PB2002-101951

PREDICTING CRASHES FROM INCREASED SIGNALIZATION: *PROTOTYPE SOFTWARE FOR CORRIDOR PLANNING*

Final Report

by

Sangjun Kim
Undergraduate Research Assistant

Report No. UVA/29472/CE01/105

Center for Transportation Studies
University of Virginia
Charlottesville, VA 22904

Prepared for
Mid-Atlantic Universities Transportation Center

August 2001

Disclaimer

Technical Report Documentation Page

1. Report No. UVA/29472/CE01/105	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Predicting Crashes from Increased Signalization: Prototype Software for Corridor Planning		5. Report Date June 2001
		6 Performing Organization Code III-0104
7. Author(s) Sangjun Kim		8. Performing Organization Report No. UVA/29472/CE01/105
9. Performing Organization Name and Address Center for Transportation Studies University of Virginia PO Box 400742 Charlottesville, Virginia 22904-4742		10. Work Unit No. (TRAIS)
		11.Contract or Grant No.
12. Sponsoring Agency Name and Address Virginia Department of Transportation FHWA 1401 E. Broad St. PO Box 10249 Richmond, VA 23219 Richmond, VA 23240		13. Report Type and Period Covered Final
		14. Sponsoring Agency Code
15. Supplementary Notes		
16. Abstract		

As a corridor evolves or develops, pressure is often places on transportation administrators to increase the number of signalized intersections. In some cases, these signals may be justified based on Manual on Uniform Traffic Control Devices (MUTCD) warrants or other types of analyses. In other cases, however, the signals may be providing access that should be provided by other methods. A common example is the use of two signalized access points instead of having only one signalized access point along with a judicious sharing of driveways. Continuing with this example, it could be that fewer crashes will occur yet sufficient amount of access has been provided- under the second scenario of only one signal. There are several models that can predict crash rate depending on various roadway conditions, daily traffic volume, signalized access density, mathematical complexity, they can be time consuming for administrators to apply in real-world situations. In order to bridge the gap between the user requirements of transportation administrators and the computational requirements of these detailed models, a user-friendly software package has been developed that automates these computations. The executable software application developed in this study can be used to estimate the number of crashes a corridor will experience as a function of changes in average daily traffic (ADT), signalized access, unsignalized access, and median treatment. The software implements models developed by others, as described in the pages that follow.

17. Key Words	18. Distribution Statement No restrictions. This document is available to the public through NTIS, Springfield, VA 22161		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 28	22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

ACKNOWLEDGMENTS

This study was conducted under the sponsorship of the United States Department of Transportation through a grant to the Mid-Atlantic Universities Transportation Center and also received funding from the Virginia Transportation Research Council. The author acknowledges the sponsors' support gratefully.

The author would like to thank John Miller, research scientist at Virginia Transportation Research Council, and Lester Hoel, L.A. Lacy Distinguished Professor of Engineering at University of Virginia, for their assistance with the project. The author also would like to thank Frances Adkins, Kendall Drummond, Rod Turochy, Jared Ulmer, and Don Wells from Virginia Transportation Research Council who provided invaluable feedback in software evaluation process.

TABLE OF CONTENTS

ABSTRACT

As a corridor evolves or develops, pressure is often placed on transportation administrators to increase the number of signalized intersections. In some cases, these signals may be justified based on Manual On Uniform Traffic Control Devices (MUTCD) warrants or other types of analyses. In other cases, however, the signals may be providing access that should be provided by other methods. A common example is the use of two signalized access points instead of having only one signalized access point along with a judicious sharing of driveways. Continuing with this example, it could be that fewer crashes will occur yet sufficient amount of access has been provided- under the second scenario of only one signal. There are several models that can predict crash rate depending on various roadway conditions, daily traffic volume, signalized access density, median treatment, and unsignalized access density. However, because of their mathematical complexity, they can be time consuming for administrators to apply in real-world situations. In order to bridge the gap between the user requirements of transportation administrators and the computational requirements of these detailed models, a user-friendly software package has been developed that automates these computations. The executable software application developed in this study can be used to estimate the number of crashes a corridor will experience as a function of changes in average daily traffic (ADT), signalized access, unsignalized access, and median treatment. The software implements models developed by others, as described in the pages that follow.

INTRODUCTION

Access management techniques are often controversial because they are believed to affect roadway safety, congestion, and the number of trips made to adjacent businesses. Accordingly when transportation administrators make decisions regarding signal spacing or median treatments, they want to show developers, concerned citizens, and elected officials what the expected impact that such decisions may have on roadway safety. To give administrators tools to quantify the safety impacts of access management techniques, researchers have developed mathematical models based on regression methods that quantify expected crashes or crash rates for various access management strategies. There are several models that can predict crash rates depending on various roadway conditions such as daily traffic volume, signalized access density, median treatment, unsignalized access density (e.g. commercial driveways), land use, left-turn availability, and speed. Five promising models are considered in this study (Miller, Hoel, Kim, Drummond; 2001). They are:

- Model 1: *Joint Transportation Research Program* (JTRP) report published in 1998 by H. C. Brown, S. Labi, A. P. Tarko, and J. D. Fricker.
- Model 2: This is a combined model of two submodels. An accident prediction model in *Transportation Research Record 1581*(TRR) by J. A. Bonneson and P. T. McCoy is the first submodel and the second one is published by B. Persaud and T. Nguyen in *Transportation Research Record 1635* (TRR).
- Model 3: J. Gluck, H. S. Levinson, and V. Stover's model published by *National Cooperative Highway Research Program 420* (NCHRP) in 1999.

- Model 4: A model (Guidelines For Commercial Driveway Spacing On Urban And Suburban Arterial Roads) developed by N. Garber and T. White that published as a Mid-Atlantic Universities Transportation Center Report, Charlottesville, VA. in 1995.
- Model 5: A model developed by H. Preston, D. Keltner, R. Newton, and C. Albrecht that uses univariate regression method, published by Minnesota Department of Transportation (MNDOT) in 1998.

Two other software applications are also suitable for access management. They are:

- TRAF-SAFE by A. R. Kaub: This is a commercially available software package. This software evaluates the safety impacts of various access management strategies.
- NCHRP 420 by Gerry Gluck: This software projects future number of crashes based on known past number of crashes. This safety analysis software package was developed from the model published by *National Cooperative Highway Research Program 420* (NCHRP) in 1999.

II. PROBLEM STATEMENT

Although, mathematical models may be able to predict crash rate accurately, a manual application of these models can be time consuming in many cases. Therefore, a user-friendly software package is necessary to realistically implement these needs.

The goal of this project was therefore to develop this new software. This project automated Models 1, 4, and 5 as shown previously. Model 2 and 3 were excluded from this effort, as model 2 requires extensive data entry. (For example, if a user wants to conduct a study of a five mile corridor which has 10 signalized access, the user needs to make ready to enter more than 30 pieces of data such as each sections' AADT, entering volume of each intersections, median treatment of each sections, and much more.) Model 3 has been previously automated by the original developer. Thus, the remainder of this report deals with Models 1, 4, and 5 only. The software program AccuPack is developed as explained in the following section.

III. SOFTWARE DEVELPMENT APPOARCH

A. Five Key Steps In Creating The Software

Step 1. *Create a requirements document,* where an exhaustive identification of all the tasks the software would accomplish was documented. Admittedly, these requirements were modified as the project progressed, but this document provided a firm blueprint for the goals of the project. Requirements were set across three areas: user interface, paths of processing, and output. The step to determine how the screen would look and what variables it would accept were included in setting up the user interface requirement. This was accomplished by getting feedback from staffs. The path processing step was used to determine which models were used, and how the computations were done for the study and how data was passed among modules. How data and results were archived for future modification was considered in the design of the output. In addition to what the user sees, the storage of data entered for future reference was critical, because the software needs to store and access the data for future use.

Step 2. *Create an initial design.* The flow of control and data was outlined in the form of an algorithm, independently of a specific programming language. Again, as was the case with Step 1, this "design" was modified as the project progressed, but it gave an overview of the project's scope before delving into the writing of code.

Step 3. *Write code in Visual Basic.* A sample of the code is in Appendix A, and the complete file is on the accompanying compact disk.

Step 4. *Conduct testing.* Five Virginia Transportation Research Council staff members evaluated the software. Feasible improvements that were suggested such as grammatical changes and error handling - were implemented. Promising modifications that could not be accomplished within the project's time frame are listed at recommendation of Appendix B.

Step 5. *Make enhancements to the software.*

- Error handlers were added. For example, when a user entered zeros in data input stages, AccuPack displayed an error message and terminates the current process. To minimize these kinds of error messages, a few comparable modifications were made.
- A chart option for each calculation was added to the software package to expand the AccuPack's usability. The chart shows users the calculated results. With the capability of creating a chart using one data set at each time, a user can easily find the pattern of the safety impact resulting from different roadway conditions. For example, if a user tries to determine how the installation of signals impacts the roadway safety, he or she selects the "generating a chart option". The package shows the accident curve that varies by number of signals. (See the Figure 6 on page 13)

B. Summary of AccuPack's Functionality

This software package, AccuPack, allows the user to apply three different models to estimate the crash rate. AccuPack also compares the expected result with historical data and then modifies the result if there is a discrepancy. When Model 1 was applied to the three study corridors, the average difference between actual number of crashes and computed crashes was about 30 percent. Thus, including this "proportional computation" was necessary to increase the software's usability.

As an example of this proportional computation, suppose that a corridor had 200 actual crashes in 1999 and that a local transportation administrator wants to know how the crash rate would increase if two signals were added to the section. Using Model 1, the crashes are computed as 138.1 crashes. In this case the result is not really useful because of the error. Accordingly, a site specific modification - as recommended by NCHRP 420 - is employed:

$$\text{Original Crashes}\left(\frac{\text{Computed Crashes with Future Data}}{\text{Computed Crashes with Original Data}}\right) = \text{Future Crashes}$$

or $$200\left(\frac{138.1}{131.7}\right) = 209.7 \text{ Crashes with adding two signal lights}$$

More specific examples and detailed instruction of this software usage can be found in section IV.

AccuPack is suitable for resident engineers, transportation planners, or other persons charged with managing corridor safety, AccuPack requires approximately 15 - 20 minutes per corridor for data entry and analysis. The following section provides a step-by-step user manual for AccuPack.

IV. AccuPack User's Manual

1. This is a self- explanatory program and designed to be easy to follow. When a user starts the program, AccuPack's introduction interface explains the objective of this software and asks if the user wants to continue.

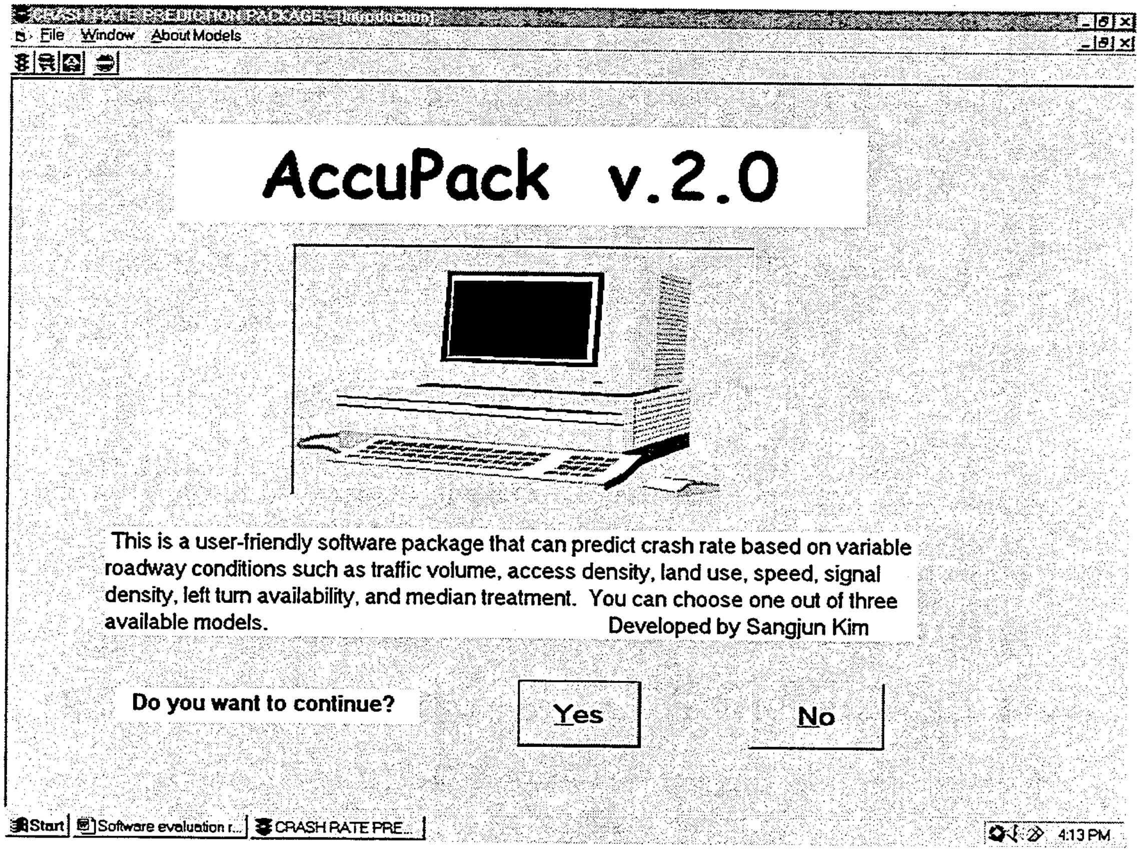

Figure 1. The introduction screen

2. If the user clicks "yes", the program proceeds to the next step where the user can choose one of three available models. This is the general selection screen and it explains each model using pop up message boxes. After the user decided which model to go with reading the message boxes, he or she clicks the "start" button next to the model the user chose.

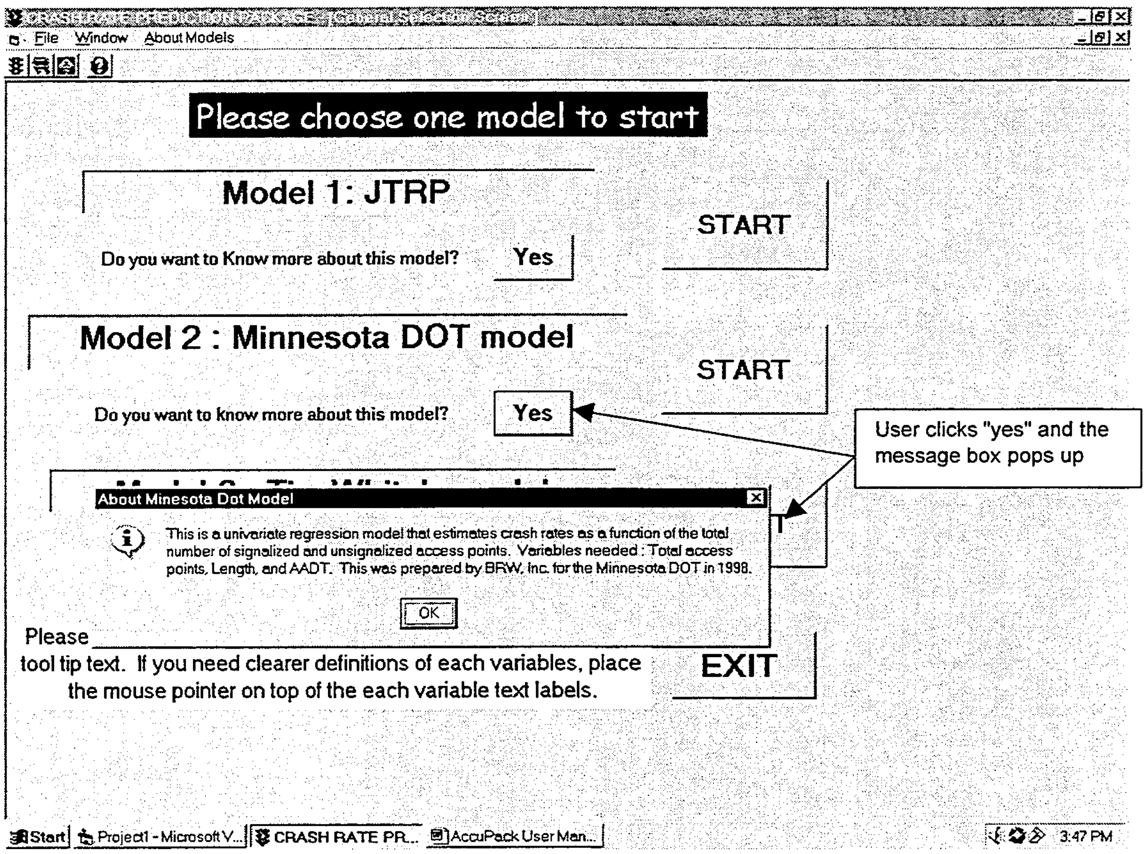

Figure 2. The general selection screen

3. The next step is to compute the crash rate. This step requires two sets of data for the same segment. When these models were applied to testing corridors without site specific modification, the average difference between actual number of crashes and computed numbers ranged from 30 percent to several hundred percent. Thus a site specific modification, known also as a proportional modification based on a computation of actual crashes and predicted crashes using historical data, often can bring increased reliability approaching error rates of approaching 28 to 30 percent.

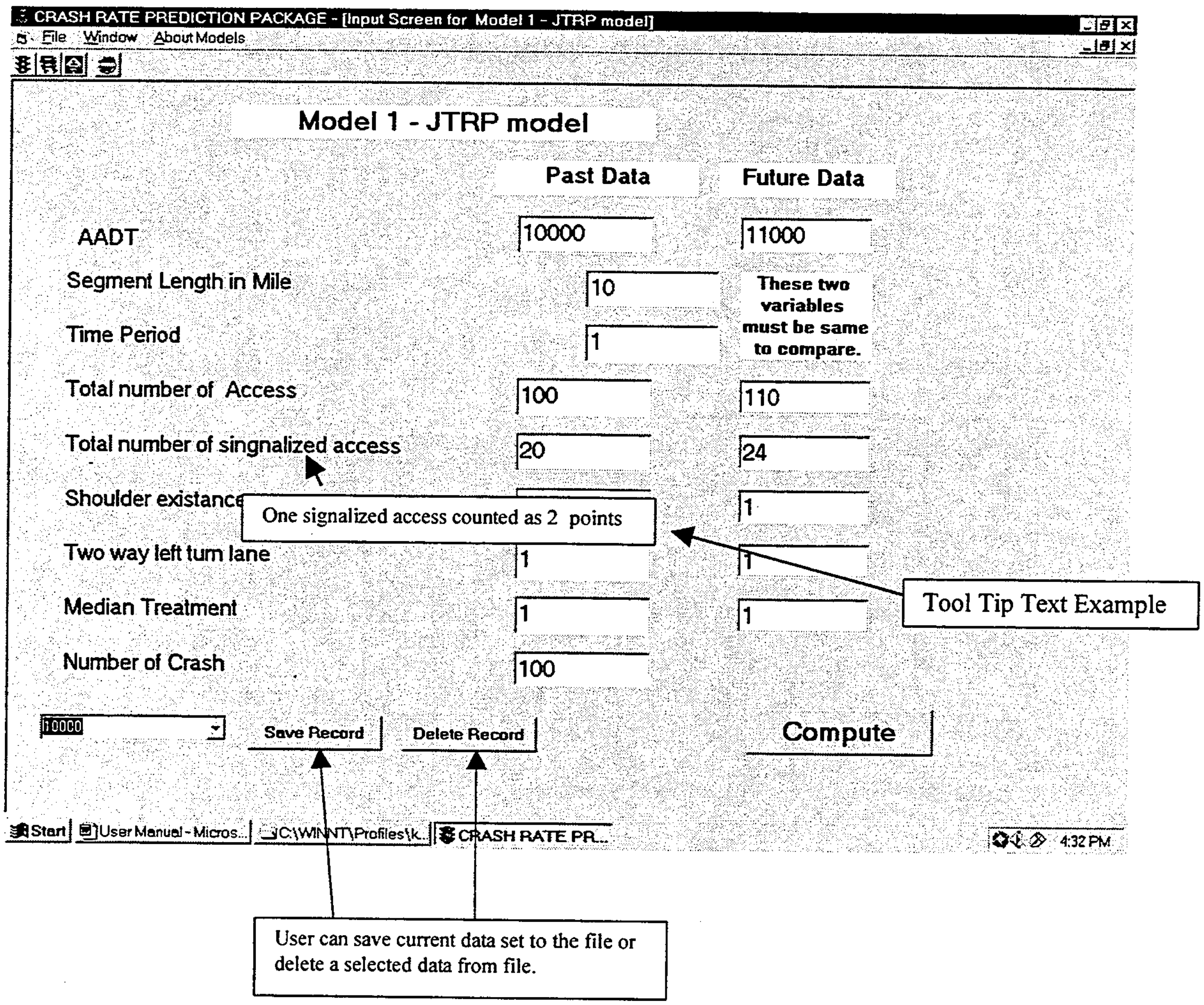

Figure 3. A sample data input screen

4. Before the user clicks the "compute" command button, he or she could save the current data set to a file on the hard drive. The saved file can be easily recovered any time during the current run of software or in the future. The feature also has "delete" command button that can erase the saved data set.

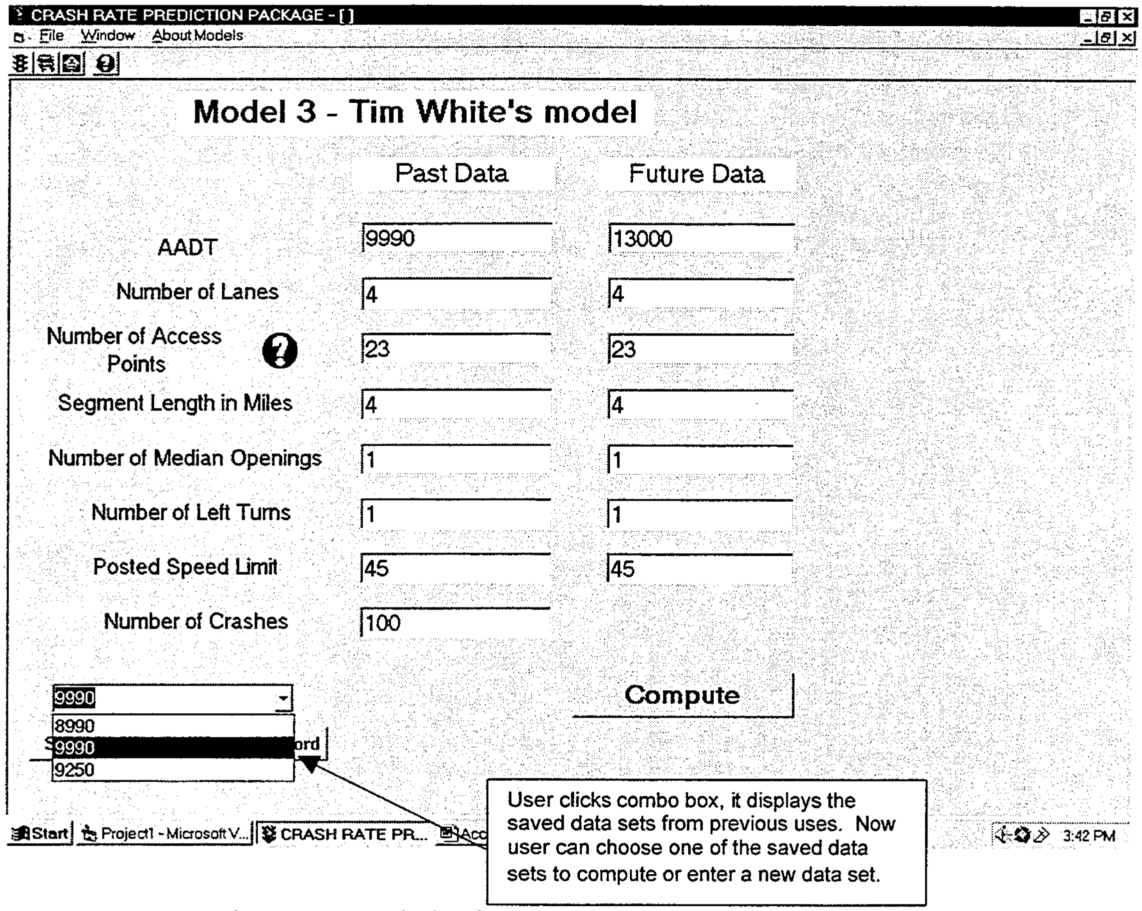

Figure 4. A sample data input screen explaining combo box feature

5. When all data requirements are filled, AccuPack is ready to compute the crash rate, the number of crashes, and the modified number of crashes. If the user wants to compute another corridor or same corridor with different data, he or she could click "Go back for another Computation" and AccuPack goes back to the model's input screen. If the user wants to compute a crash rate with another model, he or she could choose "Compute Crash Rate with another Model" and AccuPack leads the user to the general selection screen, or he or she could simply terminate AccuPack by clicking the "Exit" button. Many of the command buttons and labels are featured with 'Tool Tip Text' that appears when the pointer is placed over. When the user needs more information or needs to clarify each button or label, tool tip text explains the requirement or gives more information.

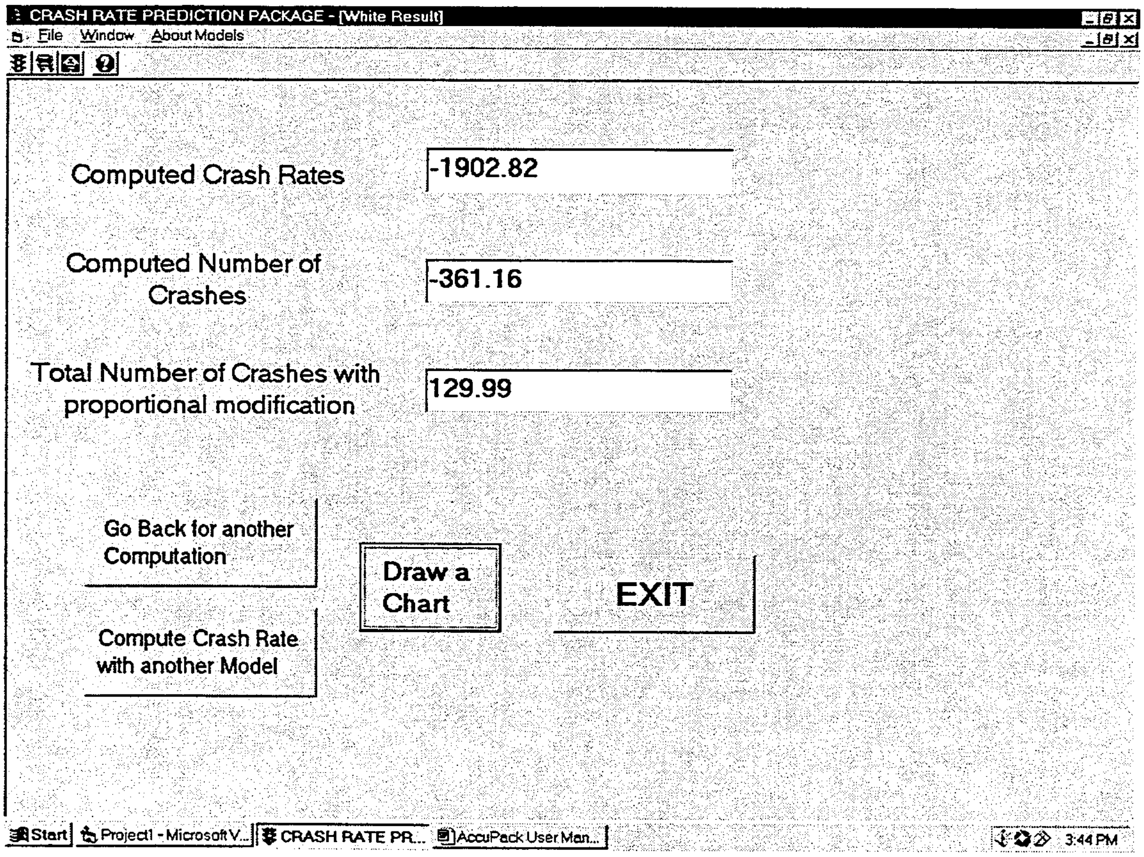

Figure 5. A sample result screen

6. The Chart option enhances the understanding of the crash rates pattern by enabling users to see the model trends: how do crashes increase of the number of accesses increase?

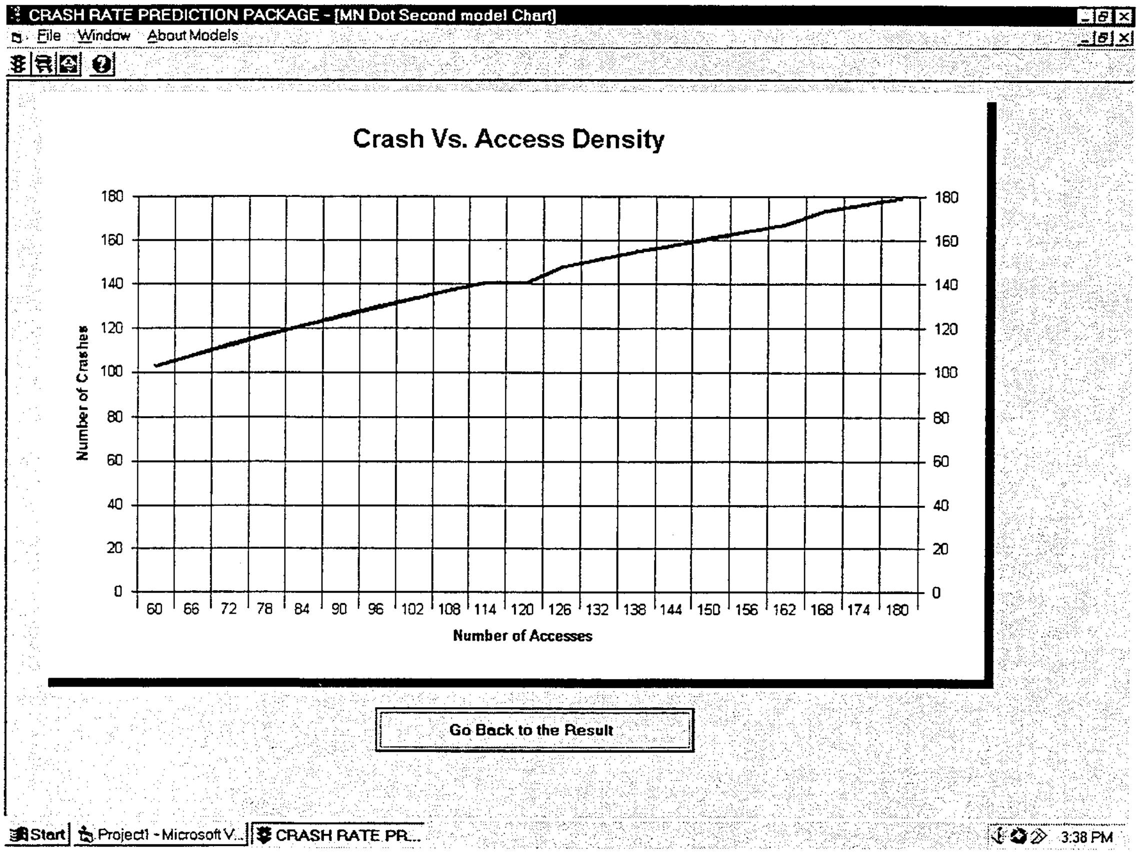

Figure 6. A chart example

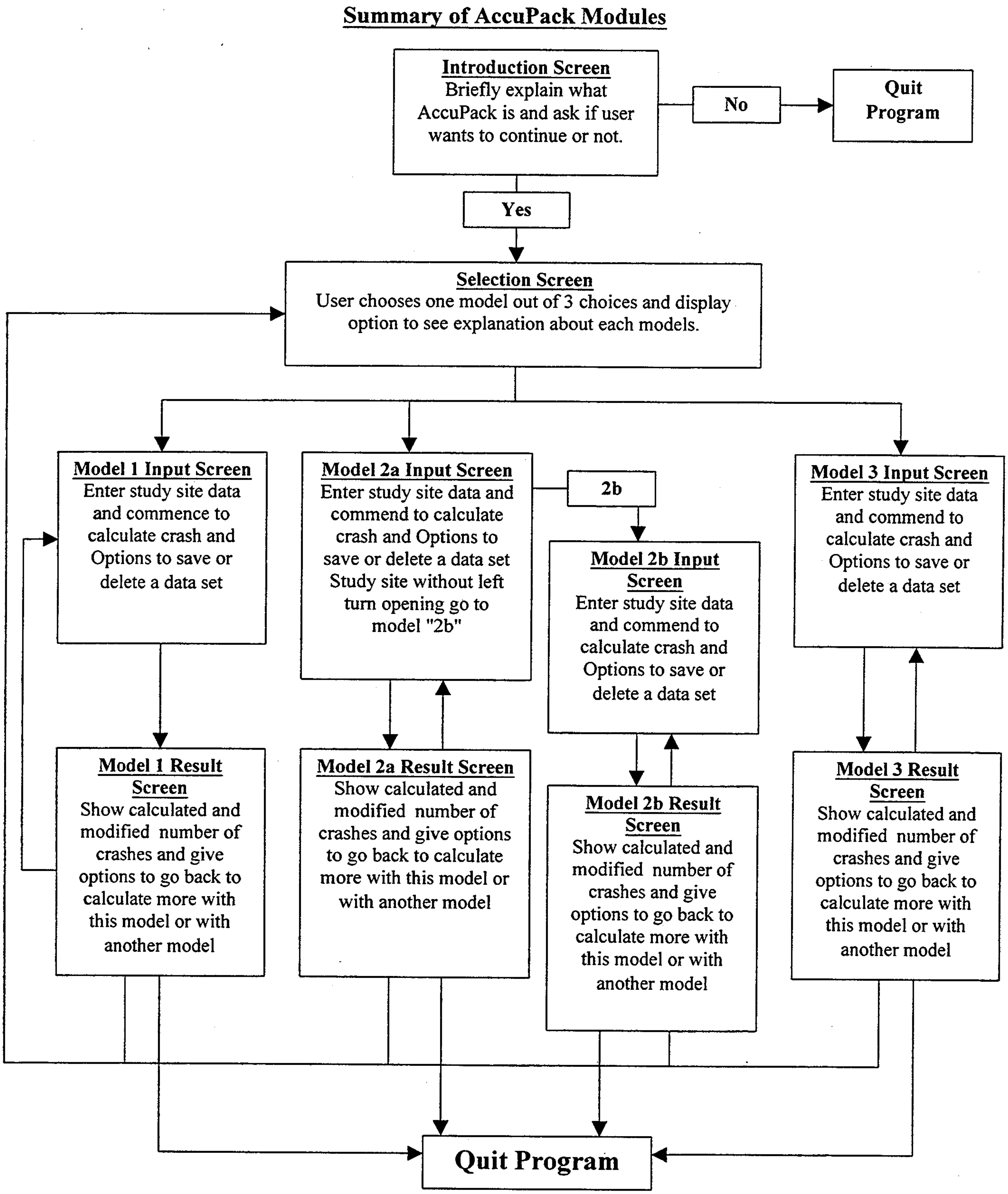

Figure 7. AccuPack operation diagram

V. EPILOGUE

This project was driven by the fact that the "typical" transportation professional - if that term can be used - is exceedingly busy. Finding engineers who have corridor management responsibilities to field test the AccuPack was a challenge simply because they - like many administrators - have quite hectic schedules. Clearly software that wades through complex subroutines and complex data entry requirements is inappropriate. Software developers in this area need to focus on the simplicity of application. Even simplification requires some assumptions or elimination of detailed information, software packages should be easy to apply and meaningful for use in practice. This is not to say that transportation professionals are unlikely to apply new technology, rather, it is clear that time-saving approaches, such as these types of corridor analyses, must be as efficient as possible with respect to demands placed on practitioners' time.

Bibliography

AASHTO. A Policy on Geometric Design of Highways and Streets. Washington, D.C., 1994

Bonneson, James A., and Patrick T. McCoy. "Effect of Median Treatment on Urban Arterial Safety." Transportation Research Record 1581. Washington, D.C., 1997. 27-36.

Brown, Henry C., Samuel Labi, Andrzej P.Tarko, and Jon D. Fricker. A Tool For Evaluation Access Control on High-Speed Urban Arterials. Indiana DOT and Purdue University, West Lafayette, Indiana, 1998.

Federal Highway Administration. Highway Safety Performance-1992: Fatal and Injury Accident Rates on Public Roads in the United States. U.S. Department of Transportation. Washington, D.C. 1995.

Gluck, Jerome, Herbert S. Levinson, and Vergil Stover. NCHRP report 420: Impact of Access Management Techniques. Transportation Research Board, National Research Council. Washington, D.C., 1999.

Garber, Nicholas J., and Timothy E. White. Guidelines For Commercial Driveway Spacing On Urban And Suburban Arterial Roads. Mid-Atlantic Universities Transportation Center Report. Charlottesville, VA. 1995.

Miller, John S., Lester A. Hoel, Sangjun Kim, and Kendall P. Drummond. Evaluating Methods to Quantify the Safety Impacts of Access Management: Are Current Models Transferable? Virginia Transportation Research Council. Charlottesville, VA. 2000.

Persaud, Bhagwant, and Thu Nguyen., "Disaggregate Safety Performance Models for Signalized Intersections on Ontario Provincial Roads." Transportation Research Record 1635. Washington, D.C., 1998. 113-126.

Preston, Howard, Deanna Keltner, Randy Newton, and Chris Albrecht. Statistical Relationship Between Vehicular Crashes and Highway Access. BRW, Inc., Minneapolis, MN. for Minnesota Department of Transportation. Minneapolis, MN: 1998.

Schneider, David I., QBASIC with an Introduction to Visual Basic 5.0. Upper Saddle River, NJ: Prentice Hall, 1998.

Spenik, Mark, Andrew J. Indovina, Pierre Boutquin, and David Jung., Visual Basic Interactive Course. Corte Madera, CA: Waite Group Press, 1998.

Urbanik, Thomas II. NCHRP Synthesis 259: Management of Surface Transportation System. Transportation Research Board, National Research Council. Washington, D.C., 1998.

Virginia Department of Transportation. 1998 Virginia Traffic Crash Facts. Richmond, VA. 1999.

Appendix A

Description of Code

[The complete code is included on the CD, but a brief overview of selected modules are presented in order to assist future developers who may modify this product. Actual code is written in italic.]

- Introduction Screen

When user clicks the "Command1" button, AccuPack loads 'Selection Screen' and unloads 'Introduction Screen'. SetFocus moves the focus to the object (Selection Screen). End Sub terminates the event procedure of this sub program.

```
Private Sub Command1_Click()
  SelectionScreen.Show
  SelectionScreen.SetFocus
  Unload Intro
End Sub
```

When the 'Command2' button is clicked, AccuPack terminates whole program.

```
Private Sub Command2_Click()
        End
End Sub
```

- Selection Screen

When the command buttons are clicked in 'Selection Screen', AccuPack loads and moves focus on the object (e.g. highlights it on the screen and activates appropriate subroutines) but does not unload 'Selection Screen' for the instances where the user wants to choose another model yet use the same data.

```
Private Sub Command1_Click()
  JTRPInput.Show
  JTRPInput.SetFocus
End Sub

Private Sub Command2_Click()
  frmMNDot.Show
  frmMNDot.SetFocus
End Sub

Private Sub Command7_Click()
  frmWhiteInput.Show
```

```
  frmWhiteInput.SetFocus
End Sub
```

When user wants to know more about each model, he or she clicks the 'Command3' button and AccuPack loads the message box.

```
Private Sub Command3_Click()
        MsgBox "This is a multivariate regression model that estimates " & _
        "absolute crashes as a function of total number of signalized and" & _
        "unsignalized access points, the percentage of signalized access" & _
        "points, the presence of a shoulder, and the median type." & _
        "Variables needed:access points(both signalized and unsignalized)," & _
        "existance of shoulder, median type(divided, two way left turn lane," & _
        "or undivided). This was published by Indiana DOT and Purdue
        University in 1998.", vbInformation, "About JTRP Model "
End Sub
```

- Model 1 (JTRP) Input Screen

This sub program assigns the data type for the 'Combo box' which enables the user to add and delete data.

```
Private Sub cboData_Click()
   Dim RecNum As Integer
   Dim i As Integer
      RecNum = cboData.ItemData(cboData.ListIndex)
      Get #1, RecNum, Record
   With Record
      txtAADT.Text = .AADT
      txtLength.Text = .Length
      txtYears.Text = .Year
      txtAccess.Text = .Access
      txtSignals.Text = .Signal
      txtShoulder.Text = .Shoulder
      txtTWLTL.Text = .TWLTL
      txtMedian.Text = .Mediam
      txtPastNoCrash.Text = .PastNoCrash
      Text1.Text = .ADDT1
      Text4.Text = .Access1
      Text5.Text = .Singal1
      Text6.Text = .Shoulder1
      Text7.Text = .TWLTL1
      Text8.Text = .Mediam1
   End With
End Sub
```

When user wants to save current data set to be used later, the user clicks 'Add' button and AccuPack save it to a temporary file on the user's hard disk.

```
Private Sub cmdAdd_Click()
    Dim NumOfRec As Integer
    NumOfRec = LOF(1) / Len(Record)
     With Record
     .AADT = txtAADT.Text
     .Length = txtLength.Text
     .Year = txtYears.Text
     .Access = txtAccess.Text
     .Signal = txtSignals.Text
     .Shoulder = txtShoulder.Text
     .TWLTL = txtTWLTL.Text
     .Mediam = txtMedian.Text
     .PastNoCrash = txtPastNoCrash.Text
     .ADDT1 = Text1.Text
     .Access1 = Text4.Text
     .Singal1 = Text5.Text
     .Shoulder1 = Text6.Text
     .TWLTL1 = Text7.Text
     .Mediam1 = Text8.Text
   End With
   Put #1, NumOfRec + 1, Record
 ComboUpdate
End Sub
```

When the user chooses to delete certain current data set that are saved in temporary file, the user clicks the 'Delete' button; then AccuPack deletes the data set from the temporary file on the computer's hard disk.

```
Private Sub cmdDelete_Click()
    Dim CurRecNum As Integer
    Dim i As Integer
    Dim j As Integer
    CurRecNum = cboTest.ItemData(cboTest.ListIndex)
    Get #1, CurRecNum, Record
    Record.boolDeleteRecord = True
    txtAADT.Text = ""
    txtLength.Text = ""
    txtYears.Text = ""
    txtAccess.Text = ""
    txtSignals.Text = ""
    txtShoulder.Text = ""
    txtTWLTL.Text = ""
    txtMedian.Text = ""
    txtPastNoCrash.Text = ""
```

```
Text1.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
Put #1, CurRecNum, Record
Open "C:\temp\testFileTemp.Rnd" For Random As #2 Len = Len(Record)
j = 1
For i = 1 To LOF(1) / Len(Record)
Get #1, i, Record
If Record.boolDeleteRecord = False Then
Put #2, j, Record
j = j + 1
End If
Next
Close #1
Close #2
Kill "C:\temp\testFile.Rnd"
Name "C:\temp\testFileTemp.Rnd" As "C:\temp\testFile.Rnd"
Open "C:\temp\testFile.Rnd" For Random As #1 Len = Len(Record)
ComboUpdate
End Sub
```

This function shown below notifies the user that text boxes are needed to be filled by numbers that are greater than 0 before the user can proceed to the next step.

```
Private Sub Command1_Click()
If Val(txtAADT.Text) <= 0 Then
        MsgBox "Please enter values greater than 0 to continue"
        txtAADT.Text = ""
        txtAADT.SetFocus
ElseIf Val(Text1.Text) <= 0 Then
        MsgBox "Please enter values greater than 0 to continue"
        Text1.Text = ""
        Text1.SetFocus
        .
        .
        .
ElseIf Val(txtSignals.Text) <= 0 Then
        MsgBox "Please enter values greater than 0 to continue"
        txtSignals.Text = ""
        txtSignals.SetFocus
Else
        Unload frmJTRPresult
        Load frmJTRPresult
        frmJTRPresult.Show
```

```
            frmJTRPresult.SetFocus
    End If
    End Sub

Private Sub Form_Load()
    Open "C:\temp\testFile.Rnd" For Random As #1 Len = Len(Record)
    ComboUpdate
End Sub

Private Sub ComboUpdate()
  Dim i As Integer
  cboTest.Clear
  For i = 1 To LOF(1) / Len(Record)
    Get #1, i, Record
      cboData.AddItem Record.AADT
      cboData.ItemData(cboData.NewIndex) = i
  Next
End Sub
```

- Model 1 (JTRP) Result Screen

This is the option to go back to the "Model1" input screen to compute crashes from another data set.

```
Private Sub Command1_Click()
    JTRPInput.Show
   JTRPInput.SetFocus
   Unload frmJTRPresult
End Sub
```

This option can take the user to the 'Selection Screen' to choose another available model to obtain crash rate and the number of crashes.

```
Private Sub Command2_Click()
   SelectionScreen.Show
   SelectionScreen.SetFocus
   Unload frmJTRPresult
End Sub
```

This sub program terminates the program.

```
Private Sub Command3_Click()
   Close #1
   End
End Sub
```

Shown below is sample code that computes the crash rate, the number of crashes, and the number of crashes with proportional modification.

```
Private Sub Form_Load()
        Text1.Text = Format((0.494 * (Val(JTRPInput.Text1.Text) / 1000) *
  Val(JTRPInput.txtYears.Text) * (Val(JTRPInput.txtLength.Text) / 0.62137)) *
  Exp(0.0285 * (Val(JTRPInput.Text4.Text) / (Val(JTRPInput.txtLength.Text) /
0.62137)) - 0.631 * Val(JTRPInput.Text6.Text) + 2.52 *
(Val(JTRPInput.Text5.Text) / Val(JTRPInput.Text4.Text)) - 0.748 *
Val(JTRPInput.Text7.Text) - 0.604 * Val(JTRPInput.Text8.Text)), "fixed")
        Text2.Text = Format(Val(JTRPInput.txtPastNoCrash.Text) *
(Val(frmJTRPresult.Text1.Text) / Val((0.494 * (Val(JTRPInput.txtAADT.Text) /
1000) * Val(JTRPInput.txtYears.Text) * (Val(JTRPInput.txtLength.Text) /
0.62137)) * Exp(0.0285 * (Val(JTRPInput.txtAccess.Text) /
(Val(JTRPInput.txtLength.Text) / 0.62137)) - 0.631 *
Val(JTRPInput.txtShoulder.Text) + 2.52 * (Val(JTRPInput.txtSignals.Text) /
Val(JTRPInput.txtAccess.Text)) - 0.748 * Val(JTRPInput.txtTWLTL.Text) - 0.604
*Val(JTRPInput.txtMedian.Text)))), "fixed")

End Sub
```

- Model 1 Chart Screen

The Chart module presents the graphic shown in Figure 6, - the sensitivity of crashes to change in access according to the model employed.

```
Option Explicit
Option Base 1

Private Sub Command1_Click()
  frmJTRPresult.Show
  frmJTRPresult.SetFocus
  Unload frmJTRPcht
End Sub
Private Sub Form_Load()
  ChtJTRP.Column = 1
  ChtJTRP.TitleText = "Crash Vs. Access Density"
```

This set values of each x-axis' variables within the chart option.

```
With ChtJTRP
      Dim arrData(21, 1)
      arrData(1, 1) = Val(JTRPInput.txtPastNoCrash.Text) * (Val((0.494 *
(Val(JTRPInput.Text1.Text) / 1000) * Val(JTRPInput.txtYears.Text) *
(Val(JTRPInput.txtLength.Text) / 0.62137)) * Exp(0.0285 *
((Val(JTRPInput.Text4.Text) * 0.5) / (Val(JTRPInput.txtLength.Text) / 0.62137)) -
0.631 * Val(JTRPInput.Text6.Text) + 2.52 * (Val(JTRPInput.Text5.Text) /
```

```
 (Val(JTRPInput.Text4.Text))) - 0.748 * Val(JTRPInput.Text7.Text) - 0.604 *
 Val(JTRPInput.Text8.Text))) / Val((0.494 * (Val(JTRPInput.txtAADT.Text) /
1000) * Val(JTRPInput.txtYears.Text) * (Val(JTRPInput.txtLength.Text) /
0.62137)) * Exp(0.0285 * (Val(JTRPInput.txtAccess.Text) /
(Val(JTRPInput.txtLength.Text) / 0.62137)) - 0.631 *
Val(JTRPInput.txtShoulder.Text) + 2.52 * (Val(JTRPInput.txtSignals.Text) /
Val(JTRPInput.txtAccess.Text)) - 0.748 * Val(JTRPInput.txtTWLTL.Text) - 0.604
* Val(JTRPInput.txtMedian.Text))))
        arrData(2, 1) = Val(JTRPInput.txtPastNoCrash.Text) * (Val((0.494 *
(Val(JTRPInput.Text1.Text) / 1000) * Val(JTRPInput.txtYears.Text) *
(Val(JTRPInput.txtLength.Text) / 0.62137)) * Exp(0.0285 *
((Val(JTRPInput.Text4.Text) * 0.55) / (Val(JTRPInput.txtLength.Text) /
0.62137)) - 0.631 * Val(JTRPInput.Text6.Text) + 2.52 *
(Val(JTRPInput.Text5.Text) / (Val(JTRPInput.Text4.Text))) - 0.748 *
Val(JTRPInput.Text7.Text) - 0.604 * Val(JTRPInput.Text8.Text))) / Val((0.494 *
(Val(JTRPInput.txtAADT.Text) / 1000) * Val(JTRPInput.txtYears.Text) *
(Val(JTRPInput.txtLength.Text) / 0.62137)) * Exp(0.0285 *
(Val(JTRPInput.txtAccess.Text) / (Val(JTRPInput.txtLength.Text) / 0.62137)) -
0.631 * Val(JTRPInput.txtShoulder.Text) + 2.52 *
(Val(JTRPInput.txtSignals.Text) / Val(JTRPInput.txtAccess.Text)) - 0.748 *
Val(JTRPInput.txtTWLTL.Text) - 0.604 * Val(JTRPInput.txtMedian.Text))))

        .
        .( Between arrData(3,1) to arrData(20,1) are Omitted)
        .
        arrData(21, 1) = Val(JTRPInput.txtPastNoCrash.Text) * (Val((0.494 *
(Val(JTRPInput.Text1.Text) / 1000) * Val(JTRPInput.txtYears.Text) *
(Val(JTRPInput.txtLength.Text) / 0.62137)) * Exp(0.0285 *
((Val(JTRPInput.Text4.Text) * 1.5) / (Val(JTRPInput.txtLength.Text) / 0.62137)) -
0.631 * Val(JTRPInput.Text6.Text) + 2.52 * (Val(JTRPInput.Text5.Text) /
(Val(JTRPInput.Text4.Text))) - 0.748 * Val(JTRPInput.Text7.Text) - 0.604 *
Val(JTRPInput.Text8.Text))) / Val((0.494 * (Val(JTRPInput.txtAADT.Text) /
1000) * Val(JTRPInput.txtYears.Text) * (Val(JTRPInput.txtLength.Text) /
0.62137)) * Exp(0.0285 * (Val(JTRPInput.txtAccess.Text) /
(Val(JTRPInput.txtLength.Text) / 0.62137)) - 0.631 *
Val(JTRPInput.txtShoulder.Text) + 2.52 * (Val(JTRPInput.txtSignals.Text) /
Val(JTRPInput.txtAccess.Text)) - 0.748 * Val(JTRPInput.txtTWLTL.Text) - 0.604
* Val(JTRPInput.txtMedian.Text))))

    ChtJTRP.ChartData = arrData
 End With
```

This Sets each row's labels.

```
    With ChtJTRP
    .Row = 1
    ChtJTRP.RowLabel = Val(JTRPInput.Text4.Text) * 0.5
    .Row = 2
```

```
ChtJTRP.RowLabel = Val(JTRPInput.Text4.Text) * 0.55
.Row = 3
ChtJTRP.RowLabel = Val(JTRPInput.Text4.Text) * 0.6
    .
    .
    .
.Row = 21
ChtJTRP.RowLabel = Val(JTRPInput.Text4.Text) * 1.5
End With
End Sub
```

- MDI Screen

The MDI Screen is a multiple display interface screen where all the forms are gathered and performed under this mother form.

```
Option Explicit

Private Sub AboutJTRP_Click()
  frmAboutJTRP.Show
End Sub

Private Sub AboutMNDot_Click()
  frmAboutMNDotModel.Show
End Sub

Private Sub mnuWhite_Click()
  frmWhiteInput.Show
  frmWhiteInput.SetFocus
End Sub
```

This sub program assigns the size and position of the MDI form.

```
Private Sub MDIForm_Load()
  Height = 8000
  Width = 10000
  Top = (Screen.Height - Height) / 2
  Left = (Screen.Width - Width) / 2
End Sub

Private Sub mnuCascade_Click()
    MDIForm1.Arrange vbCascade
End Sub

Private Sub mnuClose_Click()
  Unload Me
End Sub
```

```
Private Sub mnuDotmodel_Click()
    frmMNDot.Show
        frmMNDot.SetFocus
End Sub

Private Sub mnuExit_Click()
   Close #1
   Close #3
   Close #5
   Close #7
  End
End Sub
   Private Sub mnuJTRP_Click()
   Unload frmJTRPresult
   JTRPInput.Show
   JTRPInput.SetFocus
 End Sub

Private Sub mnuTile_Click()
     MDIForm1.Arrange vbTileHorizontal
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
   Select Case Button.Tag
      Case "JTRP"
         mnuJTRP_Click
      Case "MNDot"
         mnuDotmodel_Click
      Case "White"
         mnuWhite_Click
      Case "help"
         frmAboutJTRP.Show
         frmAboutJTRP.Left = 0
         frmAboutMNDotModel.Show
         frmAboutMNDotModel.Left = Screen.Width - frmAboutJTRP.Width
   End Select
End Sub
```

- Public Class Module (High Level)

  Variables' properties are declared here as Public so that they are accessible from any part of the supporting sub programs.

```
Option Explicit
Public Type Rec
AADT As Single
```

```
        Length As Single
        Year As Single
        Access As Single
        Signal As Single
        Shoulder As Single
        TWLTL As Single
        Mediam As Single
        PastNoCrash As Single
        ADDT1 As Single
        Access1 As Single
        Singal1 As Single
        Shoulder1 As Single
        TWLTL1 As Single
        Mediam1 As Single
        TotalPoint As Single
        TotalPoint1 As Single
        Length1 As Single
        Length2 As Single
        MAADT As Single
        MAADT1 As Single
        Number As Single
        boolDeleteRecord As Boolean
 End Type
Public Record As Rec
    .
    .
    .

Public Record3 As Rec3
Public Sub Main()
        MDIForm1.Show
        Intro.Show
End Sub
```

Appendix B

Limitation of AccuPack, Four enhancements are worth considering.

1. *Include a combo box option in the chart.* The chart option shows users the pattern of the calculated crashes versus signalized access density now. If the user could select different variables in lieu of signalized access density, such as AADT or speed limits, then users could more readily to understand how each roadway condition affects corridor safety.
2. *Automate data entry for other models.* AccuPack contains three different mathematical models to give users more choices to expand AccuPack's usability. If a user could enter data just once and then have these data passed to other models automatically then data entry for users would be reduced.
3. *Tabulate the variation between the modeled crash rate and the actual crash rate,* such that after a few years one could know which model had come the closest prediction to actual crash rate.
4. *Develop software that could archive, from data entry alone, enough data to develop a Department of Transportation specific model that could be compared to the available models for the study.* In other words, one could develop a software that archives results for multiple corridors, and then the software eventually a DOT – specific database replete with variables such as unsignalized driveways, signals, median type, roadway type, number of lanes, average daily traffic (ADT), speed, and crash rates. Additionally, while some VTRC staff have evaluated the software, it has not been widely distributed among state DOTs. Thus, if these four improvements are made in the future, feedback from these personnel could also be obtained.