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

Detail Design for Diagnosis task Template

TR/CLAES/217/2001.5

By

Eng.Mohammed El Helly
Eng. Mohammed Yehia

June, 2001

1- Introduction

The objective of this report is to describe the template design of the interface, which is used in the diagnosis subsystem. Also the event handlers associated with each interface component are described in more details. This report consists of two Sections the Interface Design and Events in Section 2 and Database Interface In Section 3.

2- Interface Design and Events

The main interface of the diagnosis subsystem is shown in the following Figure

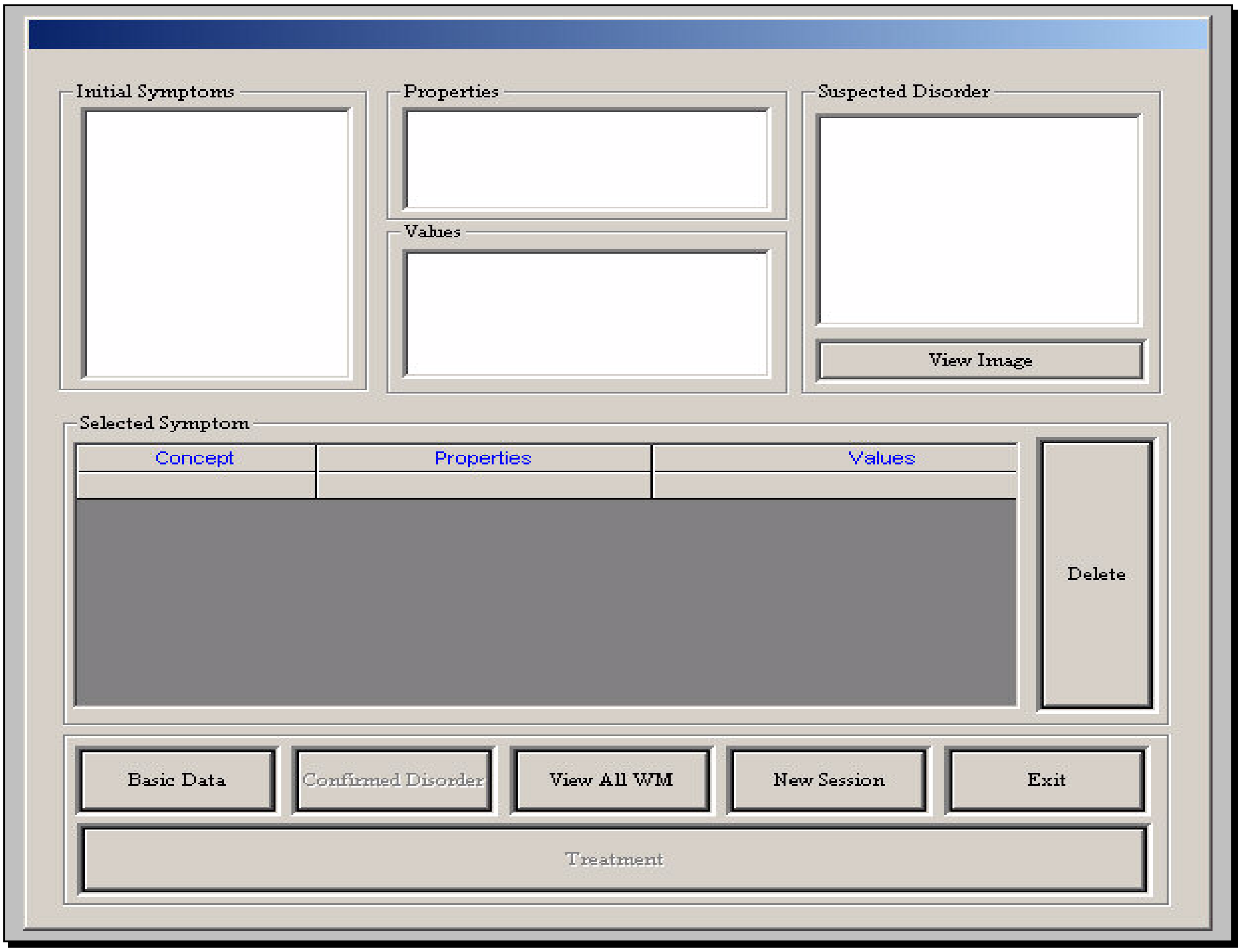


Figure 1

The above dialogue contains many controls like List Boxes, Flex Grid, and Push Buttons. Each control has many properties like label, contents, and etc. In addition many events for example on got focus, on selection change item, and etc. We can associate a method with any events.

Event used in diagnosis dialogue

The following are the events which is used in the diagnosis dialogue:

- On Init Dialoge.
- On Click BasicData Button.
- On Change Selection of Concept List.
- On Change Selection of Property List.
- On Double Click on Value List.
- On Click Confirmation Button.
- On View Working Memory Button.
- On Click New Session Button.
- On Click Delete Button.
- On Click View Image.
- On Click Exit Button

The following algorithms are the event handeler associated with the above events.

On Click Basic Data Button

```
Begin
        Initiate Working Memory ();
        Init Abduction List ();
        Clear Interface Element ();
        Display DB Dialog ()
        ExpandCaseDescription ();
        FillInterfaceWithinitialconcept();
        GrowthStage =GetValue("plant","growth_stage")
        If (GrowthStage != fruit)
                RemoveFruitConceptFromInterface();
        DisableTreatmentButton();
End
```

On Change Selection of Concept List

```
Begin
        CptName = GetCurrentSelectionFromCptList();
        PropList = GetPropertyFromKB(CptName);
        Clear Property ListBox();
        AddItems(PropList, PropListBox);
        Call OnChangeSelectionPropertyListBox();
End
```

On Change Selection of Property List

```
Begin
        CptName = GetCurrentSelectionFromCptList();
        PropName =  GetCurrentSelectionFromPropList();
        ValueList = GetLegalValueyFromKB(CptName,PropName);
        Clear ValueListBox();
        AddItems(,ValueList,  ValListBox);
End
```

On Double Click on Value List

```
Begin
    CptName = GetCurrentSelectionFromCptList();
    PropName =  GetCurrentSelectionFromPropList();
    ValueName = GetCurrentSelectionFromValueList();
    If (Not InWorkingMemory(CptName, PropName, ValueName){
        AddToWorkingMemory(CptName, PropName, ValueName);
        AddToFlexGrid(CptName, PropName, ValueName);
    }
    if(InFirstPhase()){
        PredictInference();
        SuspectedDisorder = GetSuspectedDisorderFromWM();
        ClearDisorderListBox();
        AddItems(SuspectedDisorder, DisorderListBox);
        EnableConfirmationButton();
    }
    else{
        RemovePreviousConfirmedDisorderFromWM();
        ConfirmInference();
        ConfirmededDisorder = GetConfirmedDisorderFromWM();
        ClearDisorderListBox();
        AddItems(ConfirmedDisorder, DisorderListBox);
        EnableTreatmentButton();
    }
    if(GetCount(ConfirmedDisorders)> 5)
        DisplayMessageBox("Not a real case to treat");
        DisableTreatmentButton();
    Else{
        EnableTreatmentButton();
    }
End
```

On Init Dialoge

```
Begin
    DisableTreatmentButton();
    DisableConfirmationButton();
    SetCptListBoxTilte("Initial Symptom");
    SetDisorderListBoxTilte("Suspected Disorder");
    InFirstPhase = TRUE;

End
```

On Click Confirmation Button

```
Begin
        DisableConfirmationButton();
        SetCptListBoxTilte("Additional Symptom");
        SetDisorderListBoxTilte("ConfirmedDisorder");
        InFirstPhase = FALSE;
        ClearDisorderListBox();
        InitAbductedList();
        AbductAllConfirmedDisorder();
        FillCptListBoxFromAbductedList();

End
```

On View Working Memory Button

```
Begin
        DisplayWorkingMemory();

End
```

On Click New Session Button

```
Begin
        DisableConfirmationButton();
        DisableTreatmentButton();
        SetCptListBoxTilte("Initial Symptom");
        SetDisorderListBoxTilte("Suspected Disorder");
        InFirstPhase = TRUE;
        ClearDisorderListBox();
        ClearCptListBox();
        ClearPropListBox();
        ClearValListBox();
        InitAbductedList();
        InitWorkingMemory();
        DisplayInitConcept();
End
```

On Click Delete Button

```
Begin
    CptName = GetCurrentSelectionFromCptList();
    PropName =  GetCurrentSelectionFromPropList();
    ValueName = GetCurrentSelectionFromValueList();
    if(InFirstPhase()){
        RemoveItemFromWM(CptName, PropName, ValueName);
        RemoveAllSuspectedDisorder();
        ClearDisorderListBox();
        RemoveItemFromFlexGrid(CptName, PropName, ValueName);
        PredictInference();
        SuspectedDisorder = GetSuspectedDisorderFromWM();
        AddItems(SuspectedDisorder, DisorderListBox);
        EnableConfirmationButton();
    }
    else{
        RemoveItemFromWM(CptName, PropName, ValueName);
        RemovePreviousConfirmedDisorderFromWM();
        ClearDisorderListBox();
        RemoveItemFromFlexGrid(CptName, PropName, ValueName);
        ConfirmInference();
        ConfirmededDisorder = GetConfirmedDisorderFromWM();
        AddItems(ConfirmedDisorder, DisorderListBox);
        EnableTreatmentButton();
    }
    if(GetCount(FlexGridRows)<=2){
            OnNewSession();
    }
End
```

On View Image Button

```
Begin
    CptName = GetCurrentSelectionFromDisList();
    Image = GetImageFromKB(CptName);
    DisplayImage(Image);
End
```

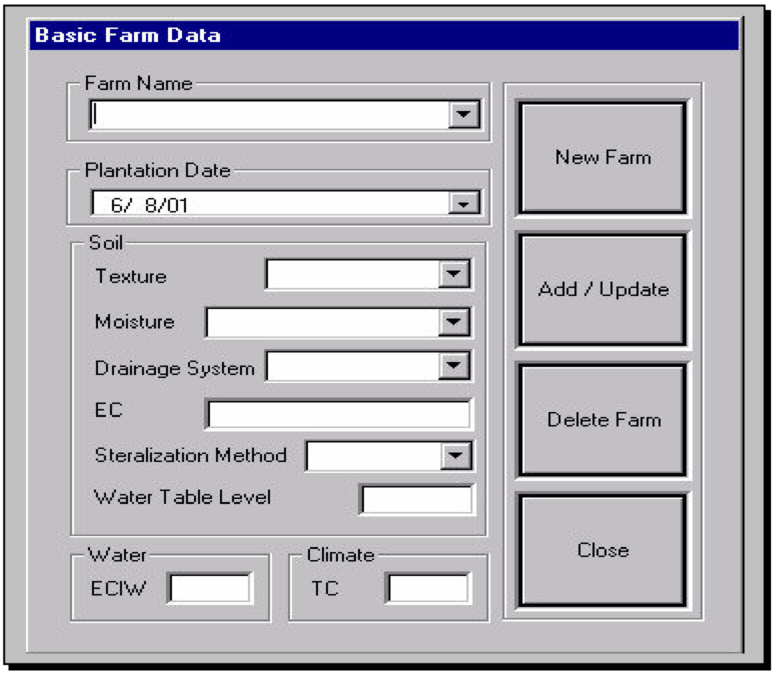
3- Database Interface

The database contains any data associated with plantation or farm and saved in separate file, which represent the static data for the farm. It should be remarked that any item in the DB is actually an item in the KB but the source of its value is a database.

For example:

Plantation date
Crop type
Soil type
….

The following database interface is the general interface for any diagnosis subsystem and it could be have any additional required field



Basic Farm Data

Plantation Date: 6/ 8/01

Figure 2

3-1 Event associated with DB dialog:

The following are the events which is used in the Database dialogue:

- On Initiate Dialog
- On click new farm button
- On click update farm button
- On click delete farm button
- On click close button
- On change selection of farm-id

There is a function associated for each of the above events, and we shall describe each function in much more details:

On Initiate Dialog

```
Begin
        for each property in the KB which has source of value equal to database
           if the type of this property is nominal or multi-value then
                add legal value of property in appropriate combo control
           end if
        end  for
        Show the  Current value of the selected plantation
End
```

On Click New Farm Button

```
Begin
        Reset all Control value
End
```

On Click Update Farm Button

```
Begin
    Collect the value from DB dialog;
    Validate the input data;
    if Valid data
        Create new raw in the database
        Add data to the table under plantation name as a key of the table;
    Load the input data in the working memory
End
```

On Click Delete Farm Button

```
Begin
    Collect the value from DB dialog;
    Validate the input data (none empty and between range);
    if Valid data
        Create new raw in the database
        Add data to the table under plantation name as a key of the table;
        load the input data in the working memory
End
```

On Click Close Farm Button

```
Begin
    if Valid data
        Close DB dialog ()
End
```

On change selection of farm-id

```
Begin
    FarmID = Get New Farm ID()
    DBobj =Lookup in DB(FarmID)
    Set Dbobj In Working Memory
End
```

4-Conclusion

The above template of the diagnosis subsystem is an event driven Concept and the designer may follow this steps to accomplish its interface and task design.

- Draw Dialog.
- Add different control specification.
- Determine event to be handled on each control.
- Write association method