AIMMS

A One-Hour Tutorial for Students

April 2000

Paragon Decision Technology

Johannes Bisschop
Koos Heerink

Paragon Decision Technology B.V.
P.O. Box 3277
2001 DG Haarlem
The Netherlands
Tel.: +31(0)23-5511512
Fax: +31(0)23-5511517
Email: info@paragon.nl
WWW: http://www.aimms.com

ISBN xx-xxxxxx-x-x

This documentation was typeset by Paragon Decision Technology B.V. using LATEX and the LUCIDA font family.

Contents

Chapter 1

Introduction

Ways to learn AIMMS ...

There are several ways in which you can learn the AIMMS language and get a basic understanding of its underlying development environment. The following opportunities are immediately available, and are part of the AIMMS installation.

- There is a *live demo* application in which you can observe the basic functioning of AIMMS through a combination of static text and a moving cursor to point at simulated actions.
- There are two *tutorials* on AIMMS to provide you with some initial working knowledge of the system and its language. One tutorial is intended for students, while the other is aimed at professional users of AIMMS.
- There is a *model library* with a variety of examples to illustrate simple and advanced applications together with particular aspects of both the language and the graphical user interface.
- There are three *reference books* on AIMMS, which are available in PDF format and in hard copy form. They are *The User's Guide* to introduce you to AIMMS and its development environment, *The Language Reference* to describe the modeling language in detail, and *Optimization Modeling* to enable you to become familiar with building models.

... *for students*

As a student of optimization modeling, you may not have much time for learning yet another tool in order to finish some course work or homework requirements. In this case, look at the 'live demo' for five to ten minutes, and then concentrate your efforts on this tutorial. After completing this tutorial, you should be able to use the system to build your own simple models, and to enter your own small data sets for subsequent viewing. The book on *Optimization Modeling* may teach you some useful tricks, and will show you different (mostly non-trivial) examples of optimization models.

... *for professionals*

As a professional in the field of optimization modeling you are looking for a tool that simplifies your work and minimizes the time needed for model construction and model maintenance. In this situation, you cannot get around the fact that you will need to initially make a substantial time investment to get to know several of the advanced features that will subsequently support you in your role as a professional application builder. Depending on your skills, experience, and learning habits you should determine your own indi-

vidual learning path. A suggested route is to look at the 'live demo' first, and then work through the extensive tutorial especially designed for professionals. This tutorial for professionals provides a good start, and should create excitement about the possibilities of AIMMS. Individual examples in the library, plus selected portions of the three books, will subsequently offer you additional ideas on how to use AIMMS effectively while building your own advanced applications.

Tutorials are different in scope

The one-hour tutorial for students is designed as the bare minimum needed to build simple models using the AIMMS **Model Explorer**. Data values are entered by hand using data pages, and the student can build a page with objects to view and modify the data. The extensive tutorial for professionals is an elaborate tour of AIMMS covering a range of advanced language features plus an introduction to all the building tools. Especially of interest will be the modeling of time using the concepts of horizon and calendar, the use of quantities and units, the link to a database, the connection to an external DLL, and advanced reporting facilities. Even then, some topics such as efficiency considerations (execution efficiency, matrix manipulation routines) and the AIMMS API will remain untouched.

Chapter 2

What to Expect

This chapter

In this chapter you will find a brief overview of the tasks to be performed, a compact statement of the underlying model to be built, and a glimpse of the output you will produce.

2.1 Scope of one-hour tutorial

Summarizing your work

Once you have read the short problem description and the associated mathematical model statement, you will be asked to complete a series of tasks that make up this one-hour tutorial, namely:

- create a new project in AIMMS,
- enter all identifier declarations,
- enter the data manually,
- save your data in a case,
- build a small procedure,
- build a single page with
  - header text,
  - a standard table and two bar charts with input data,
  - a composite table and a stacked bar chart with output data,
  - a button to execute the procedure, and
  - a scalar object with the optimal value,
- perform a what-if run.

2.2 Problem description and model statement

Problem description

Truckloads of beer are to be shipped from two plants to five customers during a particular period of time. Both the available supply at each plant and the required demand by each customer (measured in terms of truckloads) are known. The cost associated with moving one truck load from a plant to a customer is also provided. The objective is to make a least-cost plan for moving the beer such that the demand is met and shipments do not exceed the available supply from each brewery.

The following table provides the data for the problem described in the previous paragraph. *Data overview*

Customers / Plants	Unit Transport Cost: Amsterdam	Breda	Gouda	Amersfoort	Den Bosch	Supply
Haarlem	131	405	188	396	485	47
Eindhoven	554	351	479	366	155	63
Demand	28	16	22	31	12	

Table 2.1: Input data for beer transport problem

The following declarations list the identifiers that are part of the mathematical program to be built. *Identifier declarations*

Indices:

p	*plants*
c	*customers*

Parameters:

S_p	*supply at plant p*
D_c	*demand by customer c*
U_{pc}	*unit transport cost from p to c*

Variables:

x_{pc}	*transport from p to c*
z	*total transport cost*

The mathematical model summary below captures the least-cost plan to transport beer such that the demand is met and shipments do not exceed available supply. *Model summary*

Minimize:

$$z = \sum_{pc} U_{pc} x_{pc}$$

Subject to:

$$\sum_{c} x_{pc} \leq S_p \qquad \forall p$$

$$\sum_{p} x_{pc} \geq D_c \qquad \forall c$$

$$x_{pc} \geq 0 \qquad \forall (p, c)$$

Figure 2.1: The Netherlands

Using explicit names

Even though the above notation with one-letter symbols is typical of small mathematical optimization models, it will not be used when entering the model into AIMMS. Instead, explicit names will be used throughout to avoid any unnecessary translation symbols. The number of symbols needed to describe practical applications is generally large, and a clear naming convention supports the understanding and maintenance of large models.

2.3 A preview of your output

A single page

Figure 2.2 is a page that contains both input and output data associated with the beer transport model. In Chapter 6 you will be asked to construct this page using the point-and-click facilities available in AIMMS.

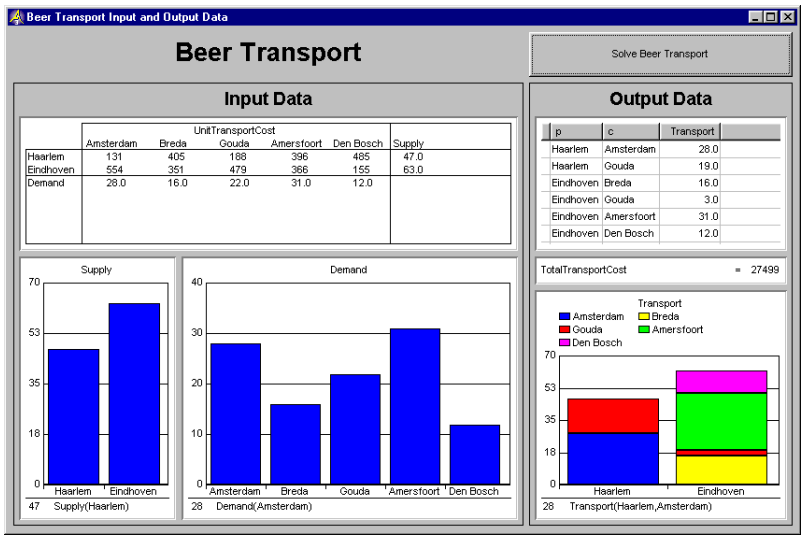

Figure 2.2: An input-output page

Chapter 3

Building the Model

3.1 Starting a new project

Creating a folder

You are advised to use the Windows Explorer to create a dedicated folder in which to store your AIMMS projects. Figure 3.1 serves as an illustration.

Figure 3.1: A selection of folders

Starting AIMMS

Assuming that AIMMS 3 has already been installed on your machine, execute the following sequence of actions to start AIMMS:

- press the **Start** button on the taskbar,
- go to the **Programs** submenu,
- go to the **Aimms 3.x** submenu, and
- select and click on the AIMMS icon to start AIMMS.

Next, you will see the AIMMS splash screen. Once AIMMS has started, the splash screen will disappear and the AIMMS window will open. Should you encounter the AIMMS **Tip of the Day** dialog box, close it, because it is not relevant to you at this point.

Creating a new project

Press the **New Project** button, which is located in the left most position on the AIMMS toolbar. The dialog box shown in Figure 3.2 will then appear, requiring you to take the following actions:

- specify 'Beer Transport' as the project name,
- press the wizard button to select the dedicated folder for your AIMMS projects, and
- press the **OK** button.

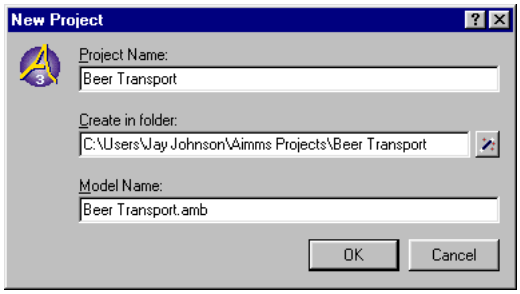

Figure 3.2: The **New Project** wizard

The AIMMS project window (see Figure 3.3) for the 'Beer Transport' project will then appear, and you are ready to enter your model.

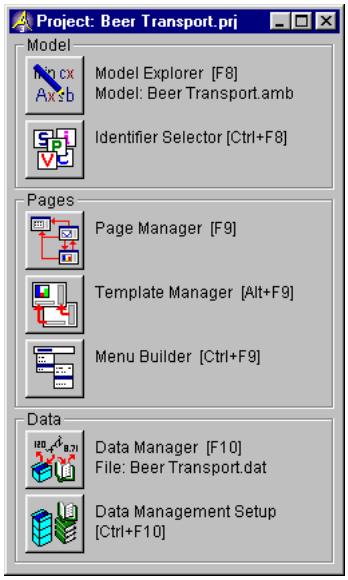

Figure 3.3: The AIMMS project window wizard

3.2 The Model Explorer

Opening the Model Explorer

Press the **Model Explorer** button on the project window to open the AIMMS **Model Explorer** which will be displaying the initial model tree shown in Figure 3.4. In this initial model tree you will see

- a single *declaration section*, where you can store the declarations used in your model,

- the predefined procedure *MainInitialization*, which is not relevant for this tutorial,
- the predefined procedure *MainExecution*, where you will put the execution statement necessary to solve the mathematical program, and
- the predefined procedure *MainTermination*, which is again not relevant for this tutorial.

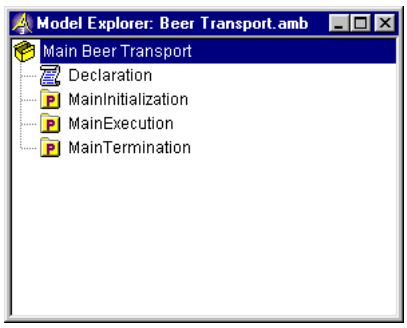

Figure 3.4: The initial model tree

3.3 Entering sets and indices

Opening the declaration section

The declaration of model identifiers requires you to first 'open' the declaration section by double-clicking on the scroll icon . Note that double-clicking on the name of the declaration section instead of on its icon will open the attribute form of the declaration section and will therefore, at this point, not lead to the desired result. After opening the declaration section the standard identifier buttons S P V C … on the toolbar will be enabled.

Creating the set 'Plants'

To create a set of plants you should take the following actions:

- press the **Set** button S to create a new set identifier in the model tree,
- specify 'Plants' as the name of the set, and
- press the *Enter* key to register the name.

Opening its attribute form

Next, you need to declare the index p as an attribute of the set 'Plants'. You can open the attribute form by double-clicking on the node 'Plants' in the model tree. The resulting initial attribute form of the set 'Plants' is shown in Figure 3.5.

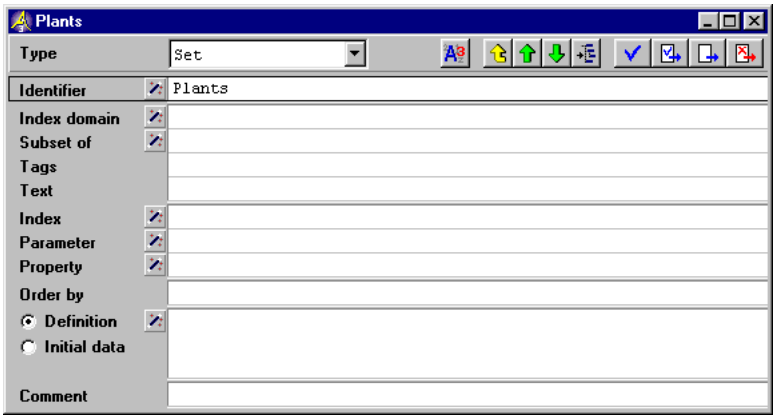

Figure 3.5: The initial attribute form of the set 'Plants'

Declaring the index p

To declare the index p as an attribute of the set 'Plants', execute the following sequence of actions:

- move the mouse cursor to the 'Index' attribute field, and click in the (empty) edit field,
- enter the letter p, and
- complete the attribute form by pressing the **Check, Commit and Close** button .

Creating the set 'Customers'

Next, create the set 'Customers' with associated index c in exactly the same way as you created the set 'Plants' with index domain p. Figure 3.6 contains the resulting model tree.

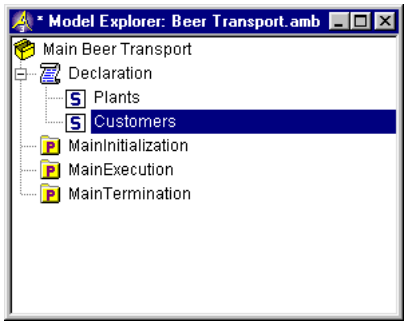

Figure 3.6: An intermediate model tree

Saving your changes

The asterisk on the left of the title bar indicates that additions to your project have not yet been saved to disk. To save your work, please press the **Save Project** button on the toolbar.

3.4 Entering parameters and variables

Domain specification

In this section you will declare the parameters and variables that are needed in your model. The sets 'Plants' and 'Customers' and their associated indices will be used to specify the index domain for the parameters and variables.

Creating the parameter 'Supply'

The declaration of a parameter is similar to the declaration of a set. To enter the parameter 'Supply(p)', you should execute the following actions:

- press the parameter button on the toolbar to create a new parameter in the model tree,
- specify 'Supply(p)' as the name of the parameter, and
- press the *Enter* key to register the name.

Note that parentheses are used to add the index domain p to the identifier 'Supply'.

Creating the parameter 'Demand'

The parameter 'Demand(c)' can be added in the same way. Should you make a mistake in entering the information, then you can always re-edit a name field by a single mouse click within the field.

Creating the parameter 'UnitTransport-Cost'

The last model parameter 'UnitTransportCost' is a two-dimensional parameter with index domain (p, c). After entering 'UnitTransportCost(p,c)', the resulting model tree should be the same as in Figure 3.7.

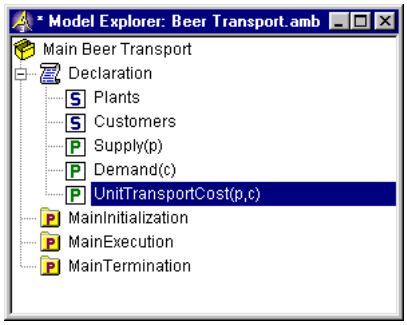

Figure 3.7: An intermediate model tree

Creating the variable 'Transport'

Declaring a variable is similar to declaring a parameter.

- press the variable button on the toolbar to create a new variable in the model tree,
- specify 'Transport(p,c)' as the name of the variable, and
- press the *Enter* key to register the variable.

Specifying range attribute

After opening the attribute form of the variable by double-clicking on the node 'Transport' in the model tree, press the wizard button in front of the 'Range' attribute field. The resulting dialog box provides the opportunity to specify the range of values that the variable 'Transport' is allowed to take. In this case, select the 'Standard Range', then select 'nonnegative', and finally press the *OK* button (see Figure 3.8).

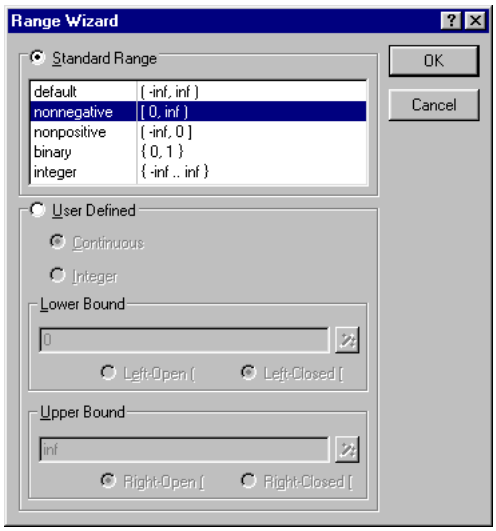

Figure 3.8: The AIMMS range wizard

Creating the variable 'TotalTransportCost'

It should be clear by now how to create the variable 'TotalTransportCost'. This variable will be used to specify the objective function. After entering its name, open the attribute form. There is no need to specify the range attribute, since the default range will suffice. You are now ready to enter the following definition of this particular variable:

```
sum[ (p,c), UnitTransportCost(p,c) * Transport(p,c) ]
```

Specifying definition attribute

Simply enter the above definition in the 'Definition' attribute field. You could type the entire sentence yourself, but you can also let AIMMS do some of the typing for you. Considering the parameter 'UnitTransportCost(p,c)', the following two support features are quite useful.

- Type the letter *u* or *U*, and press the *Ctrl-Spacebar* combination for automatic name completion. By pressing the key combination once more, AIMMS will also add the attached indices (p, c).
- Another option available to you is to drag the name 'UnitTransportCost(p,c)' from the model tree to the edit field of the 'Definition' attribute.

The attribute form should now have the same content as shown in Figure 3.9. By pressing the **Check, Commit and Close** button, you can verify whether AIMMS will accept the definition you entered.

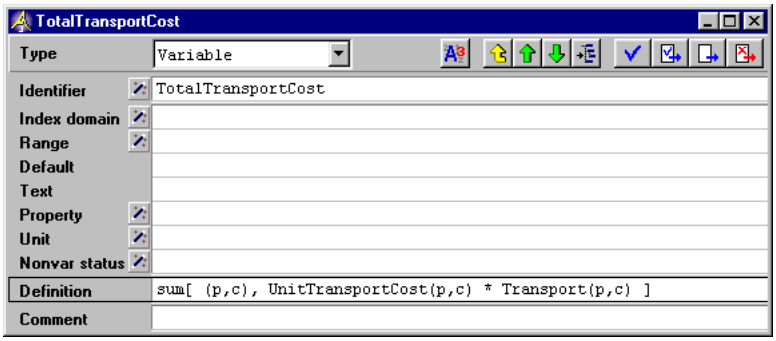

Figure 3.9: The completed attribute form for the variable 'Transport'

3.5 Entering constraints and the mathematical program

The supply and demand constraints

Creating the supply and demand constraints, each with their own definition, requires the same actions as creating a variable with a definition (as you just completed). The only difference is that you must use the button instead of the button. The following two forms should be the result of your efforts.

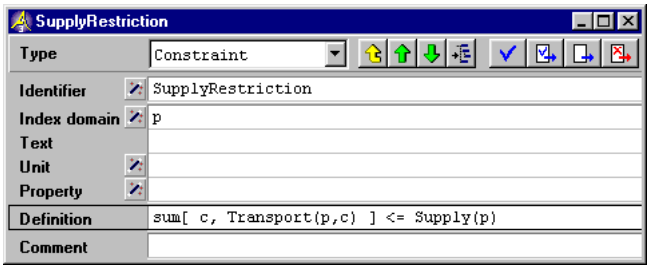

Figure 3.10: The completed attribute form for the constraint 'Supply'

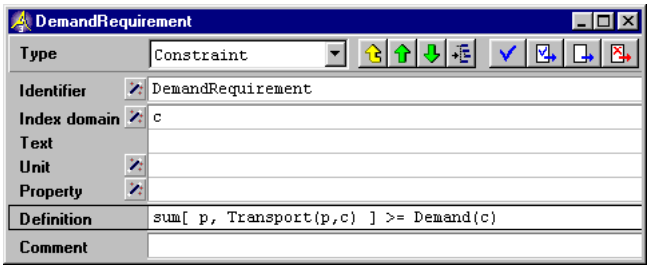

Figure 3.11: The completed attribute form for the constraint 'Demand'

Creating the mathematical program

A mathematical program, unlike sets, parameters, variables and constraints, does not have a special button on the toolbar. By using the identifier button , you obtain access to all the other types of AIMMS identifiers. After pressing this button, select the 'Mathematical Program' entry alongside the icon, press the *OK* button, and enter 'LeastCostTransportPlan' as the name of the mathematical program.

Specifying its attributes

You should then complete the attribute form of the mathematical program as illustrated in Figure 3.12. As an exercise you should use the wizards to complete the three attributes. By default, all variables and constraints will be considered as part of your mathematical program (thus there is no need to fill in these attributes). Only the **Objective** attribute wizard is discussed in more detail below since the other two wizards are straightforward.

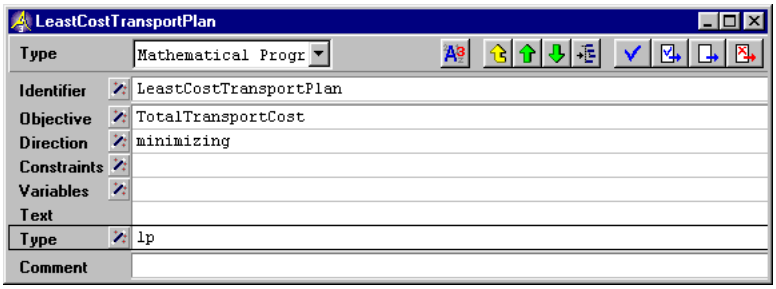

Figure 3.12: The completed attribute form of the mathematical program

Selecting the objective

The **Objective** attribute wizard requires you to select a scalar variable. In the identifier selection wizard (see Figure 3.13), simply select the scalar variable 'TotalTransportCost', and press the *Finish* button.

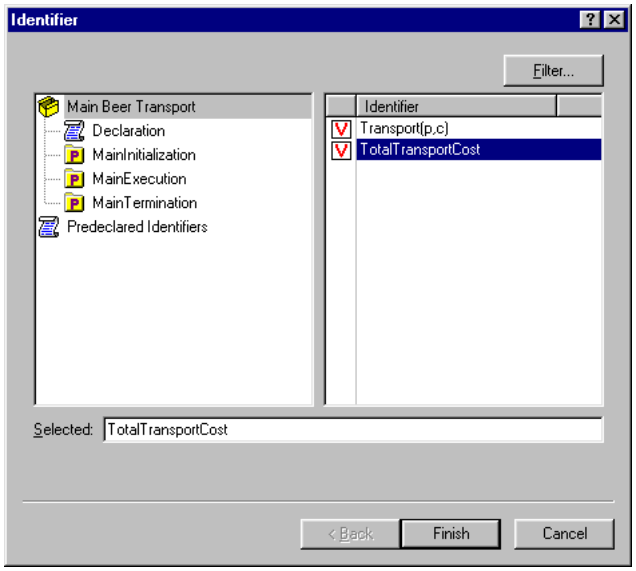

Figure 3.13: The identifier selection wizard

3.6 Viewing the identifiers

Checking your model

You have now entered and declared all model identifiers. The resulting model tree is shown in Figure 3.14. By pressing the *F5* key you can instantly check the validity of your model. You will only receive a message in the event of an error. Once the validity of your model has been verified, you should save your work by pressing the **Save Project** button .

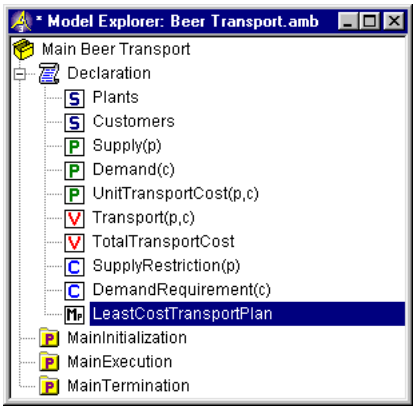

Figure 3.14: The final model tree

Identifier overviews

Even though the Model Explorer is a convenient medium with which to build and inspect your model, you may have the need to view several identifiers at the same time. In this tutorial you will encounter one such example of a predefined view, namely all identifiers with a definition (see Figure 3.15). AIMMS allows you to make your own views as you desire.

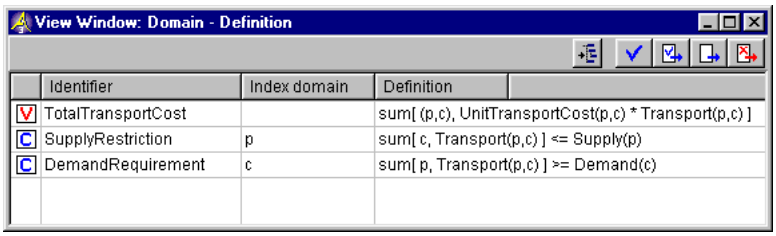

Figure 3.15: View window with identifier definitions

Creating a view

You can create a view window by executing the following steps:

- press the **Identifier Selector** button in the project window,
- select the 'Identifiers with Definition' node, and
- use the right mouse and select the **Open With**... command from the popup menu (see Figure 3.16).

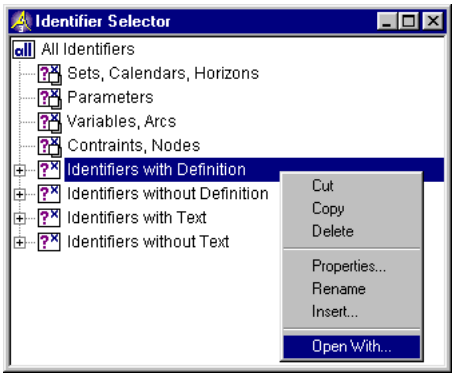

Figure 3.16: Identifier Selector window

For the selected identifiers the view can be constructed as follows:

- select the 'Domain - Definition' entry from the **View Manager** window (see Figure 3.17, and
- press the *Open* button to obtain the overall view.

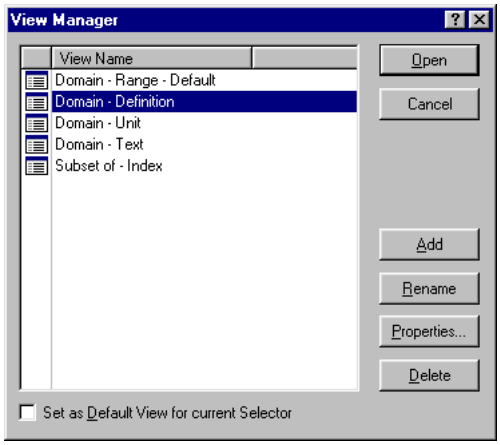

Figure 3.17: View Manager window

Chapter 4

Entering and Saving the Data

4.1 Entering set data

Manual data entry

In this tutorial there are only a few numbers, and you are asked to enter these numbers from the keyboard. In the second tutorial, data is imported from a database. In this section you will encounter a standard data entry facility. Each identifier has an associated data page that you can use both to view data and to enter data.

Elements of the set 'Plants'

To enter the two elements of the set 'Plants', you should execute the following actions:

- open the attribute form of the set 'Plants',
- press the **Data** button  ,
- move the mouse pointer to the data page as shown in Figure 4.1, and click in the empty edit field at the top of the data page,
- enter 'Haarlem' as the first element of the set,
- press the *Enter* key to register this element,
- enter 'Eindhoven' as the second element of the set,
- press the *Enter* key to register this element, and
- close the data page by clicking *Close* button (the data changes are immediately committed).

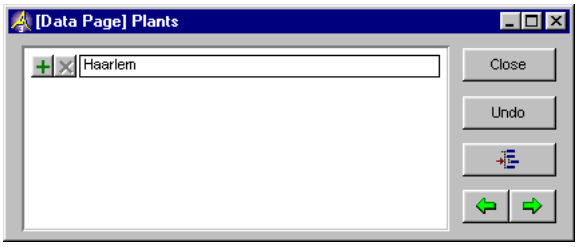

Figure 4.1: Data page for the set 'Plants'

Modifying an element

If necessary you can modify an element. Select an element, and it will appear in the edit field at the top of the data page. You can then enter the modified element name.

Elements of the set 'Customers'

The elements of the set 'Customers' are entered in exactly the same way as for the set 'Plants'. The five elements are listed in Figure 4.2. Note that the last element 'Den Bosch' contains a blank character.

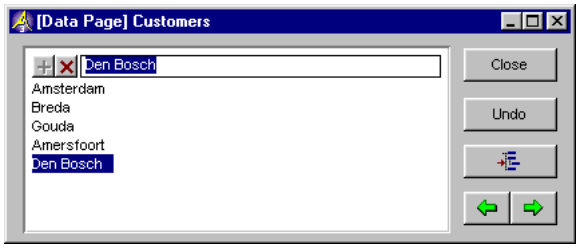

Figure 4.2: Data page for the set 'Customers'

4.2 Entering parameter data

Empty tables

The data page of each indexed parameter is automatically filled with the elements of the corresponding sets. All that is left for you to do, is to enter the nonzero data values.

Supply data

In order to enter the data for the parameter 'Supply', you should execute the following actions (which are similar to the ones described in the previous section):

- open the attribute form of the parameter 'Supply',
- press the **Data** button ,
- move the mouse pointer to the first data field and click,
- enter the number 47,
- press the *Enter* key to register the first value,
- enter the number 63,
- press the *Enter* key to register the second value, and
- close the data page by pressing *Close* button.

Figure 4.3 shows the completed data page of the parameter 'Supply'.

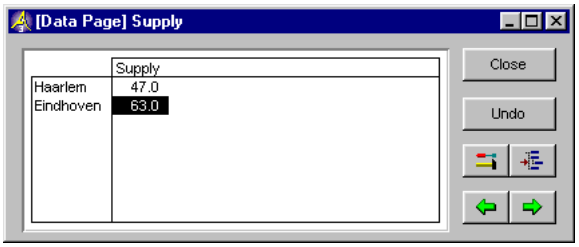

Figure 4.3: Data page for the parameter 'Supply'

Demand data

The data values for the parameter 'Demand' are entered in exactly the same way as for the parameter 'Supply'. The five data values are listed in Figure 4.4.

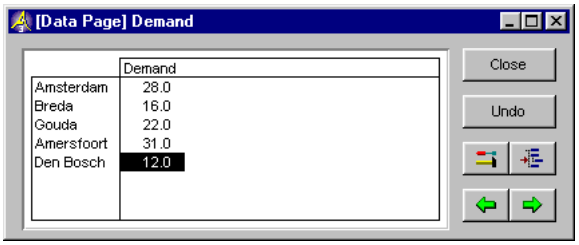

Figure 4.4: Data page for the parameter 'Demand'

Cost data

The parameter 'UnitTransportCost' is two-dimensional, and requires you to complete a table. The completed data page for this parameter is shown in Figure 4.5.

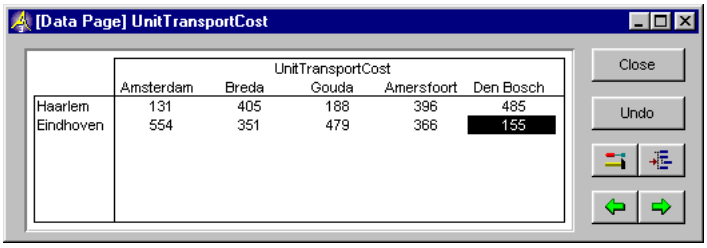

Figure 4.5: Data page for the parameter 'UnitTransportCost'

4.3 Saving your data

Case management

AIMMS has the option to store the data values of all identifiers in what is referred to as a 'case'. There are facilities both to save cases and to load cases.

Saving a case

In order to save the data that you just entered in a new case named 'Initial Beer Transport Data', you need to execute the following steps:

- go to the **Data** menu and execute the **Save Case** command,
- in the **Save Case** dialog box (see Figure 4.6) enter the name 'Initial Beer Transport Data' in the 'Name' field (without the quotes), and
- press the **Save** button to save your data.

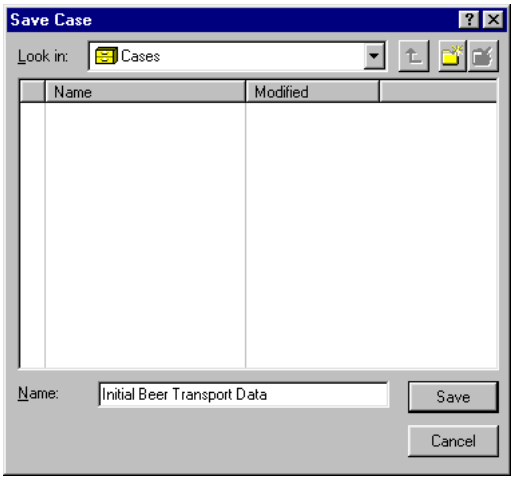

Figure 4.6: Save Case dialog box

Loading a case as the startup case

If a project in AIMMS is closed and subsequently reopened, you may want to reload your data. You may even want AIMMS to load a specific case automatically each time your project is started. This can be accomplished (without programming) using the AIMMS **Options** dialog box illustrated in Figure 4.7.

- go to the **Settings** menu and execute the **Project Options** command,
- select the **Project** - **Startup & Authorization** folder in the option tree,
- click on the Option **Startup Case** in the right-most window,
- press the wizard button,
  - select the case 'Initial Beer Transport Data',
  - press the *OK* button on the **Select Case** dialog box,
- press the *Apply* button on the AIMMS **Options** dialog box, and
- finish by pressing the *OK* button.

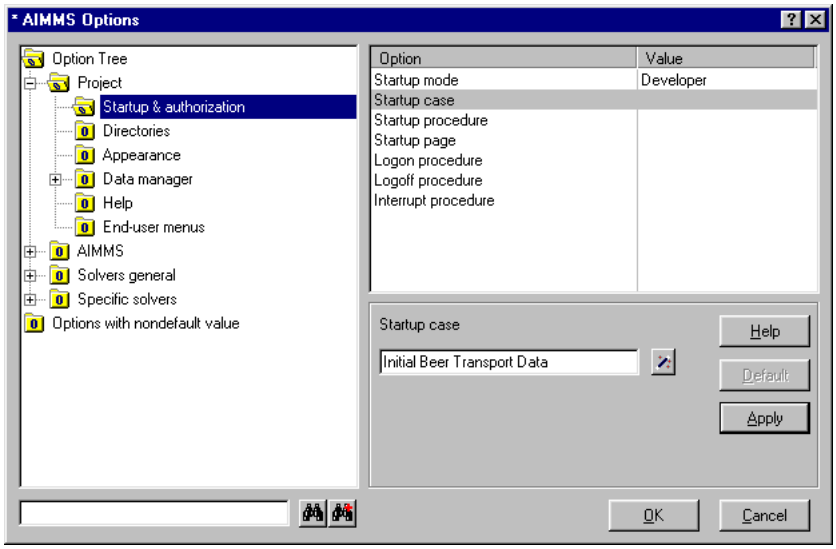

Figure 4.7: AIMMS options dialog box

Saving your project

It is a good habit to save your work regularly. The option settings above are also saved when you save the entire project. You can save the project by pressing the **Save Project** button . Note that saving a project does not mean that the data is also saved. Saving data requires you to save a case.

Loading a case manually

At any time during an AIMMS session you can load a case manually as follows:

- go to the **Data** menu, select the **Load Case** submenu and execute the **as Active**... command,
- select the desired case name in the **Load Case** dialog box (see Figure 4.8), and
- press the *Load* button.

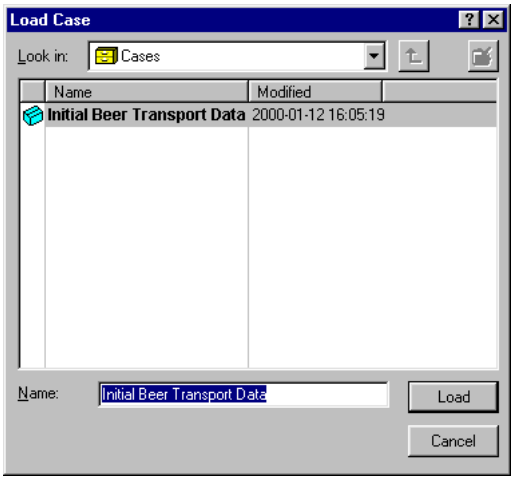

Figure 4.8: Load case dialog box

Chapter 5

Solving the Model

5.1 Computing the solution

Procedures for action

Thus far, you have entered all the identifiers, their attributes and their data. You will also need to build at least one procedure in order to be able to instruct AIMMS to take action. In this tutorial, you will enter two statements inside the body of the existing (empty) procedure *MainExecution*: one to solve the mathematical program, and the other to set the solution to zero when the mathematical program is not optimal.

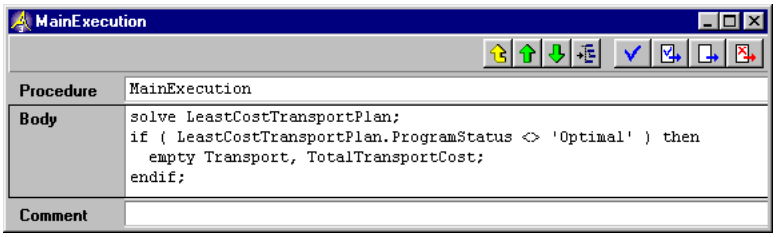

Figure 5.1: The attribute form of MainExecution

Building a procedure

The procedure *MainExecution* can be completed as follows:

- press the *F8* key to open the **Model Explorer**,
- select the *MainExecution* procedure and open it by double-clicking ,
- enter the two statements in the body attribute as illustrated in Figure 5.1, and
- press the **Check, Commit and Close** button to register the changes.

Should AIMMS report errors, simply check your input and make the necessary corrections.

Right-mouse for help

To obtain information about specific AIMMS keywords, you can use the right-mouse popup menu to open the AIMMS documentation on the appropriate page with a single click. For instance, you can obtain help on the 'ProgramStatus' keyword as follows:

- position the cursor over the 'ProgramStatus' keyword,
- right-click the mouse and select the 'ProgramStatus' entry in the 'Help' submenu (see Figure 5.2).

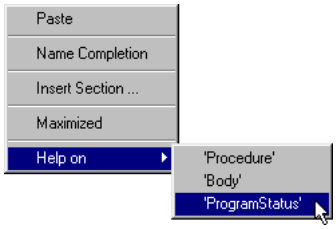

Figure 5.2: A right-mouse popup menu

Running the procedure

The procedure *MainExecution* is special in that there is a dedicated key, *F6*, to execute this procedure. For all other procedures you can use the right mouse button to select the **Run Procedure** command.

Watching execution progress

By pressing the *Ctrl* and *p* keys simultaneously, AIMMS displays a progress window with selected information on the progress it has made (or is making) during an execution phase. Figure 5.3 shows the progress window you should expect to see.

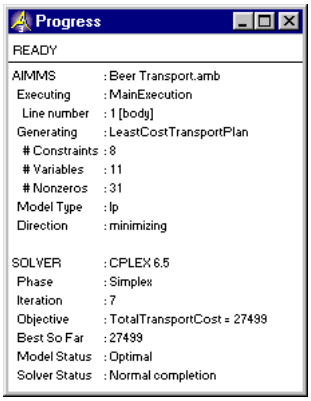

Figure 5.3: The AIMMS progress window

Results in data pages

You have already encountered data pages while entering the elements of sets and the numeric values of parameters. Once AIMMS has computed the values of the variable 'Transport', these values become immediately available on the corresponding data page. Just go to this variable in the model tree, and click on it. Then use the right mouse to select the Data... command to open the data page (see Figure 5.4).

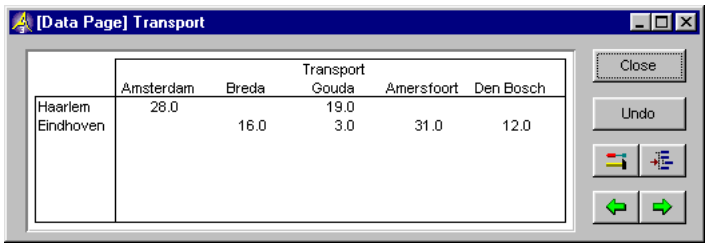

Figure 5.4: Data page displaying the solution for the variable 'Transport'

Chapter 6

Building a Page

Building custom pages

Even though AIMMS provides standard pages for each identifier, such pages are not set up to look at groups of related identifiers. That is why model builders and end-users of an application usually prefer to interact with an application through one or more custom pages.

6.1 Creating a new page

Using the Page Manager

To create a new empty page you should execute the following steps:

- press the **Page Manager** button in the project window,
- press the button on the toolbar to create a new page,
- specify 'Beer Transport Input and Output Data' as the name of this new page, and
- press the *Enter* key to register the page.

The **Page Manager** with the new page is shown in Figure 6.1.

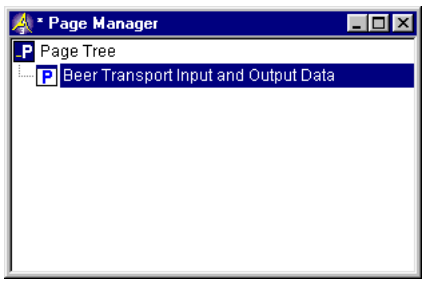

Figure 6.1: A Page Manager with a single page

6.2 Presenting the input data

Be aware of two page modes

A page is either in *Edit* mode or in *User* mode. The *Edit* mode is used for creating and modifying the objects on a page. The *User* mode is for viewing and editing the data displayed within objects on a page.

Opening the page

To open the new page in *Edit* mode:

- select the new page in the **Page Manager**, and
- press the  button on the toolbar to open the selected page in *Edit* mode.

Drawing a new table . . .

To create a new table, perform the following actions:

- press the new-table button  on the toolbar,
- position the mouse cursor at where the upper left corner of the new table should be,
- depress the left mouse button and drag the mouse cursor to where the lower right corner of the new table should be, and
- release the mouse button.

. . . and selecting an identifier

You can now complete the identifier selection dialog box as follows:

- select the parameter 'UnitTransportCost(p,c)' in the identifier selection wizard as illustrated in Figure 6.2,
- press the *Next* button,
- press the *Finish* button, and if necessary
- adjust the position and size of the table object such that all information is neatly displayed.

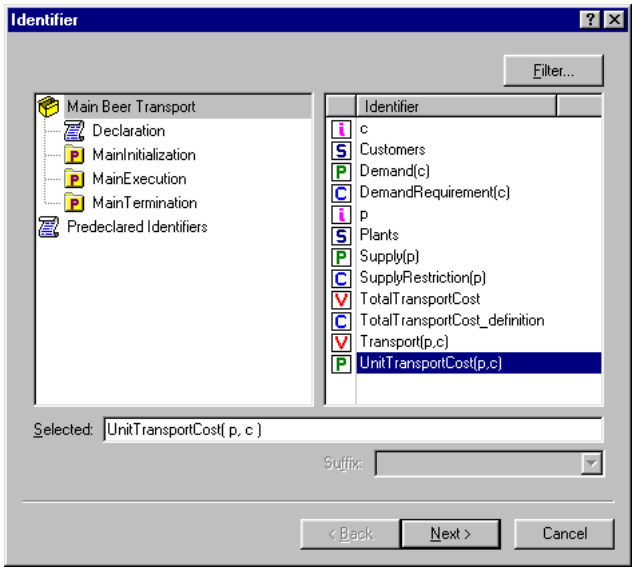

Figure 6.2: Identifier selection wizard

Adding supply data to existing table

To add another identifier to the 'UnitTransportCost' table, execute the following actions in *Edit* mode:

- select the table by clicking on it,
- press the  button on the toolbar (or alternatively, use the right mouse) to access the properties dialog box,
- select the contents tab (see Figure 6.3),
- press the *Add* button,
- select the identifier 'Supply(p)', press the *Next* button, and then press the *Finish* button, and
- back on the contents tab, press the *OK* button.

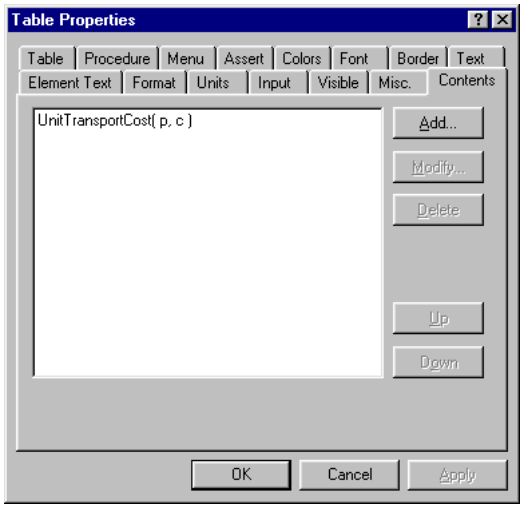

Figure 6.3: Table contents tab

Adding demand data to the table

You can add demand data to the table in the same way as you added the supply data. The resulting table is shown in Figure 6.4.

	UnitTransportCost					
	Amsterdam	Breda	Gouda	Amersfoort	Den Bosch	Supply
Haarlem	131	405	188	396	485	47.0
Eindhoven	554	351	479	366	155	63.0
Demand	28.0	16.0	22.0	31.0	12.0	

Figure 6.4: Table displaying input data

Creating two bar charts

Creating a bar chart is essentially the same process as creating a table. The following steps summarize the process for the parameter 'Supply':

- press the new-bar-chart button on the toolbar,
- position the mouse cursor, and drag to form the new bar chart,
- select the parameter 'Supply(p)' in the identifier selection wizard,
- press the *Next* button, and then the *Finish* button.

You can then create a bar chart for the demand data in the same way as you created the bar chart for the supply data. Your intermediate page should now look like the one in Figure 6.5.

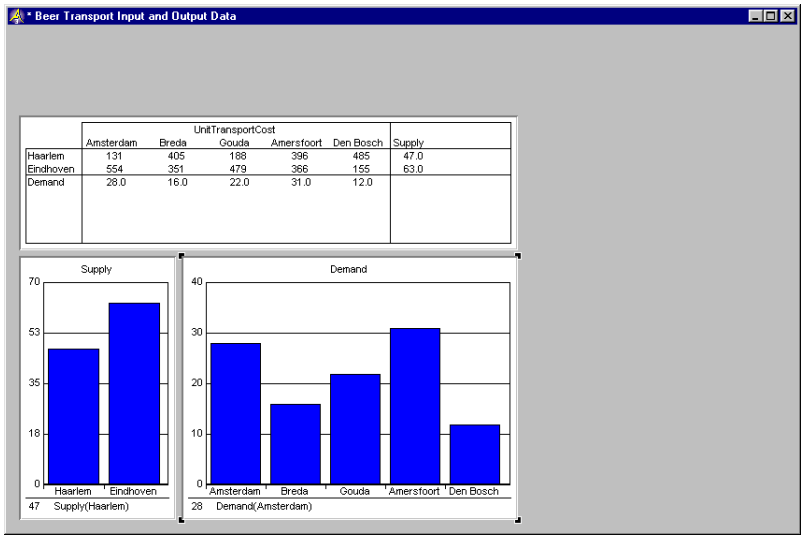

Figure 6.5: Intermediate input-output page

6.3 Presenting the output data

Creating a composite table

A composite table in AIMMS is like a relational database table: the first columns contain indices, and the remaining columns contain identifiers defined over these indices. Creating a composite table containing only the optimal solution is similar to creating a standard table or a bar chart, and requires the following actions:

- press the button on the toolbar to create a new composite table,
- draw the table using the mouse,
- select the variable 'Transport(p,c)' in the identifier selection wizard to indicate which index values must be displayed,
- press the *Next* button, and then the *Finish* button.

Once the index domain has been specified, you can use the standard 'add identifier' facility to complete the table:

- press the button on the toolbar to access the properties dialog box,
- select the contents tab in the properties dialog box,
- press the *Add* button, and add the identifier 'Transport(p,c)' to complete the composite table.

Creating a stacked bar chart

Yet another way to display the solution is by means of a stacked bar chart:

- create a standard normal bar chart displaying the variable 'Transport(p,c)'.
- select the 'bar chart' tab in the properties dialog box as illustrated in Figure 6.6),
- instead of the default 'Overlapping' option, select the 'Stacked' option, and
- press the *OK* button.

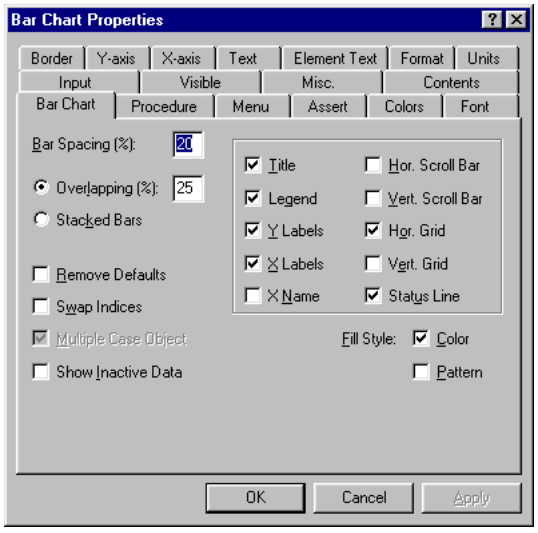

Figure 6.6: Bar chart property dialog box

Creating a scalar object

The scalar object is designed to display scalar values. To display the optimal solution value in a scalar object you should do the following:

- press the button on the toolbar to create a scalar object,
- draw the scalar object using the mouse,
- select the scalar variable 'TotalTransportCost' in the identifier selection wizard, and
- press the *Finish* button.

6.4 Finishing the page

Building a well-organized overview

Designing a professional looking graphical end-user interface is not a trivial activity, and is beyond the scope of this tutorial. Nevertheless, you will be asked to spend a little time building a nice looking page as illustrated in Figure 6.11 at the end of this section.

Creating a button

One item on this page is a button designed to trigger the solution of the 'LeastCostTransportPlan' mathematical program. To create such a button, you need to execute the following actions:

- press the button on the toolbar to create a new button, and draw the button using the mouse,
- enter the quoted string "Solve Beer Transport" as the title of the button, and
- select the actions tab.

The action to be specified is that AIMMS executes (i.e. "runs") a procedure. In this example, the procedure is 'MainExecution'. Continue with the following steps:

- select 'Run' as the action to add,
- press the *Add* button,
- select option 'Procedure',
- press the enabled wizard button ,
- select the procedure 'MainExecution',
- press the *Finish* button, and accept by pressing the *OK* button.

The completed **Actions** tab of the **Button Properties** dialog box is displayed in Figure 6.7. Note that the button can only be used to solve the model when the page is put into *User* mode by pressing the **User Mode** button .

Creating a text object

The resulting input-output page (see Figure 6.11) contains three text objects. The title text 'Beer Transport' can be created as follows:

- select the **Text** command from the **Object** menu (see Figure 6.9), and draw a rectangle using the mouse,
- specify 'Beer Transport' as the static text on the **text** tab of the **Text Properties** dialog box (see Figure 6.8) ,
- select the **Font** tab of the **Text Properties** dialog box, and
- press the *Add* button.

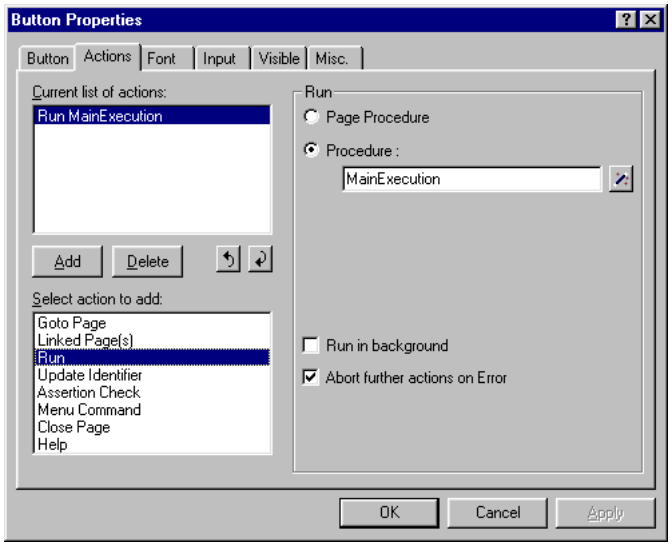

Figure 6.7: The action tab of the button properties dialog box

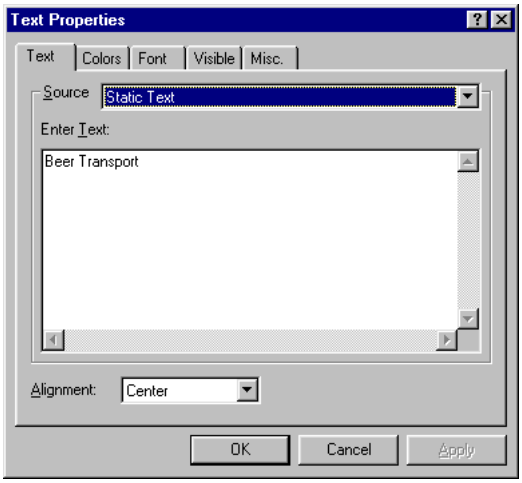

Figure 6.8: The text tab of the text properties dialog box

You can now specify and name the appropriate font, and thereby complete the text object.

- select 'Bold' as the *Font Style*, and '20' as the 'Font Size',
- press the *OK* button,
- specify 'Title' as the name of the new font,
- press the *OK* button to return to the **Text Properties** tab,
- again, press the *OK* button to leave the **Text properties** dialog box,

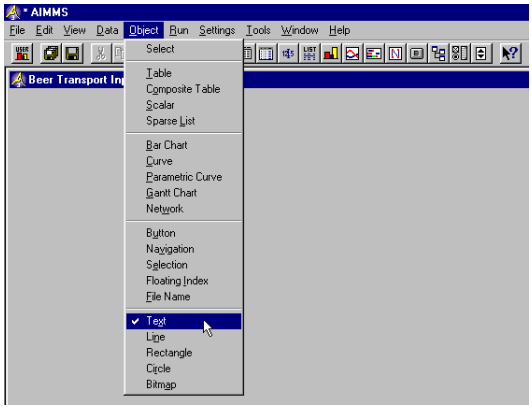

Figure 6.9: A selected area of a page

The other two text objects displaying the text 'Input Data' and 'Output Data' are created in the same way. Instead of using the newly constructed 'Title' font, you should create a second custom font, named 'Header' font, of size '14'. The font tab of the **Text Properties** dialog box is displayed in Figure 6.10.

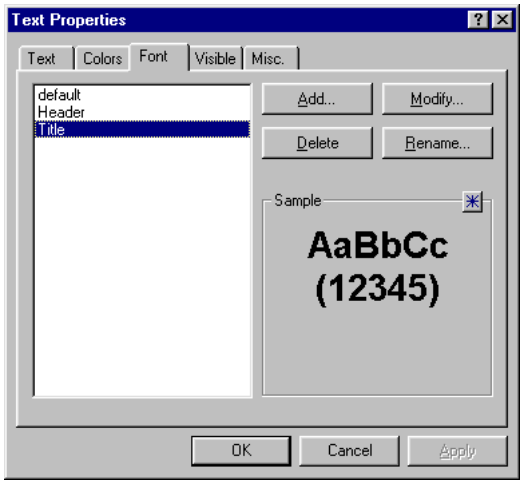

Figure 6.10: The font tab of the text properties dialog box

Creating two rectangles

The page is completed by adding two rectangles to emphasize that there are two groups of objects representing input data and output data. Assuming that you have rearranged and resized the objects to fit neatly together, you can draw the rectangles as follows:

- select the **Rectangle** command from the **Object** menu, and
- draw the rectangle using the mouse.

Your page should now look like the one in Figure 6.11.

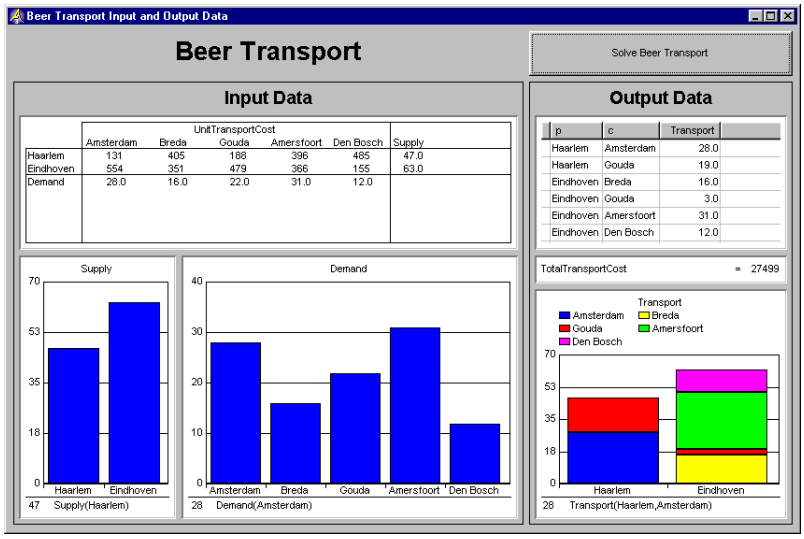

Figure 6.11: An input-output page

Chapter 7

Performing a What-If Run

7.1 Modifying input data

Page user mode

Having developed the input-output page, you are now ready to *use* the page. For this purpose you must put the page into *User* mode by pressing the **User Mode** button  .

What-if analysis

The input-output page allows you to see the effect of changes in either the demand, the supply, or the cost figures of the transport model. Just change any input data, re-solve the model, and view the resulting output.

Dragging a bar chart

For example, to change the available supply in 'Haarlem' you can perform the following actions:

- in the 'Supply' bar chart, select the bar representing the supply in 'Haarlem',
- position the mouse pointer at the top of the bar, and simply
- drag the mouse upwards to increase the supply from 47 to 57 (see Figure 7.1).

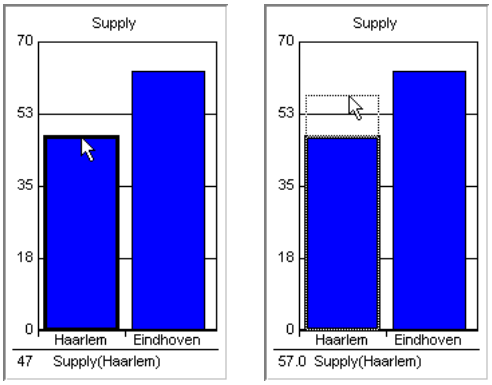

Figure 7.1: The dragging process for supply data illustrated

Alternatively, you can click on the corresponding bar, and enter the new supply value of 57 in the edit field on the lower left part of the bar chart.

Re-solving the mathematical program

You are now ready to re-solve the model. To do so, simply press the **Solve Beer Transport** button at the top of your page. You will see an improvement (i.e. decrease) in optimal cost from 27499 to 26626.

Improvement explained

Note that a cost decrease could have been expected, because the entire capacity of 'Haarlem' had been used initially. By increasing the supply at Haarlem, 'Gouda' no longer needs Eindhoven as a second supplier (see Figure 7.2).

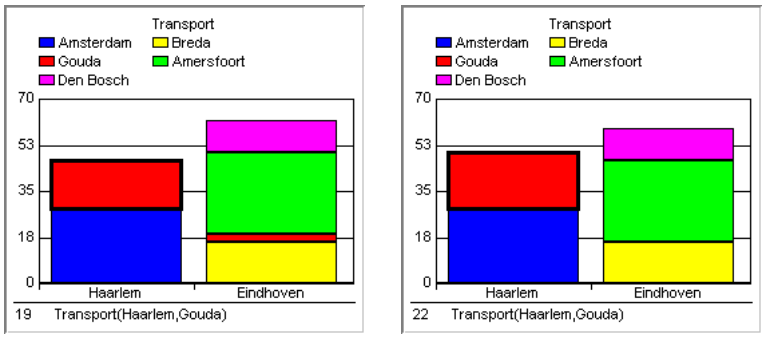

Figure 7.2: The effect of changes in the supply data