

**Ministry of Agriculture & Land Reclamation
Agricultural Research Center
Central Lab for Agricultural Expert Systems**

**The Design of the Interface
Between
KSR & Database**

TR/GLAES/234/2002.4

By

Eng.Mohammed El Helly

April, 2002

1-Introduction.....	3
2-Interface basic elements.....	3
3-KBDB Data Structure	3
3-1 CKBDB Class.....	3
3-2 CKBDBAssociation Class.....	4
4 Visual interface component design.....	4
4-1 Event associated with DB Dialogue.....	5
4-1-1 On Load Dialogue.....	5
4-1-2 On Click OpenDB Button.....	6
4-1-3 On Click Add Association Button.....	6
4-1-4 On Click Del Association Button.....	6
4-1-5 On Change Selection Table List Box.....	7
5 CKBDB Player.....	7

1-Introduction

The purpose of this document is to design the required interface, which support the association between the knowledge base in the KSR and the database schema, which is designed using Microsoft Access 2000.

2-Interface basic elements

The interface design basically consists of two components, the first one is the visual interface component, which is used by the developer to make an association, and the second component is the data structure beyond the visual interface, which keep track the association.

3-KBDB Data Structure

To make an association between the knowledgebase and database the following classes and collection are required:

3-1 CKBDB Class

The responsibility of CKBDB class is to associate specific concept-property with specific table-field if their type matched

```
Class CKBDB{
    Private:
        CString Cpt;
        CString Prop
        CString Table
        CString Field
    Public:
        CKBDB(CString C, CString P, CString T, CString F){
            Cpt = C;
            Prop = P;
            Table = T
            Field = F
        }
        GetCptPropTableField(CString& C, CString& P, CString& T, CString& F){
            C = Cpt ;
            P = Prop;
            T = Table
            F = Field
        }
}
```

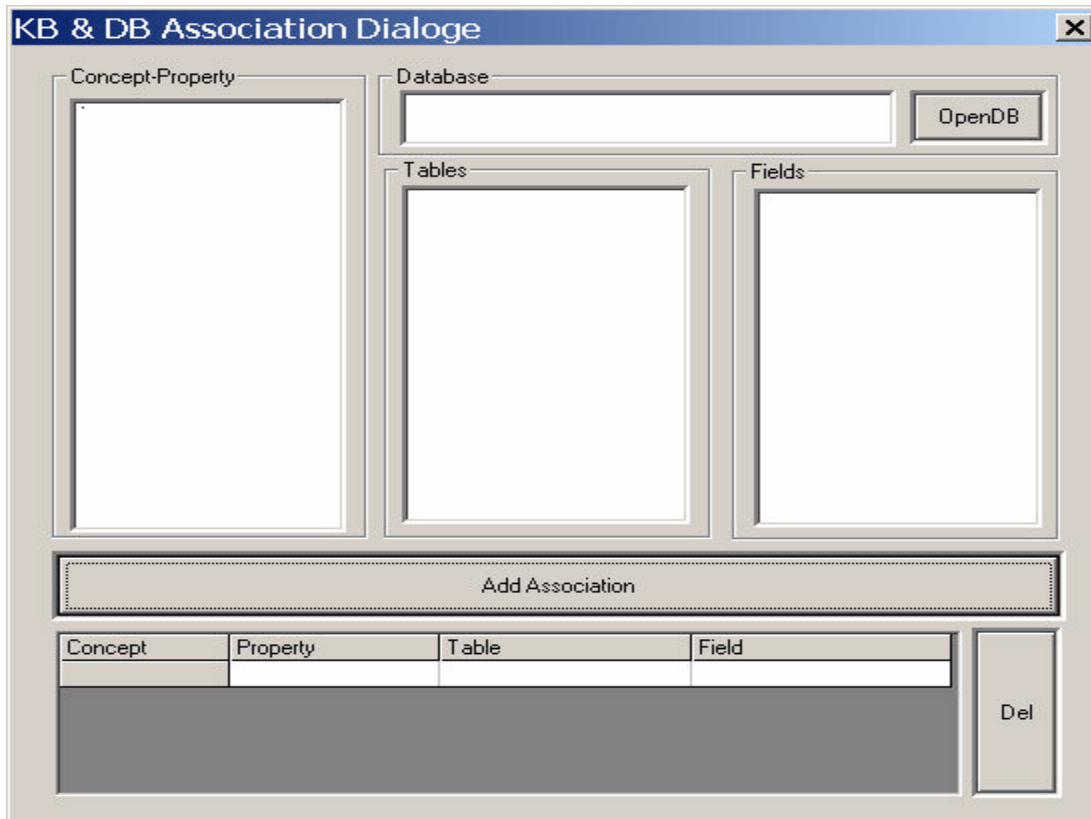
3-2 CKBDBAssociation Class

The responsibility of that class is to keep track the specific database association with an existing knowledge base. It consists of the DBName and the AssociationList the key of that AssociationList is Concept-Property.

```
Class CKBDBAssociation {  
Private:  
    CString DBName;  
    CDBKBList CAssociationList  
}  
typedef CTypedPtrMap<CMapStringToOb,CString, CKBDB *> CDBKBList
```

4 Visual interface component design

The visual interface consists of list boxes, compo boxes, textboxes, buttons, and events. The following Figure Demonstrate the proposed interface design:



4-1 Event associated with DB Dialogue

The above dialogue contains a lot of events; we will describe each of them in the following subsection.

4-1-1 On Load Dialogue

```
Begin
/*Filling Cpt Prop List Box with Concept-Property */
While (NOT End of Concept List){
    Cpt =GetFirstCpt()
    PropList = GetPropList(Cpt)
    While(Not End of PropList){
        Prop = GetFirstProp()
        If (Prop.SourceValue == "Database" )
            AddToListBox(Cpt,Prop,CptPropListBox)
        Prop = GetNextProp()
    }
    Cpt=GetNextCpt()
}
/*Fill DB Association FlexGrid*/
int Row=2;

While(Not End of CAssociationList){
    Col=0;
    Node = GetFirstNode()
    SetFlexGridItem(Row,Col++,Node.Cpt)
    SetFlexGridItem(Row,Col++,Node.Prop)
    SetFlexGridItem(Row,Col++,Node.Table)
    SetFlexGridItem(Row,Col++,Node.Field)
    Row++;
    Node = GetNextNode()
}
DBName = GetDbNameFromCKBDBAssociationClass()
If(Empty(DBName)) return
SetTextBoxValue(DbTextBox, DBName)
OpenDatabase(DBName)
TableName = GetAllTablesInDatabase(DBName)
FillListBox(TableListBox, TableName)
TableFields = GetTableFields(TableName)
FillListBox (FieldsListBox, TableFields)
Disable(OpenDB Button)
End
```

4-1-2 On Click OpenDB Button

```
Begin
    SelectedDB = DisplayOpenFileDialogue()
    SetTextBoxValue(DbTextBox, SelectedDB)
    OpenDatabase(SelectedDB)
    TableName = GetAllTablesInDatabase(DBName)
    FillListBox(TableListBox, TableName)
    TableFields = GetTableFiledS(TableName)
    FillListBox (FieldsListBox, TableFields)
    Disable(OpenDB Button)
End
```

4-1-3 On Click Add Association Button

```
Begin
    strCptProp = GetListBoxSelection(Cpt-Prop ListBox)
    Decompose(strCptProp, strCpt, strProp)
    strDBName = GetTextBoxValue(DBTextBox)
    strTable = GetListBoxSelection(TableListBox)
    strField = GetListBoxSelection(FieldListBox)
    if(strDBName Not Empty){
        CKBDBAssociation.DBName = strDBName
    }
    if(GetType (strCpt,strProp ) == GetType (strField) ) {
        cKbDb = new CKBDB(strCpt, strProp, strTable, strField)
        CKBDBAssociation.AddNode(strCptProp, cKbDb)
        SaveDocument()
    }
End
```

4-1-4 On Click Del Association Button

```
Begin
    GetFlexGridSelection(Cpt,Prop,Table,Field)
    CKBDBAssociation.RemoveNode(Cpt, Prop)
    SaveDocument()
End
```

4-1-5 On Change Selection Table List Box

```
Begin
    TableName = GetListBoxSelection(Table List Box)
    TableFields = GetTableFields(TableName)
    ClearListBox(FieldsListBox)
    FillListBox (FieldsListBox, TableFields)
End
```

5 CKBDB Player

This player is responsible for collecting the database field value according to the appropriate KEY and set this values to the concept property in the working memory. The following algorithm is the algorithm of that player

```
Begin

DBName = GetDbNameFromCKBDBAssociationClass()
If(Empty(DBName)) return
OpenDatabase(DBName)

While(Not End of CAssociationList){
    Node = GetFirstNode()
    Val = GetFromDB(DBName, Node.Table, Node.Field, strKEYJoin)
    SetToWM( Node.Cpt, Node.Prop, Val)
    Node = GetNextNode
}
CloseDB(DBName)
End
```