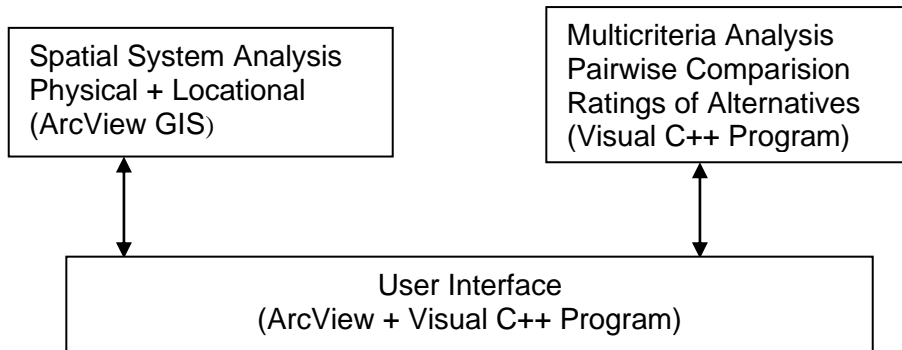


Software Architecture

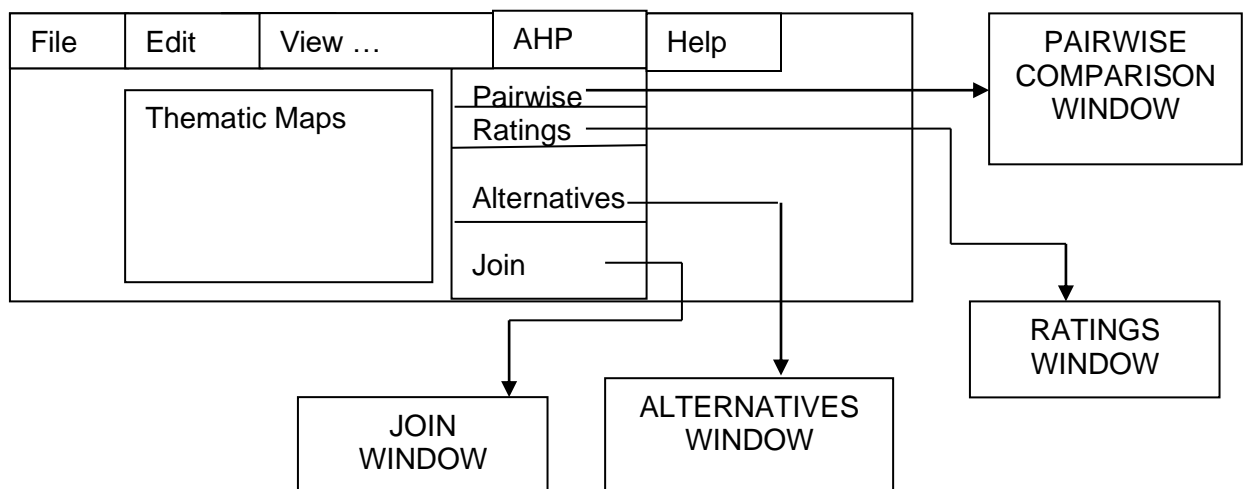
The ArcView GIS forms the driver software which invokes the User Interface written in C++.



AHP is created as a basic menu in the ArcView(ESRI,version 3.0). The sub menus of the AHP include Pairwise Comparison, Ratings, Alternatives and Join. These menus aid the user to determine (a)the relative weights of the suitability factors (b) the relative importance of sub-factors using a ratings intensity scale, and finally (c) the total suitability factor for a land use unit .Apart from these menus there is a menu to automatically update the thematic tables with the newly obtained suitability factor.

Pairwise Comparison is invoked by clicking the AHP menu in the main menu and selecting the Pairwise Comparison sub-menu. This initiates a call to the User Interface Package developed in C++(Microsoft Visual C++, version 6.0).See Appendix II. The user can perform the pairwise comparison of criteria and save the results. Ratings can be invoked by selecting Ratings sub-menu from the AHP main menu. The relative weights of the sub factors of each of the suitability factors can be determined and saved for future use. The Alternatives sub-menu when invoked enables the user to determine the composite weight of a land use unit to the user's satisfaction.

Abstract Navigation Feature:



Software Architecture

1. Pairwise Comparison: A new Dialog Box is created with a drop down list box .On selecting OK button a child dialog window is created with Edit Boxes where user can specify the Criteria names. Once OK button is pressed a series of pairwise comparison dialog boxes appear sequentially in which the user can compare one criteria with another. Finally the Consistency Index is shown. The user can either save the pariwise comparision or discard the changes depending on the Consistency Index.

2. Ratings: A new Dialog Box with an option of selecting an existing Ratings file or creating New Ratings comes up. On request for a new ratings the steps in the pairwise comparison is repeated for sub factors of the criteria shown. On selection of existing Ratings file the summary of all the factors and weights of their sub factors is shown. Here again the user has an option to save the Ratings carried out or discard the changes.

3. Alternatives: To carry out the Alternatives option Ratings must have been carried out first and the result must be stored in a ratings file. A Ratings file must be provided to compute Alternative.

4. Join: The Join Script provides a file dialog box for the user to select the alternatives file. Once the user selects the file the Avenue Script automatically updates the tables in GIS with the weights of the alternatives obtained from previous step.

One of these features (Alternatives) is shown in the figure -

ArcView GIS Version 3.0

File Edit View Theme Graphics Window **AHP** Help



Pairwise Ratings Alternatives Join

Scale 1: 789,253.91
302,623.70

View2

- Lrtstation.shp
- Onemile.shp
 - COMMERCIAL
 - PUBLIC
 - RESIDENTIAL
- Parcels.shp

Alternatives

Select a Ratings File



ALTERNATIVES

Criteria

| Land Use Unit | Mix Use | Road Net | Density | Prox Stat |
|---------------|---------|----------|---------|-----------|
| School[00] | Fine | Grid | Low | Near |

Total Score

Why C++: C++ is a general purpose programming language that provides flexible and efficient facilities for defining new constructs meaningful to an application domain.

C++ provides powerful support with libraries and documentation for implementing the AHP. Some flavors of C++ such as Visual C++ provides support for windows programming by providing the necessary support classes and other resources in the form of Microsoft Foundation Classes (MFCs) (See Microsoft Visual C++ , version 6.0, Library CDs 1, 2). The presence of built in classes enables the programmer to rapidly develop new applications. C++ is by far the most widely used language for application development.

Since C++ is an object oriented programming language and the GIS software also includes an object oriented programming language (e.g., Avenue Scripts in ArcView GIS) it is an effective programming language for the development of an integrated AHP/GIS application package.

The implementation of AHP is carried out using Microsoft Visual C++ version 6.0 on Windows NT 4.0. Microsoft Visual C++ provides built-in classes in the form of MFCs to create GUI based applications. Using the Microsoft Visual C++ AHP User Interface is developed .The development of user interface is made easy due to the existence of built-in classes like CDialog , CFileDialog .Once the user interacts with the AHP the results can be stored, updated, and retrieved in a GIS. The choice of C++ as the programming language is due to it's efficiency as both a user interface (AHP) as well as a powerful interface with a GIS.

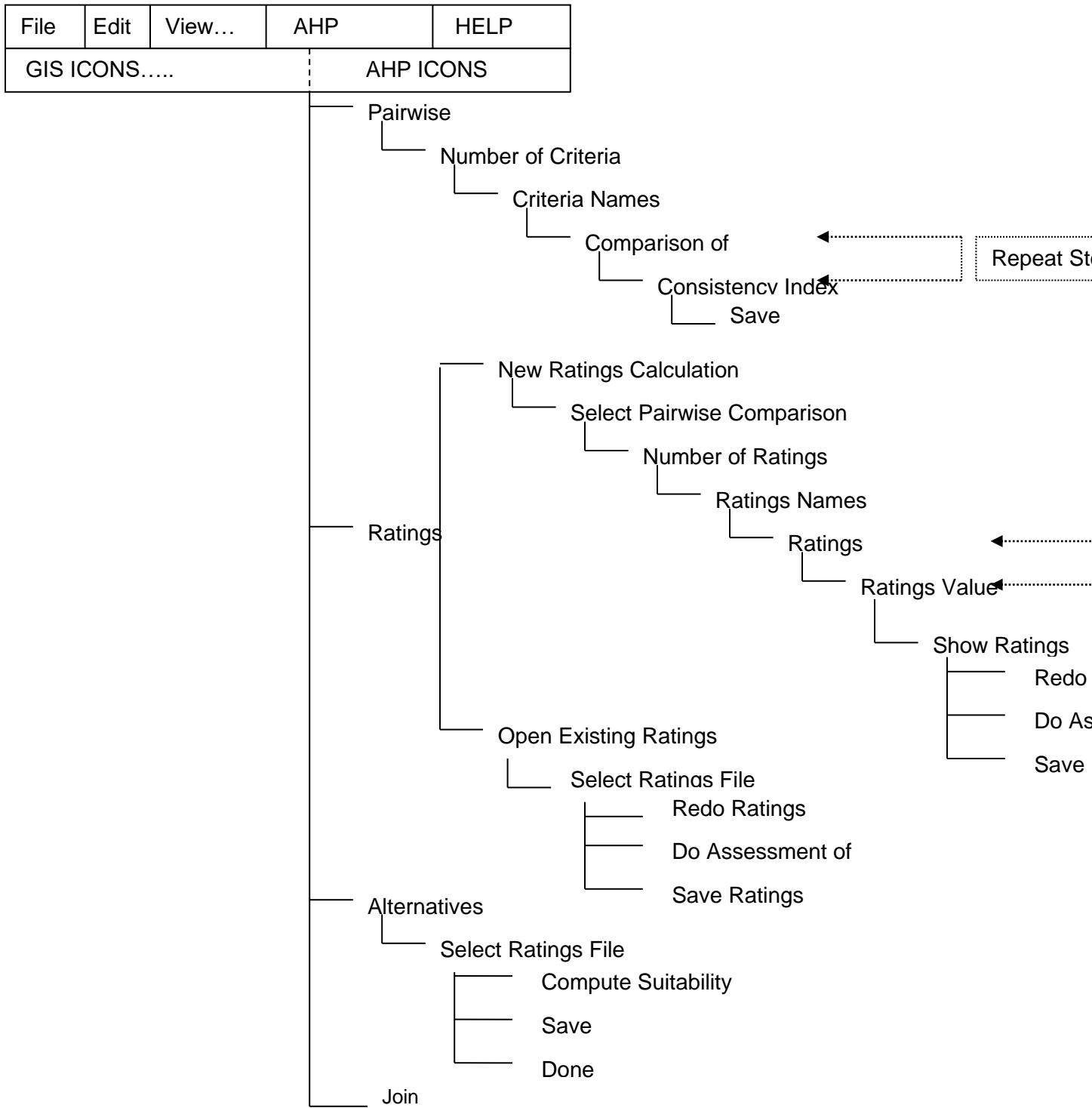


Fig ----- AHP/GIS Navigation Structure

References:

1. Avenue, Customization and Application Development for ArcView – ESRI,1998
2. Stroustrup, Bjarne. The C++ Programming Language, III Edition, Addison-Wesley Publishing Company, 1997.
3. Microsoft Developer Network Library, Visual Studio 6.0,CDROM I and II, 1998.

Appendix I. Script for ArcView GIS/AHP integration using Avenue Programming Language

The script to update the tables in the ArcView GIS (version 3.0) using Avenue programming language with the weight information obtained from the AHP is shown below. The script creates a new temporary table with two fields in it, namely ID and Weight. This table is joined with the GIS table automatically using the "one to many join". The tables in the GIS are updated with the new field called Weight. This weight information is used by the GIS software to produce the new thematic maps. The actual script is as follows:

```
f =TextFile.make("c:\temp\tmp.txt".AsFileName,#FILE_PERM_READ)
wordList = f.Read(f.getSize).AsList
f.close
```

```
theVTab =VTab.MakeNew("c:\temp\xyz.dbf".AsFilename,dbase)
```

```
weightTable = Table.make(theVTab)
weightTable.SetName("MyName")
'weighttable.GetWin.Open
f1 = Field.make("Landuse",#FIELD_CHAR,10,0)
f2 = Field.Make("Weight",#FIELD_FLOAT,10,6)
theVTab.AddFields({f1,f2})
```

```
rec = theVTab.AddRecord
word = wordList.get(0)
MsgBox.Info(word,"")
theVTab.SetValue(f1,rec,word.AsString)
word = wordList.get(1)
theVTab.SetValue(f2,rec,word.AsNumber)
MsgBox.Info(word,"")
```

```
rec = theVTab.AddRecord
word = wordList.get(2)
MsgBox.Info(word,"")
theVTab.SetValue(f1,rec,word.AsString)
word = wordList.get(3)
theVTab.SetValue(f2,rec,word.AsNumber)
MsgBox.Info(word,"")
```

```
rec = theVTab.AddRecord
word = wordList.get(4)
MsgBox.Info(word,"")
theVTab.SetValue(f1,rec,word.AsString)
word = wordList.get(5)
theVTab.SetValue(f2,rec,word.AsNumber)
MsgBox.Info(word,"")
```

```
myTable=av.GetActiveDoc
str=myTable.GetName
toTable= av.GetProject.FindDoc(str)
```

```
fromTable = av.GetProject.FindDoc("MyName")
toVT=toTable.GetVTab
fromVt=fromTable.GetVTab
toField=toVt.FindField("Landuse")
fromField =fromVt.FindField("Landuse")
toVt.Join(toField,fromVt,fromField)
```