages,
Rule-based Systems,
Software Architecture,
Software Development,
Software Evolution,
Software Testing,
UML and Modeling,
Web Services, and
Web and Workflow Management.

We are very grateful to the three keynote speakers, Ron Hira, Kuo-Wei Hwang, and Gordon Simpson; and workshop organizers, Christine W. Chan, Jerry Gao, Huaglory Tianfield, Wei-Tek Tsai, Hongji Yang, and Hong Zhu. The members of the Program Committee should be congratulated and specially thanked for their publicity effort and timely reviews of the submitted papers.

Finally, we would like to thank Shi-Kuo Chang for his guidance and leadership throughout organization of this conference. The assistance of the staff at KSI is also greatly appreciated. Special thanks go to Rex Lee, for his effective and efficient assistance in working with the paper submission and review system, which has made the review process smooth and timely.

Kang Zhang, University of Texas at Dallas, USA
George Spanoudakis, City University, UK
Giuseppe Visaggio, University of Bari, Italy
The 18th International Conference on Software Engineering & Knowledge Engineering (SEKE'2006)

July 5-7, 2006
Hotel Sofitel, San Francisco Bay, California, USA

Organizers & Committee

Steering Committee

Vic Basili, University of Maryland, USA
Bruce Buchanan, University of Pittsburgh, USA
Shi-Kuo Chang, University of Pittsburgh, USA
C. V. Ramamoorthy, University of California, Berkeley, USA

Conference General Chair

Shi-Kuo Chang, University of Pittsburgh, USA

Program Chair

Kang Zhang, University of Texas at Dallas, USA

Program Co-Chairs

George Spanoudakis, City University, UK
Giuseppe Visaggio, University of Bari, Italy

Program Committee

Silvia Teresita Acuña, Universidad Autónoma de Madrid, Spain
Anneliese Andrews, Washington State University, USA
# Table of Contents

## Foreword

---

## Conference Organization

---

## Keynote

Outsourcing America  
*Professor Ron Hira*  
---  
1

Information Services in Service Oriented Architecture -- Challenges and Opportunities  
*Dr. Kuo-Wei Hwang*  
---  
2

A Pragmatic Approach to Enterprise Services Orientation  
*Gordon Simpson*  
---  
3

## Aspect-Oriented and Collaborative Systems

Metamodel Access Protocols for Extensible Aspect-Oriented Modeling  
*Naoyasu Uabayashi, Tetsuo Tamai, Shinji Sano, Yusaku Maeno, Satoshi Murakami*  
---  
4

Modeling Complex Software Systems Using an Aspect Extension of Object-Z  
*Huiqun Yu, Dongmei Liu, Zhiqing Shao, Xudong He*  
---  
11

Customizing Aspect-Oriented Variabilities using Generative Techniques  
*Ulirá Kulesza, Carlos Lucena, Paulo Alencar, Alessandro Garcia*  
---  
17

Collaboration Support Model of Software Development Experiment  
*Saeko Matsuura, Hiroki Kurihara*  
---  
23

Enhancing Semantic Interoperability in Collaborative Systems  
*Flavio De Paoli, Marco Loregian*  
---  
29

## Rule-Based Systems

Combining AI Techniques into a Legal Agent-based Intelligent Tutoring System  
*Ig Bittencourt, Marcos Tadeu, Evandro Costa*  
---  
35

Using Conditional Probability to Measure Rule-based Knowledge Similarity  
*Chin-Jung Huang, Min-Yuan Cheng*  
---  
41
Reverse Engineering of Rule-based Systems  
*Abdelhamid Bouchachia, Daniel Wakounig*  

45

A New Method of Value-Adding Treatment Inference for Rule-based Uncertainty Knowledge  
*Chin-Jung Huang, Min-Yuan Cheng*  

51

A Rule-Based Expert System for the Diagnosis of Convergence Problems in Circuit Simulation  
*Christopher W. Lehman, Mary Jane Willshire (S)*  

57

---

**Data Mining**

Using Data Mining Schemes for Improvement on System Performance in Virtual Environments  
*Shao-Shin Hung, Damon Shing-Min Liu*  

61

An architecture based on multi-agent system and data mining for recommending research papers and researchers  
*Sílvio César Cazelia, Luis Otávio Campos Alves*  

67

Salient Phrases-based Clustering and Ranking in Chinese Bulletin Board System  
*Xiao Yuan Wu, Shen Huang, Yong Yu*  

73

GEOARM: an Interoperable Framework to Improve Geographic Data Preprocessing and Spatial Association Rule Mining  
*Vania Bogorny, Paulo Martins Engel, Luis Otavio Alvares*  

79

Classification by Multi-Perspective Representation Method  
*Jia Zeng, Reda Alhajj*  

85

---

**Software Architecture**

An Architecture for Personal Cognitive Assistance  
*David Garlan, Bradley Schmerl*  

91

Updating Styles Challenge Updating Needs within Component-based Software Architectures  
*Mourad Oussalah, Dalila Tamzalit, Olivier Le Goaer, Abdelhak-Djamel Serli (S)*  

98

Verifying a Software Architecture Reconstruction Framework with a Case Study  
*Seonah Lee, Sungwon Kang (S)*  

102

What's in Constructing a Domain Model for Sharing Architectural Knowledge?  
*Rik Farenhorst, Remco C. de Boer, Robert Deckers, Patricia Lago, Hans van Vliet (S)*  

108

A Pattern Taxonomy for Business Process Integration Oriented Application Integration  
*Helge Hofmeister, Guido Wirtz*  

114
Verification & Decision Support

A Model-based Design-for-Verification Approach to Checking for Deadlock in Multi-threaded Applications
Beata Sarna-Starosta, R. E. K. Stirewalt, Laura K. Dillon ........................................ 120

A PVS Approach to Verifying ORA-SS Data Models
Scott Uk-Jin Lee, Gillian Dobbie, Jing Sun, Lindsay Groves ........................................ 126

Decision Support for Resource-centric Software Release Planning
Jim Mc Elroy, Guenther Ruhe ......................................................................................... 132

Managing Uncertainty in Agile Release Planning

A Decision Modelling Approach for Analysing Requirements Configuration Trade-offs in Time-constrained Web Application Development
Sven Ziemer, Pedro R. Falcone Sampaio, Tor Stålhane (S) ............................................... 144

Databases

Multi-model Based Optimization for Stream Query Processing
Ying Liu, Beth Plale .......................................................................................................... 150

Applying MDA to the Conceptual Design of Data Warehouses
Leopoldo Zepeda, Matilde Celma .................................................................................... 156

A Data Warehouse Architecture in Layers for Science and Technology
André Luís Andrade Menolli, Maria Madalena Dias (S) ................................................. 162

Querying Ontology Based Databases - The OntoQL Proposal
Stéphane Jean, Yamine Aït-Ameur, Guy Pierr ................................................................ 166

Towards a Conceptual Framework to Classify Ubiquitous Software Projects
Rodrigo O. Spinola, Jobson L. M. da Silva, Guilherme H. Travassos (S) ...................... 172

Software Development

Open Source Development Process: a Review
Marco Scotto, Alberto Sillitti, Giancarlo Succi .............................................................. 176

Organizational Programming: Hierarchy Software Construction
Zhao Yin, JianMin Wang (S) .............................................................................................. 182
Towards a Methodology for Hybrid Systems Software Development
Isabel María del A’guila, Joaquín Canadas, José-Palma, Samuel Tu’nez

After the Scrum: Twenty Years of Working without Documentation
Sukanya Ratanotayanon, Jigar Kotak, Susan Elliott Sim

Applying Models of Technology Adoption to Software Tools and Methods: An Empirical Study
Scott A. Bailey, Susan Elliott Sim

Empirical SE

Measuring the Usability of Online Stores
Ernest Cachia, Mark Micallef

Key Issues and Metrics for Evaluating Product Line Architectures
Soo Ho Chang, Hyun Jung La, Soo Dong Kim

Multiple Imputation of Software Measurement Data: A Case Study
Taghi M. Khoshgoftaar, Jason Van Hulse

Polishing Noise in Continuous Software Measurement Data
Taghi M. Khoshgoftaar, Christopher Seiffert, Jason Van Hulse (S)

3D Visualization of Class Template Diagrams for Deployed Open Source Applications
Benjamin N. Hoipkemier, Nicholas A. Kraft, Brian A. Malloy (S)

Parallel Monitoring of Design Pattern Contracts
Jason O. Hallstrom, Andrew R. Dalton, Neelam Soundarajan

An Empirical Study of the Maintenance Effort
Liguo Yu, Kai Chen (S)

Experimental Study on the Impact of Team Climate on Software Quality
Silvia T. Acuña, Marta Gómez, Ramón Rico

Web Object Cacheability – How Much Do We Know?
Chi-Hung Chi, Jun-Li Yuan, Lin Liu (S)

Bayesian Estimation of Defects based on Defect Decay Model: BayesED^3M
Syed Waseem Haider, João W. Cangussu (S)

Component

A Component Model to Support Dynamic Unanticipated Software Evolution
Hyggo Almeida, Angelo Perkusich, Glauber Ferreira, Emerson Loureiro, Evandro Costa

xiii
AOSDM Workshop

AOSDM-I: Multi-Agent Models

The Dynamic CasteShip Mechanism for Modeling and Designing Adaptive Agents
Xinjun Mao, Zhiming Chang, Lijun Shang, Hong Zhu, Ji Wang .......................... 639

An Ontology Based Multi-Agent System Conceptual Model
Walid Chainbi ................................................................. 645

A Hierarchical Agent-oriented Knowledge Model for Multi-Agent Systems
Liang Xiao, Des Greer ......................................................... 651

A Comparative Analysis of l*Agent-Oriented Modelling Techniques
Gemma Grau, Carlos Cares, Xavier Franch, Fredy J. Navarrete ......................... 657

A Negotiation Model for the Process Agents in an Agent-Based Process-Centered Software Engineering Environment
Neo Li, Mingshu Li, Qing Wang, Shuanzhu Du ........................................ 664

AOSDM-II: Agent-Oriented Development

A Formal Architectural Model For Mobile Service Systems
Zuohua Ding ................................................................. 670

An Environment of Knowledge Discovery in Database
Maria Madalena Dias, Roberto Carlos dos Santos Pacheco, Lúcio Gerônimo Valentin 676

Genre-based approach to Requirements Elicitation
Aneesh Krishna, Rodney J. Clarke, Aditya K. Ghose .................................. 682

Mobility-based Runtime Load Balancing in Multi-Agent Systems
Jan Stender, Silvan Kaiser, Sahin Albayrak ............................................. 688
EECC Workshop

EECC-I: Web Service Composition

Elevating Interaction Requirements for Web Service Composition
M. Hepner, M.T. Gamble, R. Gamble (S) .................................................. 697

Unanticipated Connection of Components Based on Their State Changes Notifications
Luc Fabresse, Christophe Dony, Marianne Huchard ........................................ 702

Service Design with the ServiceBlueprint
Jochen Meis, Lothar Schöpe ................................................................. 708

Towards Context-based Mediation for Semantic Web Services Composition
Michael Mrissa, Chirine Ghedira, Djamal Benslimane, Zakaria Maamar .................. 714

EECC-II: Component-Based Systems

The Research and Design of Layered-metadata used for Component-based Software Testing
Liangli Ma, Yansheng Lu, Mengren Liu (S) .................................................. 720

QoSPL: A QoS-Driven Software Product Line Engineering Framework for Distributed Real-time
and Embedded Systems
Shih-Hsi Liu, Barrett R. Bryant, Jeff Gray, Rajeev Raje, Mihran Tuceryan, Andrew Olson,
Mikhail Auguston .................................................................................. 724

Performance Evaluation of Component System based on Container style Middleware
Yong Zhang, Ningjiang Chen, Jun Wei, Tao Huang ........................................ 730

Two Perspectives on Open-Source Software Evolution: Maintenance and Reuse
Liguo Yu, Kai Chen .............................................................................. 737

Reviewers' Index .................................................................................. 743

Authors' Index .................................................................................... 746

Note: (S) means short paper.
Using Conditional Probability to Measure Rule-based Knowledge Similarity

Chin-Jung Huang, Min-Yuan Cheng

1 Department of Mechanical and Computer-Aided Engineering, St. John’s University, Taiwan
2 Department of Construction Engineering, NTUST, Taiwan
jimrong@mail.sju.edu.tw

Abstract

In the process of rule-based knowledge accumulation, due to various knowledge sources and various expert comments in the knowledge base, specific knowledge elements in the knowledge base may be in duplicate, in conflict, or inconsistent. The application of wrong information may even lead to wrong decisions. This study proposes the O-A-RV structure for rule-based knowledge and integrates conditional probability, vector matrices, and artificial intelligence to establish the conditional probability knowledge similarity algorithm and develop the knowledge similarity calculation system, which together can quickly and accurately calculate knowledge similarity matrices and determine the relationship among knowledge items. Moreover, according to knowledge relationships, through the inference of value-added treatment such as merging, integration, deletion, innovation and additions, the accuracy of the knowledge itself can be securely ensured and wrong decisions be avoided.

Keywords: Rule-based Knowledge, Conditional Probability, Vector Matrices, Artificial Intelligence, Similarity

1. Introduction

In establishing a rule-based knowledge base, experts are always alert to whether there may exist in the base any logic or structure errors. That is, they insist on the verification of redundancy rules, conflict rules, circularity rules and incompleteness rules. However, in the process of knowledge accumulation, due to different knowledge sources in the knowledge base and different opinions that may be held by different experts, specific information in the knowledge base may be in duplicate, in conflict or inconsistent. This may cause problems such as information unsuitable for use. More importantly, the application of wrong knowledge may lead to wrong decisions.

Knowledge value-added treatment should be done according to relationships among knowledge items, which are determined on the basis of knowledge similarities. So, how to accurately calculate similarity becomes the most basic and necessary task.

2. Literature Review

According to statistics done by McGill, Koll and Noreault in 1979, current methods for measuring similarity were continually growing and already numbered more than 60 types including inner product, Dice coefficient, cosine coefficient, Jaccard coefficient, overlap coefficient, etc. [1]. However, the most popular method today remains one based on the distance between the two end-points of two vectors and the angle between the two vectors. Distance in a geometric distance model is usually represented by Euclidean distance; and the angle by dot product [2].

In 1999 and 2001, Zhang put forward the similarity measure method integrating distance and angle. In this method, distance similarity uses an exponential function, with the bottom between 0.7-0.97, and angle similarity a cosine function [4][5]. In 1997, Frank proposed that the retrieval of different information may require different known similarity measuring methods; for example, after retrieval, the file similarity measure is transformed to the vector of numeric value and then the similarity between two vectors is calculated [3].

3. Measuring Rule-based Knowledge Similarity

3.1 Knowledge Representation in the O-A-RV Format

The syntax of rule-based knowledge representation is IF <antecedent> THEN <consequent>. The antecedent and the consequent, whatever they are, can be represented as a sentence. This paper proposes an improved O-A-RV structure comprising the following four components: Object (O), Attribute (A), Relationship Operator (R) and Linguistic Value (V), as shown in Fig. 1. Some examples of sentences represented by the O-A-RV structure are shown in Table 1.

The attributes of each object in the antecedent and consequent in knowledge, after proper transformation and mapping into numerical types, can be described by the three components in the O-A-RV structure. The RV component of a null component will be Null. The attribute representation of antecedent objects forms an antecedent
vector, while that of consequent objects forms a consequent vector. If the antecedent has \( n \) object attributes, it requires \( 3^n \) components for representation as described in Eq. (1). The consequent vector is expressed in the way of the antecedent vector. The antecedent and consequent vectors are then combined to form the knowledge vector.

3.2 Transform Mapping of O-A-RV

The possible data types of O-A-V components in a knowledge sentence are nominal, ordinal, or interval and ratio, as shown in Table 2.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Operation Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal</td>
<td>Unable to compare its magnitude. Unable to do arithmetic calculations.</td>
</tr>
<tr>
<td>Ordinal</td>
<td>Finite, with order relationships, able to compare its magnitude, unable to do arithmetic calculations.</td>
</tr>
<tr>
<td>Interval ratio</td>
<td>Numerical, able to compare its magnitude and do arithmetic calculations</td>
</tr>
</tbody>
</table>

(1) Transformation of the O-A component

When the data type of the O-A component in a knowledge sentence is nominal, its transformation is 0 or 1 respectively, determined by whether there is an identical character or not, as shown in Table 3.

<table>
<thead>
<tr>
<th>Object (O)</th>
<th>Attribute (A)</th>
<th>Mapping Value (O)</th>
<th>Mapping Value (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing identical character</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>No identical character</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

(2) Transformation of RV-components

(a) When the data type of V is nominal, its transformation is 0 or 1 according to the nature of the O-A part, determined by whether or not there is an identical character.

(b) When the data type of V is ordinal, it is directly assigned to transform a specific value between 0 and 1 according to Table 4.

<table>
<thead>
<tr>
<th>Description of V</th>
<th>Transformation of V</th>
</tr>
</thead>
<tbody>
<tr>
<td>small, very small, very slow</td>
<td>VS 0</td>
</tr>
<tr>
<td>Slow, slow</td>
<td>S 0.25</td>
</tr>
<tr>
<td>Medium, common speed</td>
<td>M 0.5</td>
</tr>
<tr>
<td>Large, fast</td>
<td>L 0.75</td>
</tr>
<tr>
<td>Large, very large, very fast</td>
<td>VL 1</td>
</tr>
</tbody>
</table>

(3) If the data type of V is interval and Ratio, it is to keep its original numerical value. Furthermore, if all the Relationship operators (R) of the component are the equal sign, it is to normalize Eq. (2), so as to map the value of V into the range from 0 to 1.

\[
V_{\text{norm}} = \frac{V - V_{\text{min}}}{V_{\text{max}} - V_{\text{min}}} (2)
\]

\( V_{\text{norm}} \): the normalized value of V, ranging between 0 and 1
\( V \): the value of V before normalization
\( V_{\text{max}} \): the maximal element in component V
\( V_{\text{min}} \): the minimal element in component V

(4) If the data type of V is interval and ratio, but not all the relationship operators (R) are the equal sign, the transform mapping of RV component is determined by the conditional probability theory.

Conditional probability, \( P(B \mid A) = \frac{P(B \cap A)}{P(A)} \) is the probability that the event B occurs, under the condition that the event A occurs too. \( P(A) \) is the probability of the event A, \( P(A \cap B) \) is the probability that both the event A and event B occur.

Let \( x \) be the value of testing cases (T), \( y \) the value of knowledge cases (K), then the calculations of the transform mapping values of RV component, under the different situations of \( T>x \) or \( K>y \), are shown in Table 5.

<table>
<thead>
<tr>
<th>V1</th>
<th>V2</th>
<th>Value</th>
<th>K&gt;y</th>
<th>K&lt;y</th>
</tr>
</thead>
<tbody>
<tr>
<td>T&gt;x</td>
<td>max−x</td>
<td>max−y</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T=x</td>
<td>1</td>
<td>max−y</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T&lt;x</td>
<td>x−y</td>
<td>x−min</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Definition \( R = V_{\text{max}} - V_{\text{min}} \)

\( R \): the distributed range of the component V
\( V_{\text{max}} \): the maximal element of the component V
\( V_{\text{min}} \): the minimal element of the component V

Wherein V1 is the relationship between x and y, V2 between K and y. V2 between T and x, max is the maximal element of the component V with a value R further added, min is the minimal element of the component V with a value R further subtracted.

For the testing cases of \( x_1 < T < x_2 \) and \( y_1 < K < y_2 \), which fall within a certain range respectively, wherein V1 is the relationship among \( x_1 \) and \( y_1 \), V2 is the relationship among \( x_2 \) and \( y_2 \). max is the maximal element of the component V with a value R further added, and min is the minimal element of the component V with a value R further subtracted. Due to the pages of paper are limited, so that transformational value of the RV component for other cases can not list in this paper.
3.3 Knowledge Similarity Calculation

After the proper transform mapping and normalization for the knowledge antecedent and consequent, it is able to represent them into the O-A-RV format, and then form the antecedent, consequent and knowledge vectors. If the antecedent and consequent vectors have different dimensions, they are represented into the maximal dimension between them, with those augmented dimensions assigned with zero values. If the antecedent or consequent sentences contain the processing of logic operators like AND or OR, it is described as follows:

IF (a_i AND a_j) THEN c_k, the antecedent vector is formed by six components, [O_k A_k RV_k] [O_k A_k RV_k] IF (a_i OR a_j) THEN c_k, first to divide it into two knowledge operations, IF a_i THEN c_k and IF a_j THEN c_k, with their antecedent vectors represented as [O_k A_k RV_k] [O_k A_k RV_k] respectively, after the complete calculations, then merge them into the original knowledge form IF (a_i OR a_j) THEN c_k.

For m knowledge representations, each with a n-dimensional antecedent, and an l-dimensional consequent, then the antecedent matrix, consequent matrix and knowledge matrix are described by equations from (3):

\[
K = \begin{bmatrix}
  k_{11} & k_{12} & \cdots & k_{1(l \times n)} \\
  k_{21} & k_{22} & \cdots & k_{2(l \times n)} \\
  \vdots & \vdots & \ddots & \vdots \\
  k_{m1} & k_{m2} & \cdots & k_{m(l \times n)}
\end{bmatrix}
\]

(Antecedent Matrix)

\[
= \begin{bmatrix}
  a_{11} & a_{12} & \cdots & a_{1n} & c_{11} & c_{12} & \cdots & c_{1l} \\
  a_{21} & a_{22} & \cdots & a_{2n} & c_{21} & c_{22} & \cdots & c_{2l} \\
  \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
  a_{m1} & a_{m2} & \cdots & a_{mn} & c_{m1} & c_{m2} & \cdots & c_{ml}
\end{bmatrix}
\]

(Consequent Matrix)

When rule-based knowledge is represented into knowledge vectors, such as the two knowledge vectors, \(\vec{k}_i = (k_{i1}, k_{i2}, k_{i3}, \ldots, k_{i(n \times l)})\) and \(\vec{k}_j = (k_{j1}, k_{j2}, k_{j3}, \ldots, k_{j(n \times l)})\), their Euclidean Distance \(D(\vec{k}_i, \vec{k}_j)\), Length \(\|\vec{k}_i\|\), Inner Product \(\vec{k}_i \cdot \vec{k}_j\) are defined in equations from (4) to (6) respectively.

\[
D(\vec{k}_i, \vec{k}_j) = \sqrt{\sum_{n=1}^{nl} (k_{in} - k_{jn})^2}
\]

(4)

\[
\|\vec{k}_i\| = \sqrt{\sum_{n=1}^{nl} k_{in}^2}
\]

(5)

\[
\vec{k}_i \cdot \vec{k}_j = \sum_{n=1}^{nl} k_{in} \cdot k_{jn}
\]

(6)

The present research proposes a knowledge similarity (KKS), which is simpler than the one in [5] and much easier to understand. It is the multiplicity between the Distance Similarity (DS) and Angle Similarity (AS). The KKS between two knowledge representations with m dimensions will take m² times of pairwise calculation, which are described in equations from (7) to (10).

\[
KKS_{ij} = DS(\vec{k}_i, \vec{k}_j) \cdot S(\vec{k}_i, \vec{k}_j)
\]

(7)

\[
DS(\vec{k}_i, \vec{k}_j) = 1 - \frac{D(\vec{k}_i, \vec{k}_j)}{\max_{i,j} D(\vec{k}_i, \vec{k}_j)}
\]

(8)

Constant Coefficient \(C = \frac{1}{1 - DS_{\min}}\)

(9)

\[
S(\vec{k}_i, \vec{k}_j) = \frac{\vec{k}_i \cdot \vec{k}_j}{\|\vec{k}_i\| \|\vec{k}_j\|} = \cos\theta
\]

(10)

The KKS value should be in the range from 0 to 1 since both the Distance Similarity (DS) and Angle Similarity (AS) are ranging from 0 to 1. When the constant coefficient in Eq. (9) is 1, then DSmin is 0, DSmax is the minimal Distance Similarity, which can be set by users.

The larger the KKS of two knowledge vectors, the more similar are the two knowledge vectors. Since KKS is in the range between 0 and 1, when KKS=1, it means the two knowledge vectors are entirely identical; when KKS=0, it means they are totally different. In the same way, the pairwise Antecedent Similarity (AAS), Consequent Similarity (CCS), Antecedent Consequent Similarity (ACS) can be described by equations from (11) to (13).

\[
AAS_{ij} = DS(\vec{a}_i, \vec{a}_j) \cdot S(\vec{a}_i, \vec{a}_j)
\]

(11)

\[
CCS_{ij} = DS(\vec{c}_i, \vec{c}_j) \cdot S(\vec{c}_i, \vec{c}_j)
\]

(12)

\[
ACS_{ij} = DS(\vec{a}_i, \vec{c}_j) \cdot S(\vec{a}_i, \vec{c}_j)
\]

(13)

3.3.1 Knowledge Similarity Matrix

Equations from (14) to (16) define the knowledge similarity matrix, antecedent similarity matrix and consequent similarity matrix. KSM, ASM and CSM might not be symmetric.

(1) KSM (Knowledge Similarity Matrix):

\[
KSM = \{KKS_{ij}\}_{i=1 \times m \times j=1 \times n}, i = 1 \; to \; m, \; j = 1 \; to \; n
\]

(14)

(2) ASM (Antecedent Similarity Matrix):

\[
ASM = \{AAS_{ij}\}_{i=1 \times m \times j=1 \times n}, \; s.t. \; i = 1 \; to \; m, \; j = 1 \; to \; n
\]

(15)

(3) CSM (Consequent Similarity Matrix):

\[
CSM = \{CCS_{ij}\}_{i=1 \times n \times j=1 \times l}, \; s.t. \; i = 1 \; to \; m, \; j = 1 \; to \; l
\]

(16)

3.3.2 Conditional Probability Knowledge Similarity Algorithm (CPKSA)

This study proposes the O-A-RV structure for rule-based knowledge and integrates conditional probability, vector matrices, and artificial intelligence to establish the conditional probability knowledge similarity algorithm (CPKSA), the architecture of which is shown in
Fig. 2, operated in the following steps:
Input: typical rule-based knowledge
Output: knowledge similarity matrix, antecedent similarity matrix, consequent similarity matrix
Step 1: Represent typical rule-based knowledge in the O-A-RV structure
Step 2: Transform and map each O-A-RV component into numerical values; represent them as knowledge matrices.
Step 3: Calculate pairwise similarity
Step 4: Save the matrices of similarity
Step 5: Stop

![Fig. 2. CPKSA Architecture](image)

3.3.3 An Example of Knowledge Similarity Calculation
Five instances of rule-based knowledge are shown in Table 6. If the RV component is not null, the null value of this component is converted to 0. If the RV component is null, the component value that is not null is converted to 1. The knowledge similarity matrices are given in the Table 7.

4. Conclusions and Future Work
To sum up, the present research reaches the following three conclusions:
1. A new knowledge vector representation for rule-based deterministic knowledge can be proposed on the basis of the O-A-RV structure, which comprises four components: object, attribute, relationship operators and linguistic values.
2. A calculation method for knowledge similarity can be proposed by integrating the distance and angle. Also a conditional probability knowledge similarity algorithm (CPKSA) can be proposed.
3. The knowledge case most similar to the testing case can be quickly retrieved from the knowledge base by applying CPKSA, and used for all types of case-based reasoning (CBR) to help decision making and prediction.
Future work is applying the conclusions to real areas of knowledge reasoning, decision assistance and predicting.

5. References

6. Acknowledgements
The authors would like to thank the financial support of the National Science Council of Taiwan government and grant number: NSC-94-2213-E-129-013.

![Table 1. The components of a sentence](image)

<table>
<thead>
<tr>
<th>Sentence</th>
<th>O</th>
<th>A</th>
<th>R</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>The temperature of the engine is more than 100 °C.</td>
<td>Engine</td>
<td>temperature</td>
<td>&gt;</td>
<td>100 °C</td>
</tr>
</tbody>
</table>

Antecedent Vector = \([O_1 A_1 R_1 V_1 O_2 A_2 R_2 V_2 O_3 A_3 R_3 V_3 \ldots O_n A_n R_n V_n] \) (1)

![Table 6. Representation of Knowledge Instances](image)

<table>
<thead>
<tr>
<th>No.</th>
<th>antecedent</th>
<th>consequent</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td>=M &gt;190</td>
<td>=Null</td>
</tr>
<tr>
<td>K2</td>
<td>=M =Null</td>
<td>=45</td>
</tr>
<tr>
<td>K3</td>
<td>=M =Null</td>
<td>&lt;55</td>
</tr>
<tr>
<td>K4</td>
<td>=F &gt;180</td>
<td>=Null</td>
</tr>
<tr>
<td>K5</td>
<td>=F =Null</td>
<td>&gt;70</td>
</tr>
</tbody>
</table>

![Table 7. Knowledge Similarity Matrix (KSM)](image)

<table>
<thead>
<tr>
<th></th>
<th>K1</th>
<th>K2</th>
<th>K3</th>
<th>K4</th>
<th>K5</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td>1.00</td>
<td>0.11</td>
<td>0.11</td>
<td>0.35</td>
<td>0.00</td>
</tr>
<tr>
<td>K2</td>
<td>0.30</td>
<td>1.00</td>
<td>0.84</td>
<td>0.00</td>
<td>0.09</td>
</tr>
<tr>
<td>K3</td>
<td>0.28</td>
<td>0.84</td>
<td>1.00</td>
<td>0.00</td>
<td>0.17</td>
</tr>
<tr>
<td>K4</td>
<td>0.35</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.11</td>
</tr>
<tr>
<td>K5</td>
<td>0.00</td>
<td>0.11</td>
<td>0.19</td>
<td>0.11</td>
<td>1.00</td>
</tr>
</tbody>
</table>

44