퍼지 및 지능시스템학회 논문지 2006, Vol. 16. No. 1, pp. 101-107

Dynamic Knowledge Map and SQL-based Inference Architecture for Medical Diagnostic Systems

Jin Sung Kim

School of Business Administration, Jeonju University
Hyoja-Dong 3-1200, Wansan-Ku, Jeonju, Jeonbuk 560-759, South Korea
Tel: +82-63-220-2932, Fax: +82-63-220-2787, E-mail: kimjs@jj.ac.kr

Abstract

In this research, we propose a hybrid inference architecture for medical diagnosis based on dynamic knowledge map (DKM) and relational database (RDB). Conventional expert systems (ES) and developing tools of ES has some limitations such as, 1) time consumption to extend the knowledge base (KB), 2) difficulty to change the inference path, 3) inflexible use of inference functions and operators. To overcome these limitations, we use DKM in extracting the complex relationships and causal rules from human expert and other knowledge resources. The DKM also can help the knowledge engineers to change the inference path rapidly and easily. Then, RDB and its management systems help us to transform the relationships from diagram to relational table.

Key words : DBMS, Expert systems, Knowledge based systems, Knowledge map (KM), RDB, SQL

1. Introduction

In a specific domain where mathematical models cannot be easily built and the cost of querying a human experts is very high, the contribution ES is highly appreciated. Therefore, during past several decade directors/managers in business organizations seriously discussed about the effectiveness of the ES and applied it to their organizations. To support their hard work, several S/W tools were developed. The majority of S/W tools for building ES or KBS seem to fall into four broad categories [4]:

- *Expert system shell*, which are essentially abstractions over one or more applications programs.
- *High-level programming languages*, which to some extent conceal their implementation details, thereby freeing the programmer from low-level considerations of efficiency in the storage, access and manipulation of data.
- *Multiple-paradigm programming environments*, which provide a set of S/W modules that allow the user to mix a number of different styles of artificial intelligence programming.
- *Additional modules* for performing specific tasks within a problem solving architecture.

However, most of conventional ES and KBS construction mechanisms have several problems [9].

First, traditional KBS were non-applicable because of the conversion form tacit knowledge to explicit documented knowledge was very difficult.

Second, it is often difficult to extend and enhance a KBS with additional expert knowledge once the system is fielded.

Third, within the context of rapidly changing technologies and processes, an existing KBS might no longer seem capable of meeting the increasingly complex knowledge demands in the industry.

To overcome most of the above mentioned pitfalls, we use DKM, relational database (RDB) metaphor and database management systems (DBMS) framework.

First, DKM could help a domain expert who wants to transform his tacit knowledge into explicit knowledge.

Second, DBMS framework could assist a knowledge engineer/manager to link distributed knowledge with causal relationships.

Third, DBMS could help the ES/KBS managers to develop an efficient knowledge inference engine based on OLAP (Online Analytical Processing) concept.

The remains of this paper are organized as follows. The related works are introduced in Section 2. Then, our research methodology is proposed in Section 3. In Section 4, example of DKM construction and inference

접수일자 : 2005년 9월 5일
완료일자 : 2006년 1월 24일

processes are presented. Some of future works and concluding remarks are finally suggested in Section 5.

2. Related works

2.1 KM and DKM

Artificial intelligence (AI), KBS, and ES concept have made an important contribution to our understanding of expert knowledge. Especially, we found that the many researchers in ES field tried to develop a rigorous representation for expert knowledge so that the knowledge could be brought to life in a computer program [7]. There are several accepted methods of knowledge representation that have been devised for AI-type applications. Some of these are also suitable for use and interpretation by humans and can form a bride between human knowledge and machine knowledge.

As one of them, in this study, we use dynamic knowledge map (DKM) originated from KM. KM is the name given [5] to a type of mental diagram by means of which complex ideas can be easily and quickly set out in a logical order. KM typically point to people as well as to documents and databases to enable a person to find an appropriate knowledge source [2]. It can be considered as an effective knowledge representation tools that will encompass, both explicitly and implicitly, the static and dynamic models. The KM ultimately seeks to represent the connections between the properties identified during conceptualization and the process of inferring values on those properties.

Conventional KM locate the holders of knowledge when their expertise is needed rather than spending time with imperfect solutions or searching for explicitly documented knowledge. KM is a graphic representation of the connections made by the brain in the process of understanding facts about something. They are built starting with the attribute that defines the problem to then develop a graphical diagram that sets out on paper the manner in which the mind comes up with ideas in the process of understanding [3].

A general-purpose procedure for constructing a KM during the KBS conceptualization process is as follows:

- Phase 1: Identity the main goal of the system
- Phase 2: Design the goal decision block
- Phase 3: Add the properties for inferring or calculating the goal decision
- Phase 4: Extended the KM
- Phase 5: Repeat phase 4
- Phase 6: Check the knowledge reflected in the KM

The details for each phase are described in below [3].

Phase 1: Identity the main goal of the system

It is essential for drawing up the KM and the main goal should have been decided before the starting of whole KM construction process. The goal property in a medical diagnostics system, for example, would be the disease.

Phase 2: Design the goal decision block

The rectangle includes the whole property/concept and possible values which are used to represent the expert knowledge. It describes the whole concept including where it belongs, using the property/concept from, and the possible values of that property. The possible values used in medical diagnostic systems would be the names of the diseases. Figure 1 shows an example of the KM properties description and its shape.

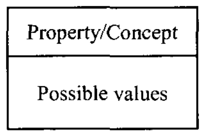

Figure 1. Property representation in medical KM

Phase 3: Add the properties for inferring or calculating the goal decision

Inside the rectangle, place the properties. Then, the relation with the goal property is expressed by means of an arrow that starts from the property used to infer the goal decision. Figure 2 shows the example of added properties for inferring the goal decision in medical diagnostic systems.

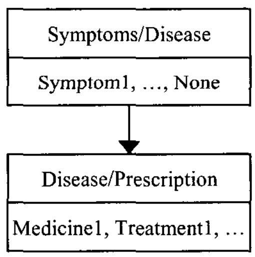

Figure 2. Added properties for inferring

Phase 4: Extended the KM

The properties used to infer the goal decision, have to be added in this phase.

Phase 5: Repeat phase 4

In this phase, the phase-4 should be repeated until none of the peripheral properties are inferred. They may be given from external sources: user input, sensors, files, DB, etc.

Phase 6: Check the knowledge reflected in the KM

First, during the phase 6 we should check related to the validation of the knowledge reflected in the KM with the human expert. The human expert, assisted by the knowledge engineer, will have to check the KM and its information correctly reflects how the human expert reasons to solve the problem in question. Second, we should check related to the verification of the KM against the static and dynamic models. All the attributes and values stored in the Concept/Attribute/Value tables of the static model should be checked. If not, they should be added. Through the process the completeness of KM is validated.

As shown in previous examples, KM has many positive features. Nevertheless, the static nature of most KMs is an obstacle to disseminating tacit knowledge. To overcome these limitations, there is need to find a DKM construction and refinement mechanism. The foundation and basic concept for DKM construction was already proposed by Davenport and Prusak (1998). However, they didn't suggest the details for technical and graphical tools for DKM. In this study, we will combine the Davenport and Prusak's (1998) DKM concept and Gómez et al.'s (2000) KM construction process to extract our own DKM construction mechanism. The details for DKM concept is omitted because the main process of our own DKM construction mechanism is based on Gómez et al.'s (2000) KM construction process.

2.2 SQL-based data classification and knowledge inference

Despite of the large importance of the knowledge management based on DB framework, KBS and ES inference are not widely used in RDB framework. The main reason is that most of conventional ES inference systems were developed based on text or programming language-oriented KB construction and inference framework.

To overcome this limitation, Veryha (2005) proposed SQL-based fuzzy inference system. The main object of that system is fuzzy data classification. He mainly concentrated on comparison of implementation in RDB of conventional data classification and fuzzy data classification using atomic values of attributes. To show the difference between conventional data classification and fuzzy data classification based on SQL and his fuzzy interpreter, he used to the following simple example. Table 1 show the supplier's OFFERs table used in the example.

The OFFERs table contains data about *Suppliers* (Supplier column), *Material Quality* (Quality column), and *Delivery Delay* (Delay column). To query the presented data and produce a list of 'good' suppliers, one can use a typical SQL query as follows:

Table 1 OFFERs table

Supplier	Quality	Delay
BAW	Sufficient	8
G	Average	5
KBA	Sufficient	7
MD	Sufficient	9
MTX	High	2
MAM	Low	7
KBA	Low	9
ZT	Average	2
MTX	High	4
MAM	Average	6
MD	Average	4
DB	High	8
KBA	Sufficient	3
ZT	High	9

```
SELECT Supplier, Quality, Delay FROM Offers
WHERE (Quality = "high" or Quality = "average") and
(Delay<5)
```

The above SQL sentence assumed that 'good' suppliers are those who provide a material of 'high' or 'average' quality with delay of less than five hours. However, if the number of attributes used in SQL query is more than four and ranges of data in columns are significantly larger, SQL queries will become much more complex. Veryha (2005) pointed out the limitation. To overcome this limitation, he proposed fuzzy SQL interpreter as an alternative to shorten the length of SQL queries. Implementation scheme of SQL for fuzzy classified data querying include the following steps [8].

Step-1: Design of DB tables or views

Before the composing of queries, DB administrator or knowledge engineer should have designed DB tables to query them.

Step-2: Design of DB extensions

Extension of DB means that the additional DB table and activities to support the fuzzy classification based on fuzzy interpreter. Additional tables contain linguistic variables, membership values and descriptions of atomic values. This step should be carried out by a human expert in the given application area. In some case, additional programming may be required in this step.

Step-3: Design and implementation of interpreter

In this step, fuzzy interpreter should be designed on the frame of given RDBMS using lexical and syntactical analysis of queries. The interpreter transforms the fuzzy SQL into a native SQL. This step should be carried out by software developer or DB programmer. In Veryha's

(2005) research the interpreter was developed in the form of the stored procedure for the given RDBMS.

Step-4: Generation of DB reports and views

As a result of fuzzy classification by using conventional SQL queries and fuzzy interpreter, fuzzy classified data is presented.

The scheme of the framework implementation in RDB is shown in Figure 3 [8].

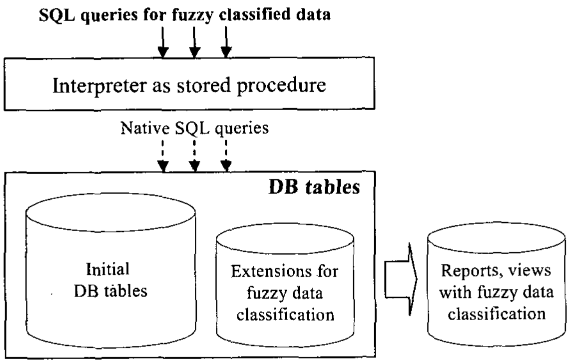

Figure 3. Scheme of fuzzy classification framework implementation in RDB

3. Research Methodology

Considering Gómez et al.'s (2000) KM construction process and Davenport & Prusak's DKM (1998) concept , we propose a technical and graphically manageable DKM construction mechanism. A general-purpose 6-phased DKM construction procedure of totally 9-phased DKM construction procedure is similar with the process proposed by Gómez et al. (2000). However, the detailed activities are some different.

Phase 1: Identify the main goal of the system

In a medical diagnostics system, main goal will be described as a disease suffered by the patient and a prescription presented by a doctor.

Phase 2: Design the goal decision block

Basically, rectangle includes the Property/Concept and its possible values. In our study, we replaced the Property/Concept as Disease/Prescription. Figure 4 shows an example of the properties in KM.

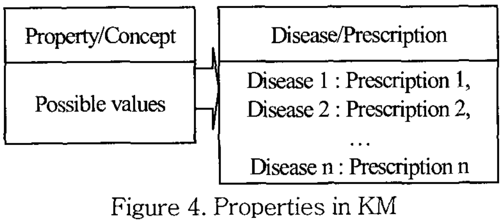

Figure 4. Properties in KM

Phase 3: Add the properties and inference rules for inferring or calculating the goal decision

After the design of goal decision block, to construct the DKM, we added the inference rule into the basic KM block. That is, to infer the properties or find goal decision, we changed the form of block into a form of frame. This phase is a critical difference between our DKM construction mechanism and Gómez et al.'s (2000) KM construction process. Figure 5 shows the example of our proposed DKM frame. The left-hand side means the basic concept of DKM and the right-hand side an example of DKM.

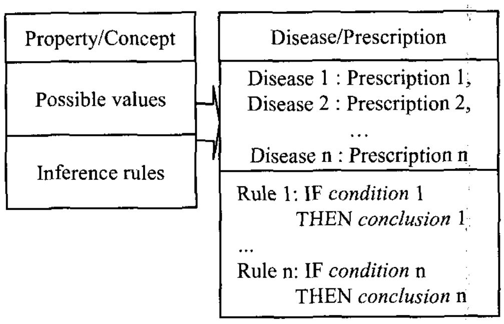

Figure 5. Properties and inference rules in DKM

Phase 4: Extend the DKM

If there were more additional information (property, concept, possible values, inference rules) to infer the decision goal, the information used to infer the values of each property have to be added.

Phase 5: Repeat phase 4

Repeat phase 4 until none of the *peripheral* properties and inference rules are inferred. They will be presented from external sources such as user input, sensors, files, DB, and other external or internal changes.

Phase 6: Check the knowledge reflected in the KM

Knowledge engineer, DB administrator and human expert should validate the completeness of the DKM. Especially the validation of inference rule is the most important activities to improve the performance of diagnostic systems. Then, to combine the DKM with RDB, we added two additional phases to the Gómez et al.'s (2000) KM construction process. Additional phases for DKM are given below.

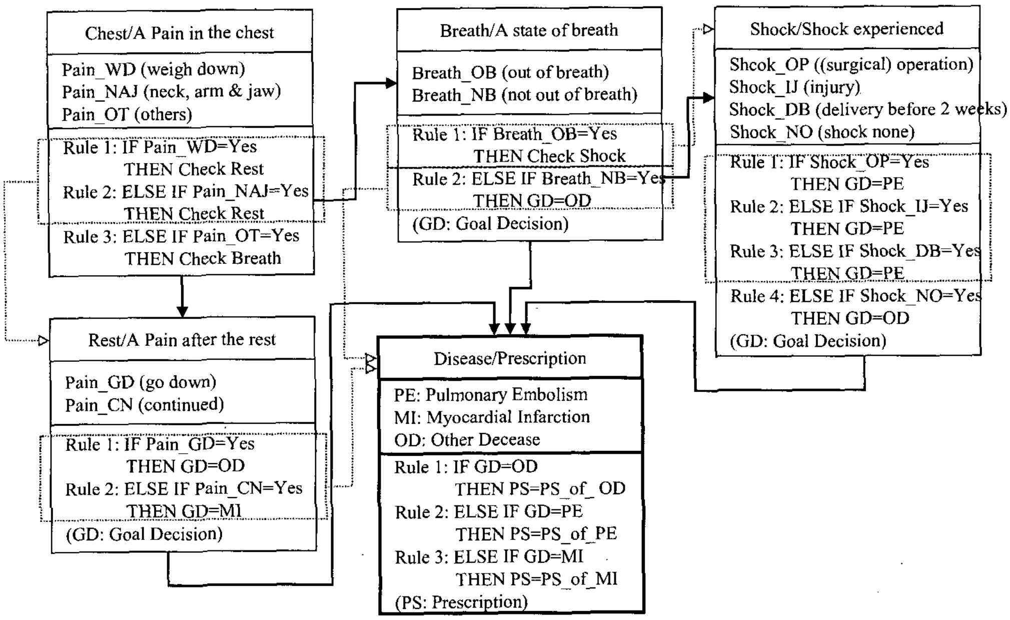

Figure 6. Example of DKM (sub-problems)

Phase 7: Frame-based RDB table construction

After the check of DKM, DB administrator or knowledge engineer should transform the DKM into a form of frame-based RDB table.

Phase 8: Relate the RDB tables

To implement the relationships among each node in DKM, connect the RDB table with other RDB tables by using SQL connection facilities. Figure 6 shows an example of DKM representation.

4. DKM Construction and Inference

To validate the performance of our proposed mechanism, we propose a practical application. The example is a part of a real medical expert's knowledge and illustrates how the DKM is drawn up from the static and dynamic models.

We will omit the process from phase 1 to 6 because of the detailed example was shown in the Section 3. The implementation process of phase 7 and 8 are shown as follows:

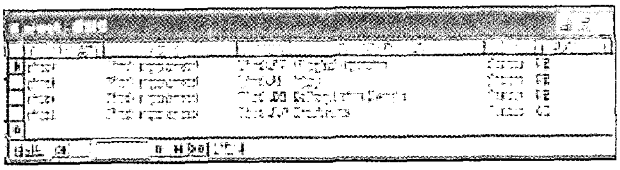
Figure 7. Example of frame-based RDB table

Phase 7: Frame-based RDB table construction

Figure 7 show the example of frame-based RDB table representing one of DKM frames.

Phase 8: Relate the RDB tables

As a result of transformation from DKM into RDB tables, Figure 8 shows the whole RDB tables and their relationships. RDB tables were implemented on MS-Access environment. The relationship between DKM and RDB was shown in Figure 9.

Figure 10 shows the example of inference process and its result. In contrast to Gómez et al.'s (2000) KM construction, our proposed mechanism has several advantages.

First, each RDB table has its inference rules. Therefore, there is no need to construct a huge rule base independently.

Second, it is very easy to revise and extend the KB through the graphical user interface supposed by RDB management systems (RDBMS).

Third, on the basis of our proposed mechanism, ES has no need to have special inference engine. Because of every inference is performed by each knowledge module respectively.

Fourth, there are no conflicts among inference rules. Because of every DKM nodes possess his own inference rules, and its' decision depends on his own properties.

Fifth, DKM node has several inference rules within his block. Contrary to Gómez et al.'s (2000) KM, therefore, our mechanism could handle the multiple choices and inference rules.

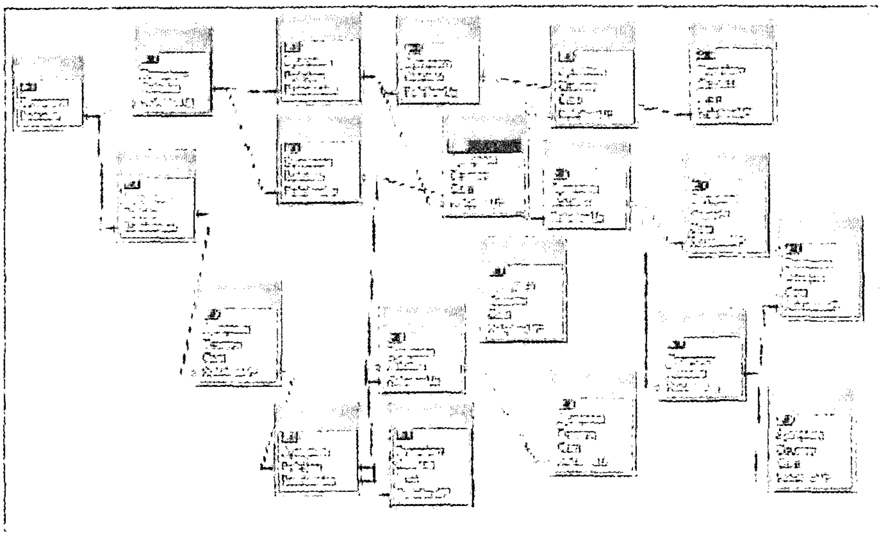

Figure 8. RDB tables and their relationships transformed from DKM

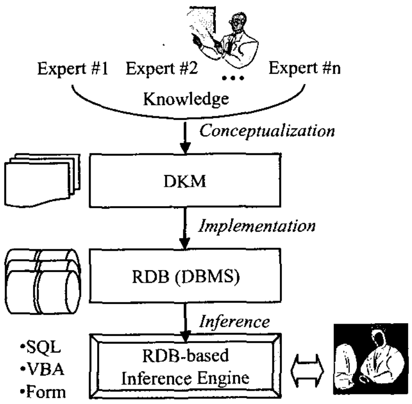

Figure 9. The relationship between DKM & RDB

5. Conclusion

In this study, we extended traditional KM construction process and combined these two different knowledge representation tools as a DKM. The method could support the organizations in several ways:

- It can improve the efficiency of knowledge inference and its application
- It is applicable to real world decision, because of the conversion form tacit knowledge to explicit documented knowledge is very easy.
- It is easy to extend and enhance a knowledge-based system with additional expert knowledge once the system is fielded.

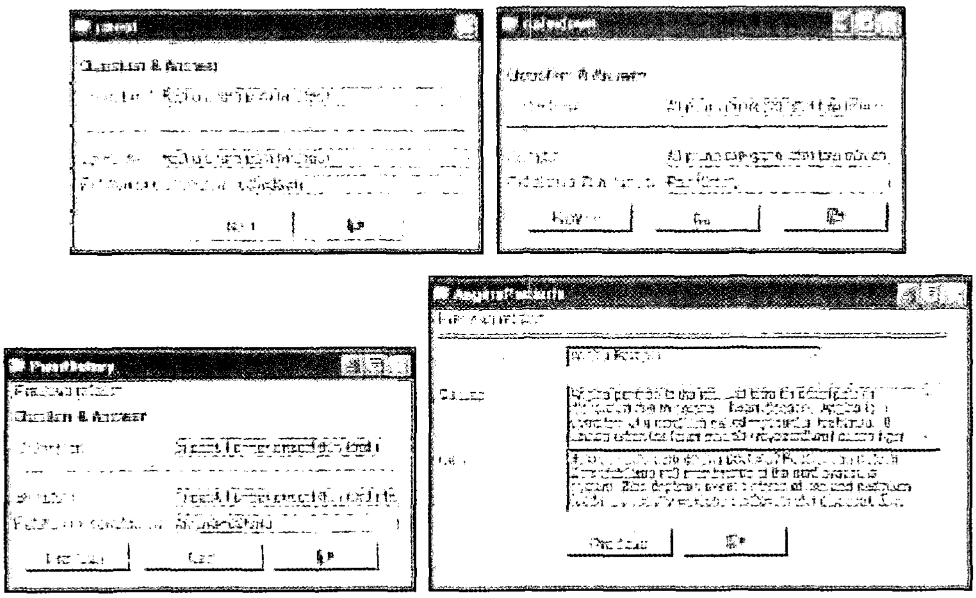

Figure 10. Example of inference

The method also has advantages for the individual and for organizations specializing in education:

- It allows an individual to see and understand a conceptualization process of knowledge and its applications.
- It can be easily applied to the educational field such as ES development or decision support systems (DSS) construction.
- It will identify appropriate directions for the use of knowledge management systems.

Further research should be conducted in order to test the suitability of DKM in real-world application. First, a Web application should be constructed to support the knowledge collection and management on the Web site. Second, a set of real-world experiments will prove the efficiency and robustness of DKM.

References

[1] Beier, J. and Tesche, T. (2001), Navigation and interaction in medical knowledge spaces using topic maps, *International Congress Series*, 1230(0), 384-388.

[2] Davenport, T. and Prusak, L. (1998), *Working knowledge: How organizations manage what they know*, Harvard Business School Press, Boston.

[3] Gómez, A., Moreno, A., Pazos, J., & Sierra-Alonso, A. (2000), Knowledge maps: An essential technique for conceptualization, *Data & Knowledge Engineering*, 33, 169-190.

[4] Jackson, P. (1999), *Introduction to Expert Systems* (3rd eds.), Addison Wesley Longman Limited.

[5] McCagg, E.C. and Dansereau, D.F. (1991), A convergence paradigm for examining knowledge mapping as a learning strategy, *Journal of Educational Research*, 84(6), 317-324.

[6] Polanyi, M. (1966), *The tacit dimension*, Dobuleday, Garden City.

[7] Shortliffe, E.H. (1976), *Computer based medical consultations: MYCIN*, Elsevier, New York.

[8] Veryha, Y. (2005), Implementation of fuzzy classification in relational databases using conventional SQL querying, *Information and Software Technology*, 47(5), 357-364.

[9] Woo, J., Clayton, M.J., Johnson, R.E., Flores, B.E., and Ellis, C. (2004), Dynamic knowledge map: reusing experts' tacit knowledge in the AEC industry, *Automation in Construction*, 13, 203-207.

Appendix: Sample source for inference

```
Private Sub ComboSymptom_AfterUpdate()
 Set rs = Me.Recordset.Clone
 rs.FindFirst "[sID] = " & Str(Nz(Me![ComboSymptom],
    0))
  If Not rs.EOF Then Me.Bookmark = rs.Bookmark
  Me.TextSymptom.ControlSource = "Symptom"
  Me.TextRelation.ControlSource = "Relation"
End Sub

Private Sub CmdNext_Click()
On Error GoTo Err_CmdNext_Click
    Forward = True            '''' Forward Inference
    Me.TextRelation.SetFocus
    If Me.TextRelation.Text = "" Then
       Err.description = "There's no relation"
       GoTo Err_CmdNext_Click
    Else
      stDocName = Me.TextRelation.Text
    End If

    TCell = Me.Name
    TSelection = Me.ComboSymptom.Text
    Forms![Inference]![ListCell].AddItem Item:=TCell
    Forms![Inference]![ListInference].AddItem
     Item:=TSelection
    DoCmd.Close
    DoCmd.OpenForm stDocName, , , stLinkCriteria
End Sub
```

저 자 소 개

Jin Sung Kim 김진성(金珍成)

He has been assistant professor of MIS at the School of Business Administration, Jeonju University, South Korea. His current research interests are in fuzzy logic and AI-based intelligent decision support systems, neural networks, e-business, and Web-based negotiation support systems.

TEL : +82-63-220-2932
FAX : +82-63-220-2787
E-mail: kimjs@jj.ac.kr