

第 1 章 简介

学习 AIMMS 语言、了解其基础开发环境的方法有很多种。以下方法可供使用。

学习 AIMMS 的方法……

- AIMMS 的两套教程(*tutorials*)，可提供有关 AIMMS 系统及其语言的基本使用知识。一套专为初学者设置，另一套则面向专业人士。
- *模型库(model library)*提供多个示例，阐述简单及高级应用程序，以及 AIMMS 语言及图形用户界面的特定层面。
- AIMMS 的三本参考书目(*reference books*)支持 PDF 格式和打印稿格式。其中，《使用说明》介绍了 AIMMS 及其开发环境，《语言参考》详细说明了如何使用建模语言，而《优化建模》则可使您熟悉建模方法。
- 《函数参考》详细地描述了 AIMMS 中的所有可用函数，包括其参数及返回类型；还详细介绍了 AIMMS 中可用的预先声明的标识符。
- *在线帮助(Online Help)*介绍了使用 AIMMS 的很多细节。您可以通过情境帮助(Context Help)功能获得在线帮助，了解和使用 AIMMS 的大部分工具、属性形式和目标。
- AIMMS 网站上有相关的 *专题研讨会(workshops)*，通过一系列实践课程，让您了解到一项完整的决策支持应用的所有开发过程。有关研讨会的更多信息，请参考 www.aimms.com。

作为优化建模语言的初学者，您可能无暇掌握其他工具来完成一些项目或作业要求。鉴于此，请集中精力阅读本教程。学完本教程后，您将可以使用 AIMMS 系统自行建立简单模型，并可输入小数据集以供后续处理。《优化建模》将提供一些实用技巧，向您展示不同的（多为重要的）优化模型。

……对于初学者

作为优化建模领域的专业人士，您可能需要一种可简化工作、缩短模型构建和模型维护时间的工具。在此情况下，不可否认的是，您从一开始就需要投入大量时间来了解多项高级功能，这些功能将在日后使您更加胜任专业应用程序开发人员一职。您应根据您的技能、经验以及学习习惯，确定适合自己的学习途径。按照您的途径，我们建议您仔细阅读专为专业人士所设置的泛读教程。该教程为专业人士提供了良好的学习起点，将激发您对 AIMMS 潜在功能的浓厚兴趣。在构建您的高级应用程序时，代码库中的个例，以及三本书目内的精选章节，将使您了解更多有效使用 AIMMS 的方法。

……对于专业人士

一小时初学者教程旨在提供使用 AIMMS **模型浏览器** (AIMMS **模型浏览器**) 构建简单模型所需的最基本内容。初学者可通过数据页手动输入数值，并可建立对象页面查看及修改数据。专业人士泛读教程则是一本详细的 AIMMS 导览，涵盖了一系列高级语言功能的说明，并介绍了所有模型构建工具。其中尤其能激发兴趣的包括，使用范围和日程表的概念建立模型、数量和单位的使用、链接至数据库，以及连接至外部 DLL（动态链接库）。尽管如此，本教程未涉及到效率考虑（执行效率、矩阵操作规程）以及 AIMMS API（应用程序接口）等话题。

教程范围有所不同

目前面向专业人士的泛读教程要求大量的数据导入。从头开始构建整个应用程序需花费一些时日。但如果时间有限，可以根据教程导入模型及

尚需时日……

界面的一部分。

本教程是从使用 ODBC（开放式数据库连接）以 MS Access 格式存储的数据库中读取数据的。因此，为了完成课程，您需要安装 Microsoft Access。

在本教程中，您将构建自己的终端用，您将创建的页面如图 1.1 所示。

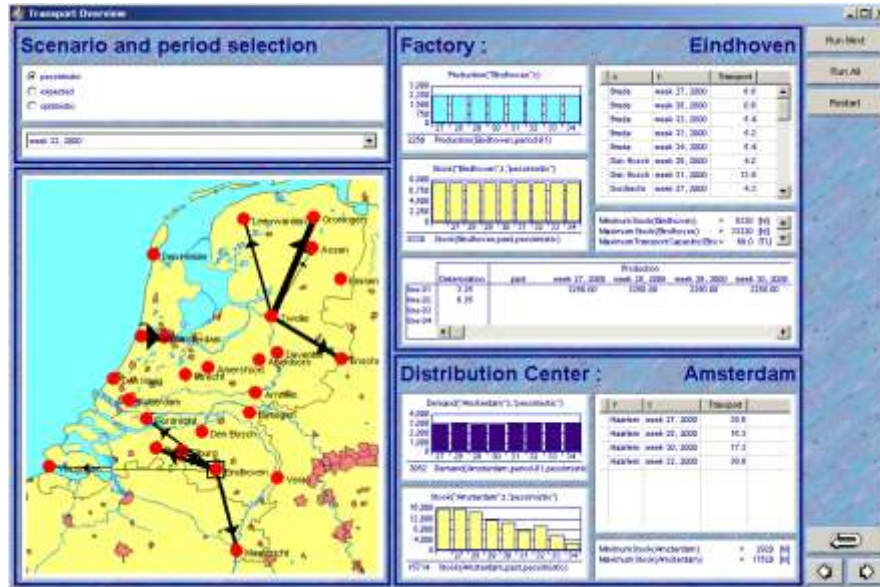


图 1.1: 最优传输数据概况

第 2 章 问题描述

在本章，您将了解如何将实际问题转化成最优模型。该问题陈述涵

本章

2.1 初始问题组件

本教程讨论的应用考量了年度规划范围和周度规划周期。该应用旨在建立一个稳固的生产和维护计划表。

规划范围

以周为基础，衡量某一特定软饮料的生产和分配。该软饮料共有 3 个工厂和 22 个配送中心，全部位于荷兰（如图 2.1 所示）。每周均通过货车将软饮料从工厂运送到各配送中心。每周之内，从特定工厂运出的货物数量有上限。

生产和分配

每个工厂有多条生产线，每条生产线每天产量是固定的（以公升为计量单位）。在任一周内，生产线要么以固定的生产水平运转，要么完全不运转。

生产线

生产线的模态切换 (mode switch) 一词意指生产的开始/关闭变动。因此，当生产线在前一周不运转，而在某一特定周处于运转时，将出现模态切换；反之亦然。

模态切换



图 2.1: 荷兰

工厂和配送中心都设有贮存设施。库存同产量一样，以公升为计量单位。各地都有储备贮存，且贮存量是有限的。

总成本以美元计算，由生产、分配、贮存和模态切换所产生的成本构成。前三个部分很显然，但对于模态切换需加以阐述。在该应用中，生产线雇佣的一部分工人是临时工，但人们认为频繁的雇佣和裁员是不可取的，因此模态切换将造成额外的人工费。

2.2 维护和假期计划

生产线需根据其相应的损耗率进行定期维护。假设一条生产线连续运转 16 周，不久之后则必须关闭一周，并由先前在该生产线工作的员工对其进行维护；如果一条生产线超过 64 周不运转，则在运行这条生产线的前一周必须进行维护；如果一条生产线几周内处于运转和停用交替状态，则每周停用机器时的恶化率是运转时的四倍。

生产线上的工人也执行生产线的维护工作。因此，当出于维护引起生产停产或再次运转时，前文所述的模态切换惩罚不适用。

为确保每个工厂的生产持续性，要求在同一时间内，每个工厂仅可

贮存

成本构成

维护要求……

……不造成模态切换

……以及保持连

有一条生产线处于维护状态。

续性

工厂的生产线在周末和国定节假日将关闭。此外，在这期间，不会从工厂运送软饮料到配送中心，因此一个生产周通常最多只有五个工作日。

非运转期间

除了国定节假日，还有一部分时间供工人进行旅游。出于规划目的，假定并非每位工人都享受休假，且在工人休假期间，所有处于运转的生产线的生产水平将以特定的百分比降低。当产生下滑或随后的增长时，模态切换惩罚不适用。

休假

2.3 需求的多种情况

每周由分配中心提供给消费者的软饮料的需求是不确定的。需求变动亦即储备贮存的原因，已对这一变动进行多年观察。然而，当构造以需求作为参数的模型时，必须选择每周的变动量，即构成需求情形 (demand scenario)。

需求的不确定性

为了制定更加稳固的生产和维护计划，我们设置了三种需求情形来代替单一需求情形。这三种需求情形分别是：期待需求、乐观需求及悲观需求，从而代替了前几年的所有需求情形。

三种情形

稳健计划的核心是构建同时适用于以上三种情形的生产和维护计划。而针对每一种情形唯一允许的决策不同在于分配量和库存量。有关情景最优化的更多内容，请查阅 AIMMS 的《优化模型》第 16 及 17 章。

稳固的计划

2.4 规划目标

公司的总体目标旨在获取年度总规划中以周为基础的生产和维护计划。制定的计划应以线条图表示（如图 2.2 所示），该图具体到三家工厂中每家工厂的每一个生产线的生产和维护计划。该计划使产能使用、库存建立、每家工厂所需雇佣的临时工可视化。

总体目标

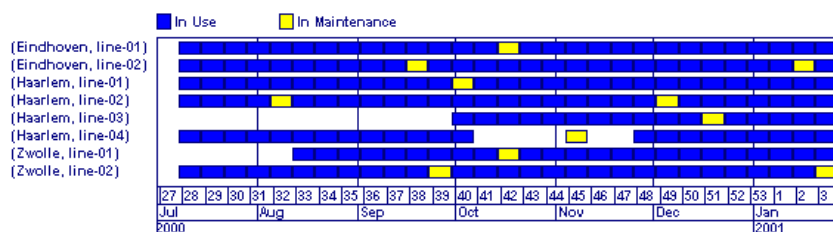


图 2.2: 线条图的选定部分

数学规划模型的具体目的是使规划周期的总成本最小化。这就直接指定了需要详细列出与生产和模态切换相关的个别成本构成。但是，与贮存和分配相关的成本构成成为独立情形，因此赋予不同的权重。此模型中，乐观情形和悲观情形的发生概率均分别为 0.25。

具体目标

2.5 周期滚动法

在实际应用中，本章所示的工厂和分配中心的数量远远大于此处所述的数量。此外，产品的数量不止一种。基于要考虑年度中每周的情况，本教程中所建立的数学规划可能因数据过大而在实际应用中无法一次性解决。

规格问题

针对这种情形，可以考虑一个较短的时间周期作为补救。这直接影响到决定变量的数量，因为这些决定变量都是以周数作为指标。这种方法的缺点也很明显，它无法满足全年的管理要求计划。

限制周期

在实际应用中，这种方法表现为按顺序进行一个个数学规划，每个数学规划的时间段为八周。一旦第一周的规划问题得以解决，与第一周相关的决策则视为既定。随后第二周的数学规划将照此进行，所有与第二周相关的生产和维护决策则固定下来。这个过程一直持续到年度规划中的最后一个八周。

滚动周期

周期滚动模型在运算效率和精确度之间得以平衡。如果规划时间间隔较长，则方案将得到更好的优化。相应的数学规划的规模越大，计算机求解过程所占用的时间就越长。时间间隔的长度应准确反映未来数据对于一期决策的滞后性。这种选择依赖于实际应用。在本教程中，八周的规划间隔已足够。

对未来数据的依赖性

周期滚动法的优势之一在于，维护计划在大部分情况下可以放在整个数学规划的外部。每次做出与第一周相关的决定之后，就可以通过调整每一条生产线上机器恶化率的参数来反映维护计划。一旦某条生产线在下一个八周期间进行维修，在相应的预估维护期间这条生产线的生产水平设为零。具体细节实施将在后文中讨论。

外部维护

从教程角度而言，使用时间及滚动区域法是项有趣的练习。但在实际应用中，应注意到，短期规划不足以考量到未来情况。本例中，由于需求波动并不剧烈，且所有库存缓冲量都在合理的范围之内，所以八周期的计划足够长。

评估

模型描述

在本章中,我们将通过前面的问题描述去寻找数学规划的模型描述.

3.1 产品流程

本章用于表示问题维度的索引如下所示:

索引:

- L 位置 (location)
- F 工厂 (factories) 包含于位置 (location)
- C 配送中心 (distribution centers) 包含于位置 (location)
- P 生产线 (production line)
- T 时间 (time period)
- S 需求情况 (demand scenarios)

生产, 分配, 储存的产品流程中决策变量如下所示:

变量:

- q_{ft} 工厂总产量 (Hl 公升 (hectoliter))
- u_{fpt} 生产线是否运行的二进制表示
- x_{fct} 运输量
- y_{lts} 库存量

对于所有需求情况来说, 生产变量都是相同的, 但分配和储存变量可变化的。注意这里的公升和货车容量都是软饮料的测量单位。本教程中, 货车容量等于 12 立方米。

在本章中, 与产品流程相关参数如下所示:

参数:

- D_{cts} 需求
- L_d 实际时间
- Q_{fp} 生产线全部运行的产量
- M_{fpt} 生产线 是否维修的二进制表示
- V_{ft} 是否在休假期间的二进制表示
- F 休假期间生产线的生产水平下降率
- A_{fpt} 潜在产能
- X_j 空闲货车量
- Y_l 最大库存量
- Y_l 最小库存量

与生产线产能, 需求和假期相关的参数可以从外部数据库读取。维修参数由周期滚动法的求解过程中决定。

生产线的潜在产能 A_{fpt} 依赖于维修和休假相关参数, 定义如下:

$$A_{fpt} = L_t(1 - M_{fpt})(1 - F \cdot V_{ft})Q_{fp}, \quad \forall (f, p, t)$$

因为参数 M, F, V 都是非零值, 所以潜在产能 A_{fpt} 小于全运行产能 Q

库存与上一期库存，生产，配送，和需求相关，其等式约束如下所示：

$$y_{lts} = y_{l,t-1,s} + q_{lt} + \sum_f x_{fllts} - \sum_c x_{lcts} - D_{lts}, \quad \forall(l, t, s)$$

$$y_{lts} \in [\underline{Y}_l, \bar{Y}_l], \quad \forall(l, t, s)$$

注意上面的等式约束适用所有位置(包括工厂和配送中心)，并且在某些情况下，等式约束中的某些项是不存在的，譬如对于配送中心库存的等式约束而言，就不存在生产项，而对于工厂库存的等式约束而言，则没有消费者需求项。在AIMMS软件中，你可以为任一标识符指定全局索引，系统将会自动对标识符进行索引约束。

利用前面定义的潜在产量 A_{fpt} ，即可得到每个工厂每周的产量公式：

$$q_{ft} = \sum_p A_{fpt} u_{fpt}, \quad \forall(f, t)$$

同样，由于工厂货车数量的限制，即可得到每周从工厂运送出去的货物量的约束模型：

$$\sum_c x_{fcts} \leq X_f, \quad \forall(f, t, s)$$

注意上面的条件约束只是实际问题的一个简化，现实调度应用中，车辆路线，行程距离，和司机的时间窗口都会影响最终的时间调度。但对于当前年度问题简化而言，这些因素因其不够重要从而不被考虑。

3.2 模式切换

在模式开关中，我们增加以下变量：

变量：

v_{fpt} 模式切换的二进制表示

读者开始可能对模式切换难以理解，但是添加一些补充解释可以让我们理解。请看下面的两个不等式：

$$v_{fpt} \geq u_{fpt} - u_{fp,t-1}, \quad \forall(f, p, t)$$

$$v_{fpt} \geq u_{fp,t-1} - u_{fpt}, \quad \forall(f, p, t)$$

当一条生产线从运行到关闭，或反向变化时，变量 v 始终大于或等于1。在下一章节中，我们希望在目标函数的惩罚项部分，模式切换变量能尽可能的小。如果不存在模式切换，则变量值为0。

考虑一个正在运行的生产线。只要生产线处于维修状态，则生产线产量为0。一旦维修结束，生产线重新开始运行。在本文的案例中，产量的改变并不意味模式切换的开启。在前文中潜在产能 A_{fpt} 的定义是符合我们的实际观察的，在维修发生时，维修参数 M_{fpt} 设为1，则此时产能 A_{fpt} 取0。然而目标函数中惩罚项部分使得 u 变量位于第一级水平，从而避免了模式转换。这同样适用生产线处于停产状态。

3.3 目标

数学规划中与目标函数相关的参数如下所示：

参数:

C_f^q	单位生产成本
C_l^y	单位储存成本
C_{fc}^x	单位运输成本
C^v	模式转变的成本
P_s	需求变化的概率

变量:

r_s	需求情况成本
z	总成本

每种需求情况的单位成本由下列部分构成: 生产成本, 该情况下的储存和配送成本, 和反映模式切换的惩罚成本:

$$r_s = \sum_{ft} C_f^q q_{ft} + \sum_{lt} C_l^y y_{lts} + \sum_{fct} C_{fc}^x x_{fcts} + \sum_{fpt} C^v v_{fpt}, \quad \forall s$$

考虑到每种需求情况发生的概率权重, 则目标函数为:

最小化:

$$z = \sum_s P_s r_s$$

3.4模型总结

优化模型的数学描述如下所示:

Minimize:

$$z = \sum_s P_s r_s$$

Subject to:

$$y_{lts} = y_{l,t-1,s} + q_{lt} + \sum_f x_{fllts} - \sum_c x_{lcts} - D_{lts} \quad \forall (l, t, s)$$

$$q_{ft} = \sum_p A_{fpt} u_{fpt} \quad \forall (f, t)$$

$$\sum_c x_{fcts} \leq X_f \quad \forall (f, t, s)$$

$$v_{fpt} \geq u_{fpt} - u_{fp,t-1} \quad \forall (f, p, t)$$

$$v_{fpt} \geq u_{fp,t-1} - u_{fpt} \quad \forall (f, p, t)$$

$$r_s = \sum_{ft} C_f^q q_{ft} + \sum_{lt} C_l^y y_{lts} +$$

$$\sum_{fct} C_{fc}^x x_{fcts} + \sum_{fpt} C^v v_{fpt} \quad \forall s$$

$$u_{fpt} \in \{0, 1\} \quad \forall (f, p, t)$$

$$x_{fcts} \geq 0 \quad \forall (f, c, t, s)$$

$$y_{lts} \in [\underline{Y}_l, \bar{Y}_l] \quad \forall (l, t, s)$$

$$v_{fpt} \geq 0 \quad \forall (f, p, t)$$

第二部分

模型声明

辅助项目文件

在本章中,我们将会了解如何使用命令安装辅助项目文件,这是完成本教程所必须的。此外,本章还将讲述如何导入模型。

4.1 目录结构

建议您首先使用Windows Explorer创建一个专用文件夹以方便存储AIMMS项目,并在该文件夹的子目录中创建一个储存本教程项目的文件夹。如图4.1所示:



图4.1: 创建子文件夹

为方便您学习本教程中AIMMS项目的创建,可建立如下文件夹:

- a text file containing example project data,
 - an MS Access database containing project data,
 - a DLL with a function external to AIMMS
 - several bitmaps for the end-user interface,
 - a number of model sections for possible import,
 - a number of cases and datasets for possible import,
 - all end-user pages for possible import,
 - a user menubar and toolbar also for possible import, and
 - a copy of this tutorial in PDF format.
-
- 示例项目数据的文本文件
 - 项目数据的MS Access 数据库
 - AIMMS外部函数动态数据库
 - 终端界面的视图文件夹
 - 可能导入的模型文件
 - 可能导入的案例和数据集
 - 可能导入的所有终端界面
 - 本教程的PDF格式的副本

与复制整个教程项目类似,按要求复制上面列表中辅助项目文件夹,你也可以从下面网址下载这些文件夹:

<ftp://ftp.aimms.com/pub/Projects/AIMMSTutorialProjectFiles.exe>

在AIMMS目录中,创建一个教程(Tutorial)目录,该目录包含‘软饮料计划-辅助文件夹’(Soft drink Planning -Auxiliary Files)和‘软饮料计划-完成项目’(Soft drink Planning - Completed Project)。在目录‘教程(Tutorial)\软饮料计划-辅助文件夹(Soft

drink Planning -Auxiliary Files)’下，有8个子目录，在AIMMS目录中复制它们，然后创建一个新的项目目录‘软饮料计划’（soft drink planning）。

最终创建的项目目录结构如下所示：

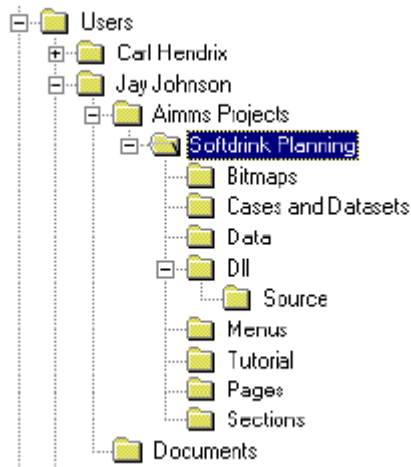


图4.2 教程项目目录结构

4.2 外部项目文件

数据(data)子目录中包含三个文件:文件‘Soft drink Planning.Mdb’,里面包含MS Access数据库储存本教程所需的数据资料;文件‘Soft drink Planning.Dsn’指定了开放数据库互连(ODBC)的文件数据以方便AIMMS连接MS Access数据库;文件‘Locations.dat’包含了第五章所需的案例数据。

教程项目的子目录‘DLL’,包含单一文件‘外部视图’(External Routines.dll)和单一子目录‘资源’(SOURCES),DLL文件中包含一个AIMMS外部函数,但可以在AIMMS中调用此函数。资源(SOURCES)目录中的Microsoft Visual C++ 6.0可以用来创建‘外部视图’(External Routines.dll)。

位图目录中包含用来创建终端用户界面的位图文件。位图文件提高了终端用户界面的可视性,其类型如下所示:

- ‘AIMMS Logo.bmp’
- ‘Background.bmp’
- ‘Button Next.bmp’
- ‘Button Prev.bmp’
- ‘Button Up.bmp’
- ‘Netherlands.bmp’

4.3模型导入

读完本教程的前几章后,你可能想跳过基础章节的学习。下面你可以抛开目录,然后导入一

个或多个模型，在一个更高级的层次继续学习本教程。

子目录中段落(Section)文件夹包含以下可能需要导入的模型文件：

- Absentee Overview.amb'
- 'DLL Link.amb'
- 'Data Management.amb'
- 'Database Link.amb'
- 'Planning Overview.amb'
- 'Production Overview.amb'
- 'Production and Maintenance Model.amb'
- 'Quantities and Units.amb'
- 'Rolling Horizon Procedures.amb'
- 'Scenario Overview.amb'
- 'Softdrink Planning Menubar.amb'
- 'Time.amb'
- 'Transport Overview.amb'

一旦在模型中导入量和单位(Quantities and Unites)段落(等价于6.2章中创建的模型段落)。6.2章中创建的所有标识符都将成为的模型的一部分，注意此时无需执行导入步。本章最后列举了一些关于模型导入的动作参考。

- 在模型树中修改 'Quantities and Units'
- 在 'Edit' 菜单执行 'Import' 命令
- 在模型导入 (Import Model Section) 对话框中打开文件 'Quantities and Units.amb'
- 点击 'Open' 按钮

此时弹出 'Confirm Import' 对话框如图4.3所示,对话框列出了计划导入类型的转换，确认后请点击 'OK' 按钮。

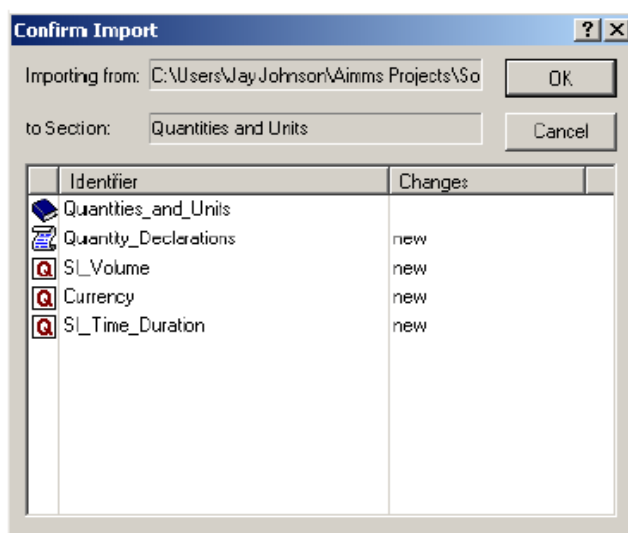


图 4.3 确认导入 (Confirm Import) 对话框

为确保正确执行导入步，可检查模型浏览器 (Model Explorer) 中 'Quantities-units' 段落的目录。

4.4 终端页面导入

建立一个终端界面通常要花费大量时间。本教程仅介绍终端界面的基本内容，而通过绘图对象使页面更人性化和提高界面位置感和尺寸精确度等相关细节不在本教程讨论范围之内。读者可以自己选择依照自己想法绘制界面，或者导入为本教程准备的页面。

‘page’子目录应包含以下页面文件：

- ‘About Softdrink Planning. pag’
- ‘Absentee Overview. pag’
- ‘Background Bitmapping’
- ‘Contents. pag’
- ‘Locations. pag’
- ‘Navigation Buttons. pag’
- ‘Planning Overview. pag’
- ‘Production Overview. pag’
- ‘Scenario Overview. pag’
- ‘Transport Overview. pag’

按照章节 11 要求，创建生产概述（Production Overview）页面。读者可以导入相关的页面文件与自己的页面文件进行比较，或者替换。为实现这一目标，请执行以下操作：

- 按 F9 键打开页面管理器（Page Manager）
- 在 ‘Edit’ 菜单中选择 ‘Import’ 命令
- 在对话框的 ‘Import Pages’ 选择文件 ‘Pages\Production Overview. pag’
- 点击 ‘Open’ 按钮，然后
- 在导入界面对话框中点击 ‘Import’ 按钮（如图 4.4）

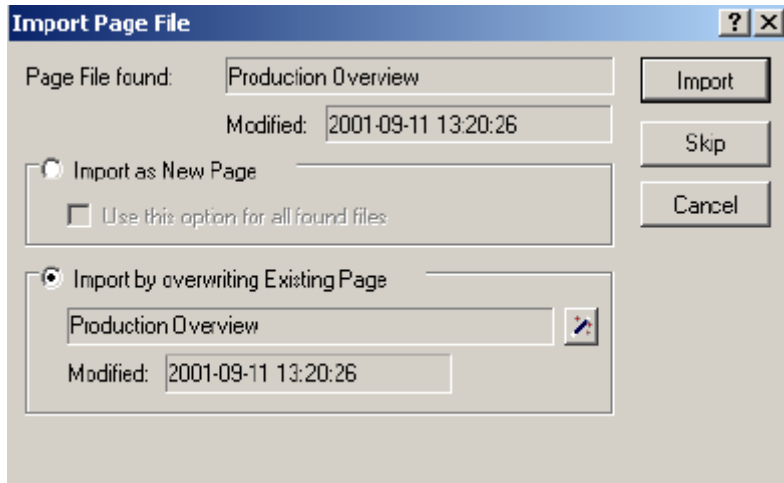


图 4.4: ‘导入界面文件’（Import Page File）对话框

如果想比较你的页面和导入页面，请选择 ‘Import as New Page’ 选项。另一方面，如果想用导入界面代替当前界面，请点击 ‘Import by overwriting Existing Page’ 选项并通过向导进行界面覆盖操作。

导入步完成之后，在界面管理器（Page Manager）核实新的界面，如图 4.5 所示：

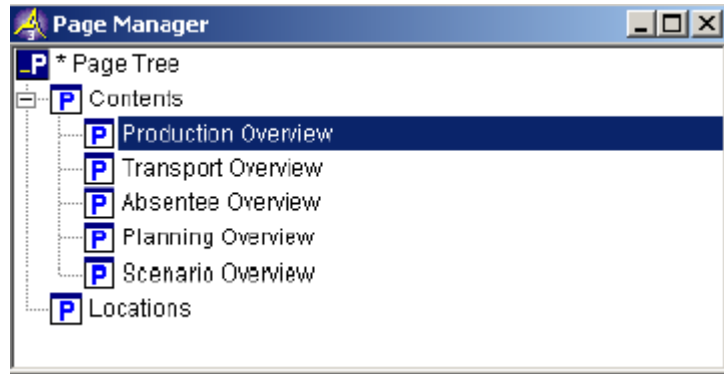


图 4.5: 界面管理器包含导入的终端界面

4.5 案例和数据集导入

完成本教程后，读者可以选择导入数据代替数据录入以图省时省力。周期滚动法过程的相关运行结果也已保存在案例中，可方便读者导入。此外，通过导入相关数据集可以避免节假日周的指定

子目录‘案例和数据集’ (Cases and Database) 包含如下 4 个数据文件：

- ‘Holiday and Vacation Data.dat’
- ‘Initial Data From Database.dat’
- ‘Length 4 to 10.dat’
- ‘Solution After First Roll.dat’

在本节中，从数据文件‘Length 4 to 10.dat’中导入全部案例。数据文件夹包含 7 个已保存的案例，这些案例来自于 14 章结尾的实验。读者可按如下操作导入它们：

- 按 F10 键打开数据管理器 (Data Manager)
- 在 ‘Edit’ 菜单中选择 ‘Import’ 命令
- 从 ‘Import Pages’ 对话框中选择文件 ‘Cases and Datas-ets\Length 4 to 10.dat’
- 点击 ‘Open’ 键

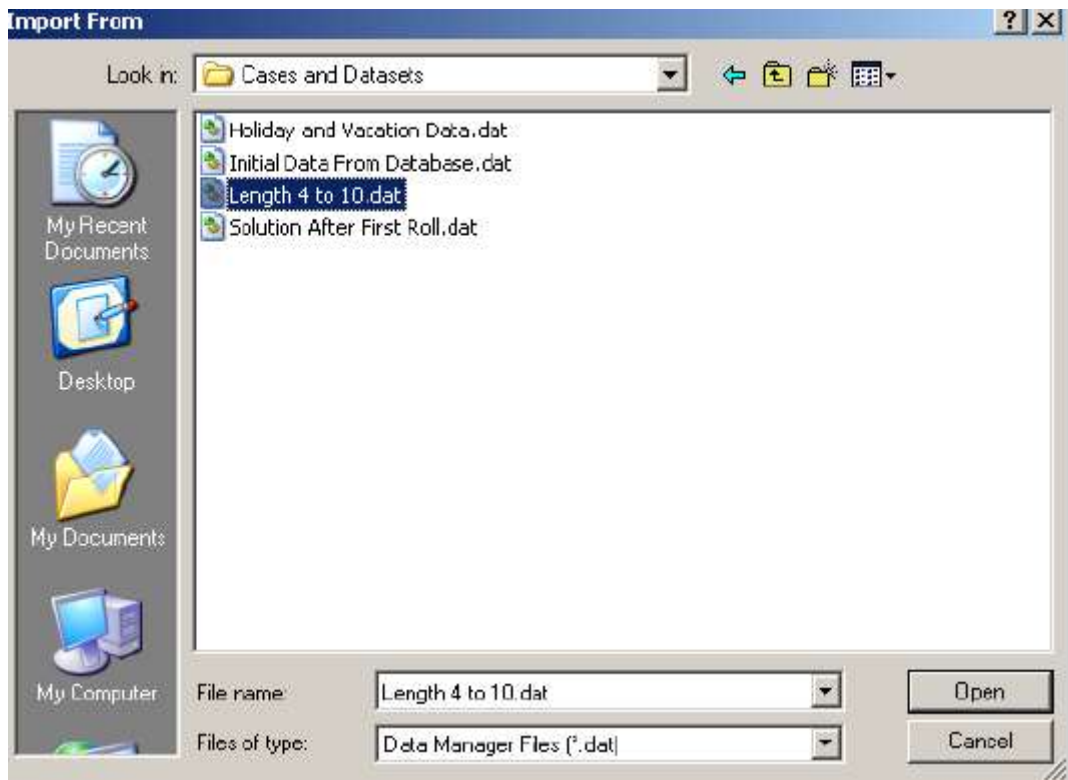


图 4.6: ‘Import From’ 对话框

点击 Open 按钮后，从 Length4 到 length10 的所有案例将自动出现在 Data Manager 界面，如图 4.7 所示：

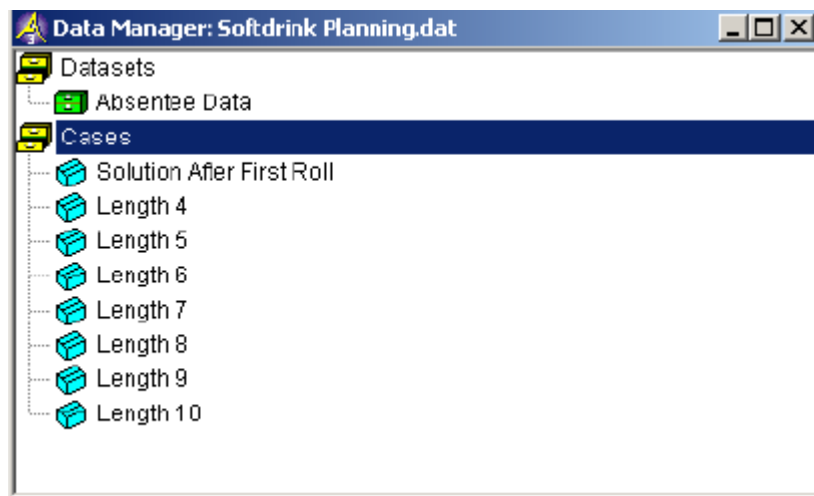


图 4.7 包含导入案例的数据管理器

4.6 导入终端用户菜单

除模型段落，页面，实例，和数据集外，AIMMS 还可导出或导入菜单树和相关条例。本教程 13 章所建立的 ‘Softdrink Planning Menubar’ 以导入到文件 ‘Softdrink Planning Menubar.mnu’，并可用于后续导入，‘Menus’ 子目录包含文件夹：

- ‘Softdrink Planning Menubar.mnu’

本节中，单一菜单条的导入将被说明，你可以按照下列步骤的导入菜单条：

- 按 ‘Ctrl-F9’ 组合键打开菜单生成器
- 移动鼠标到 ‘User Menu Tree’
- 在 ‘Edit’ 目录中选择 ‘Import’ 命令
- 在 ‘Select a file’ 对话框中找到文件 ‘Menus Softdrink Planning Menubar.mnu’
- 点击 ‘Open’ 按钮

根据当前的菜单树的内容提示，决定是否在菜单树中导入新的菜单条，或者重写现有菜单条。当菜单树中仅包含文件默认菜单条（Default Page Menubar）和默认工具栏（Default Page Toolbar），‘Import Menu’ 对话框如图 4.8 所示，点击 ‘Import’ 按钮后，新的菜单条将被导入到菜单树中。

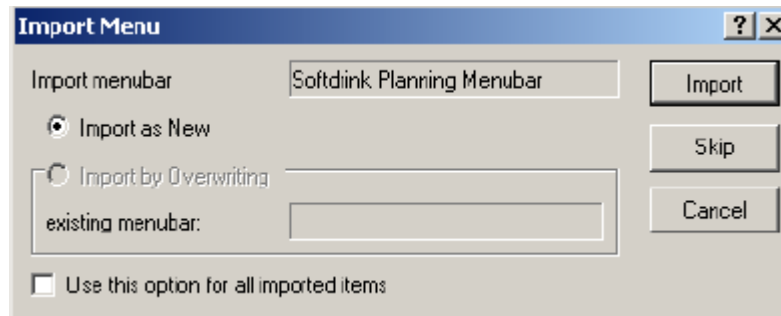




图 4.8 目录导入对话框

在本章中，你将创建一个非常小的 AIMMS 模型和一个终端用户页面，本章的目的是让你对 AIMMS 软件功能有初步认识。


5.1 开始新项目


假设您的机器已经安装了AIMMS 3。如果桌面上存在 AIMMS 3 的快捷键，双击即可启动 AIMMS 3；否则，请按顺序执行以下步骤以启动 AIMMS：

- 按下任务栏上的 ‘Start’ 按钮  Start，
- 进入 ‘Programs’ 子菜单，然后
- 选中并单击其中的 AIMMS 图标 ，启动 AIMMS。

接下来您就会看到AIMMS的启动屏幕画面，AIMMS启动后，启动屏幕画面将会消失，AIMMS 窗口则随之出现。如果在启动的过程中遇到 “Tip of the Day（当天提示）” 对话框，直接关闭，因为此时我们不需要查看其中的内容。

按下 ‘New Project’ 按钮 ，它位于 AIMMS 工具栏的左上角。随后将会弹出图5.1中所示的对话框，此时请执行以下步骤：

- 指定项目名称为 ‘Softdrink Planning’
- 如果不想将项目保存在默认文件夹 “...\\My Documents\\My AIMMS Projects\\Beer Transport” 中，则可以按下 ‘Wizard’ 按钮  选择合适的文件夹，然后
- 按下 ‘OK’ 按钮

注意AIMMS将自动以项目名称扩展项目文件夹命名。自动命名功能图标为 ‘Wizard’ 按钮 。您也可以手动进入项目文件夹，指定项目名称以代替系统自动命名。

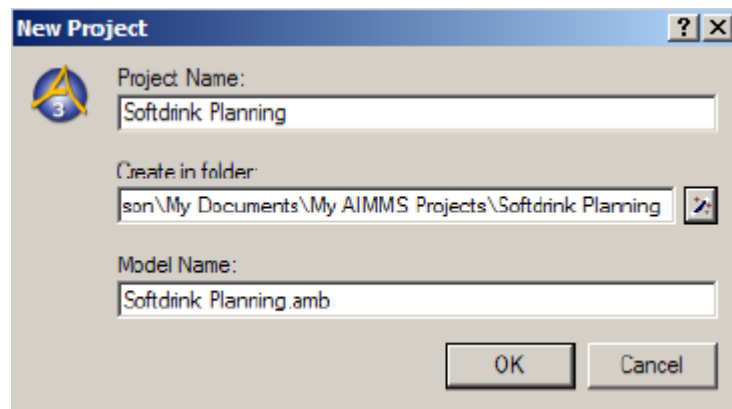



图 5.1 新建项目向导

完成 ‘New Project’ 向导之后，AIMMS 模型浏览器将会自动打开，您可以开始您的项目模型-软饮料计划

注意 AIMMS 工具栏自动添加项目工具栏 ，方便你进一步开发模型和终端界面。可用的工具包括：


- 模型浏览
- 标识符选择器
- 页面管理器
- 模板管理器

- 菜单生成器
- 数据管理器
- 数据管理设置工具

也可以进入工具（Tools）菜单栏使用这些工具

此外，在窗口管理器中，从鼠标右击弹出菜单条中选择‘New-Aimms Project File’命令，创建一个新的项目。本例中如图 5.1 所示的‘New Project’向导会自动弹出。新的 AIMMS 项目将创建在当前子目录中。

5.2 模型浏览器

项目创建后，模型浏览器将会自动打开，初始模型树如图 5.2 所示。模型浏览器也可通过单击工具栏中按钮手动打开，或者按快捷键 F8。在初始模型树中，您将会看到三个预定义步骤和一个预定义的空白声明段落。

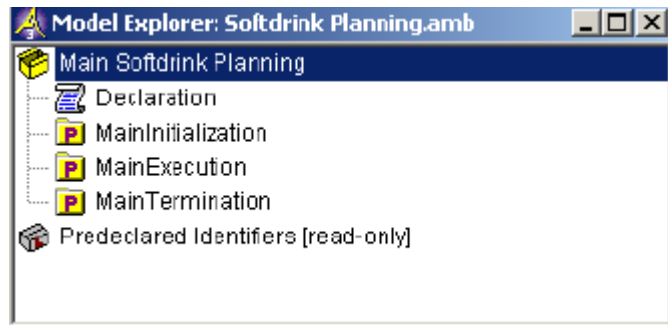





图 5.2 初始模型树

5.2.1 输入集合标识符


声明模型标识符首先要单击滚动图标（不是名称本身）扩展模型节点，然后双击打开模型声明段，你也可以在选择模型树相应节点后，点击右箭头键打开。打开声明段后，工具栏‘New Identifier’按钮将会启动。

执行以下步骤来创建位置集合：

- 按下‘New Set’按钮在模型树中创建一个新的集合标识符，
- 将集合名称指定为‘Location’，然后
- 按下‘Enter’键以记录该名称。

您也可以通过右击鼠标弹出菜单中的‘Insert’命令或者点击‘Edit’菜单栏中的‘Insert’键来创建新的集合标识符。

对于模型树中的每个节点，你可以赋予额外的信息作为其属性。在AIMMS属性栏中，您可以查看或修改这些属性。您可以通过以下方式打开属性栏：

- 选中模型树中的任一节点，按下回车键
- 双击模型树中节点名称
- 选中模型树中的任一节点，点击‘Attributes’按钮

您可能已经发现双击中间节点图标和名称会造成不同结果，双击图标会打开子一级模型树，双击名字则会打开相应的属性栏。

接下来,您需要声明集合“Location(位置)”的属性为索引L,首先打开集合“Location(位置)”的属性栏,初始属性栏如图5.3所示:

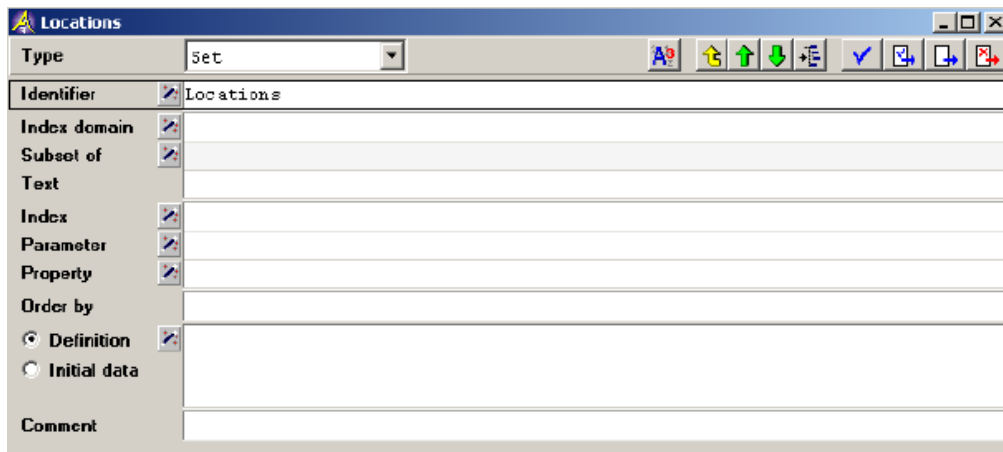




图 5.3: 集合‘Location’的初始属性表

按顺序执行以下步骤,声明集合‘Location’的属性为索引L:

- 将鼠标光标移动到‘Index’属性字段,然后在空白的编辑区域内单击鼠标左键,
- 输入字母‘1’ (没有引号), 然后
- 按下‘Check, Commit and Close’按钮, 完成属性表的输入。

您也可以使用Ctrl-Enter组合键代替使用‘Check, Commit and Close’按钮, 这样也可以执行属性表操作。模型树如图5.4中所示:

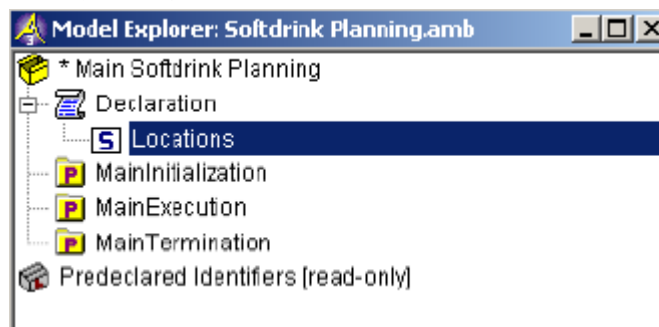




图3.4: 中间过程模型树


模型节点MainSoftdrinkPlanning左边的星号表面所编辑的文件尚未保存。请按工具栏中的‘Save Project’按钮保存文件, 或选择Ctrl-S组合键保存文件。

参数声明与集合声明十分相似。在本章中,描述位置集合中每个位置需要两个参数,他们分别代表地理中的经度(X)和纬度(Y)。您可以按以下流程声明X坐标(L)

- 点击工具栏中的‘New Parameter’按钮, 在模型中创建一个新参数。
- 指定参数名称‘XCoordinate(1)’, 且
- 按‘Enter’键记录该名称

请注意,使用括号将索引域1添加到标识符“XCoordinate”中。

可以用完全相同的方法添加参数‘YCoordinate(1)’.如果输入的信息有误,只需单击该字段即可重新编辑名称信息,您也可以按F2键进入编辑模式。

现在您已成功输入并声明位置集合的两个参数XCoordinate和YCoordinate,最终的模型树如图5.5所示。按下F5快捷键可以快速查看模型的有效性。如出错您将会收到对应的错误信息。模型有效性一旦得到验证,就可以按下‘Save Project’按钮保存。

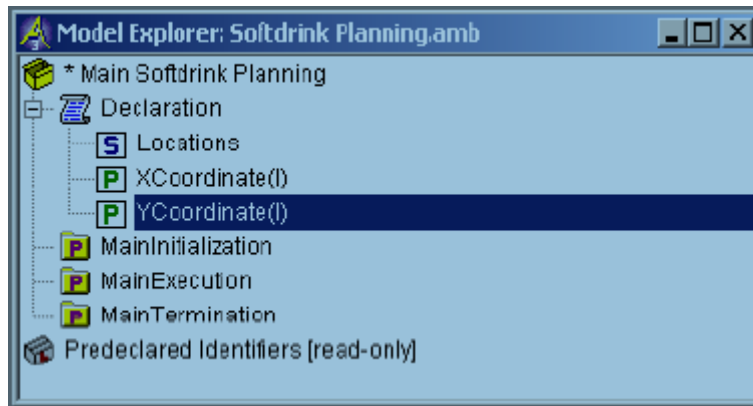


图 5.5 当前模型树

5.3 数据读取

本教程中，为了简要说明 AIMMS 数据读取方面的功能，我们将从 Data 目录中的外部文件夹 ‘Locations.dat’ 读入原始数据，该文件夹包含位置集合的相关数据和位置坐标。如果您想查看原始数据文件夹内容，您可以使用外部文本编辑器打开或者使用 AIMMS 内部的文本编辑器打开，其步骤为：File - Open - Text File. . .menu。在 ‘Open File’ 对话框中，选择 ‘All Files (*.*)’ 类型即可找到文件 ‘Locations.dat’。如果您使用 AIMMS 文本编辑器打开，结果如下所示：.

Locations.dat dir:C:\Users\Jay Johnson\Aimms...

COMPOSITE TABLE:

l	XCoordinate	YCoordinate
Amersfoort	5.377	52.158
Amsterdam	4.88	52.376
Apeldoorn	5.953	52.213
Arnhem	5.917	51.982
Assen	6.561	53.011
Breda	4.778	51.588
'Den Bosch'	5.308	51.701
'Den Haag'	4.303	52.079
'Den Helder'	4.754	52.958
Deventer	6.158	52.263
Dordrecht	4.679	51.794
Eindhoven	5.461	51.432
Emmen	6.885	52.788
Enschede	6.89	52.22
Groningen	6.574	53.226
Haarlem	4.618	52.382
Leeuwarden	5.782	53.212
Maastricht	5.696	50.857
Nijmegen	5.845	51.84
Rotterdam	4.482	51.929
Tilburg	5.071	51.568
Utrecht	5.118	52.107
Venlo	6.158	51.374
Vlissingen	3.571	51.458
Zwolle	6.09	52.522

图 5.6: 文件 ‘Locations.dat’ 使用 AIMMS 文本编辑器打开

在 AIMMS 中, 如果使用数据文件 ‘Locations.dat’ 初始化数据, 则需要在标准 ‘Main Initialization’ 程序中输入阅读声明, 一旦打开此项目, 该程序将自动运行。您可以按以下步骤实现:

- 在模型树中选择 ‘MainInitialization’ 子程序节点
- 打开该程序的属性栏
- 在主体声明中输入下面一行文字:

read from file "Data\\Locations.dat";

- 完成属性窗口后, 请点击 Check, commit and close 按钮 

注意在 AIMMS 中, ‘MainInitialization’ 子程序的主体声明中使用双反斜杠符号, 单反斜杠用在 AIMMS 字符串中有其他作用, 这与 C 语言的约定符号相关联。例如: ‘\n’ 代表 ‘return’, ‘\t’ 代表 ‘tab’。

图 5.7 演示了 ‘MainInitialization’ 子程序的声明内容



图 5.7: ‘MainInitialization’ 子程序中完成后属性栏

您可以不必打开此项目即可运行 ‘MainInitialization’ 子程序:

- 在模型树中选择 MainInitialization 子程序节点
- 鼠标右击, 在弹出菜单中选择 ‘Run Procedure’ 命令 (如图 5.8 所示)

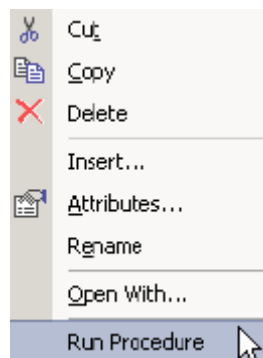




图 5.8 鼠标右击弹出菜单栏

数据一旦在 AIMMS 中读取, 所有模型标识符将被初始化, 您可以在一个或多个界面查看当前数据值。例如, 你可以按下面步骤打开 ‘XCoordinate’ 标识符数据页面:

- 在模型树中选择参数 ‘XCoordinate’, 然后
- 鼠标右击, 在弹出菜单中选择 ‘Data’ 命令

数据页面如图 5.9 所示, 点击 ‘Left Arrow’ 按钮  即可得到位置集合的数据页面, 点击 ‘Right Arrow’ 按钮  则可得到参数 ‘YCoordinate’ 的数据页面。

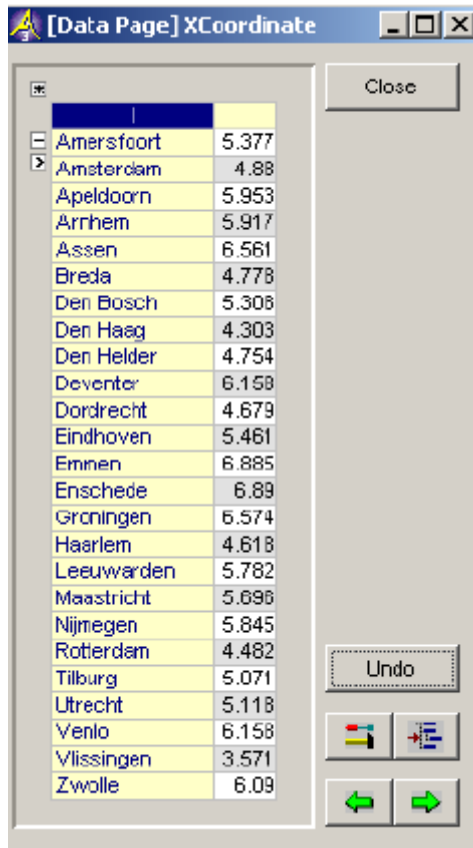




图 5.9: 参数 XCoordinate 的数据页面

5.4 第一个页面

为了说明 AIMMS 的图像功能，现在我们创建一个包含网络对象的页面，该网络对象能在地图上显示地理位置。AIMMS 以图和表的形式实现数据对象的可视化。

您可以按以下步骤创建一个新的页面：

- 点击工具栏中的‘Page Manager’按钮  (或者，按 F9 键)
- 点击工具栏中的‘New Page’按钮  创建新的页面
- 指定‘Location’作为新页面名称
- 按‘Enter’键注册该页么

新页面的页面管理器如图 5.10 所示：

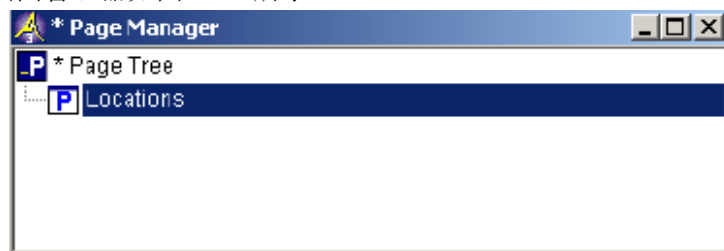



图 5.10 单一页面的页面管理器


编辑模式和使用模式是两个重要的页面模式，编辑模式是用来创建和修改页面对象，


而使用者模式可以查看和编辑显示页面对象所需的数据。

打开页面编辑模式：

- 在页面管理器选择 ‘Location’ 页面
- 在工具栏中点击 ‘Edit Mode’ 按钮，打开编辑模式中已选页面

按以下操作创建一个新的网络对象

- 点击工具栏中 ‘New Net Object’ 按钮
- 将鼠标光标放置该对象左上角
- 按下鼠标左键，将鼠标光标拖拽到屏幕的一点，使产生的矩形长宽比大约为2
- 松开鼠标左键

此时网络对象框出现，请使用对话框中的 ‘Wizard’ 按钮填写节点 ‘Node index’ 、 ‘X coordinate’ 和 ‘Ycoordinate’ ，如图5.11所示，注意节点字段中，您所填写的是字母1而不是数字1.

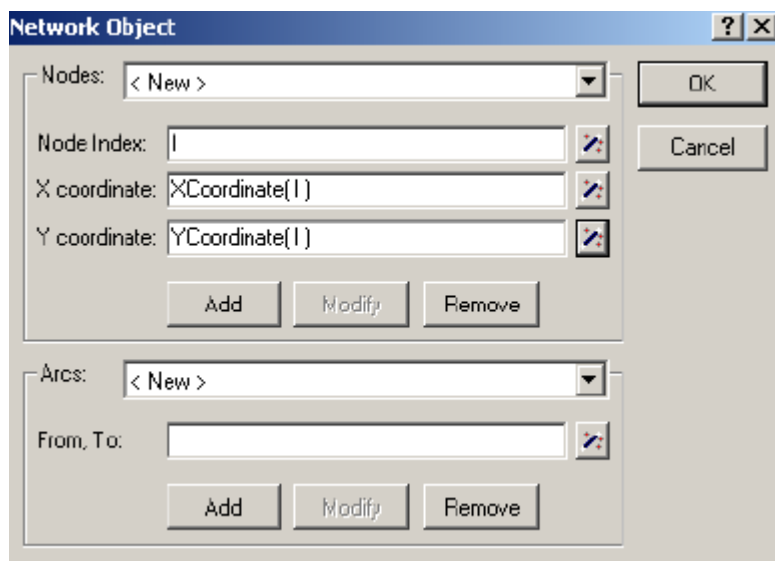


图5.11 网络对象对话框

按下 ‘OK’ 按钮。 此时所创建的网络对象如图5.12所示，如果增加适当的背景地图的话，这些位置将表现的更加直观。



图5.12初始网络对象

如果想为网络对象提供背景地图,你需要更改其属性,你可以按下列步骤实现这一目的:




- 点击工具栏中‘Properties’按钮进入属性对话框
- 选择‘Background’选项
- 点击背景属性栏中的右边‘No Image’,然后按按钮,在自动弹出目录中选择‘From File’命令
- 点击图像文件名的值域,按按钮,在自动弹出的目录中选择‘Select From File’命令,然后选择文件夹‘Bitmaps\Netherlands.bmp’
- 在编辑区域的左边输入3.3,右边输入7.3,顶部输入53.5,下部输入50.7,从而定位图片位置
- 点击‘Apply’按钮而不是OK键

图5.3展示了包含背景地图的网络对象:



图 5.3 中间过程的网络对象

刚刚输入的四个数可以定位地图以匹配这些位置。其数值表示地图边界的经度和纬度坐标。虽然地图和位置已固定下来，但当前矩形地图的尺寸不是固定的，矩形坐标也应该和地图的坐标相一致。

在专业应用中，一般使用模型标识符修改矩形的尺寸，从而控制网络对象的缩放和滚动。本章中，矩形的坐标值被设置成和地图的坐标相等，您可以按以下步骤完成网络对象的布局设计：

- 选择 ‘Network’ 选项
- 如图所示在四个编辑区域填入数字
- 取消选中四个复选框，然后
- 点击 OK 键记录该页面

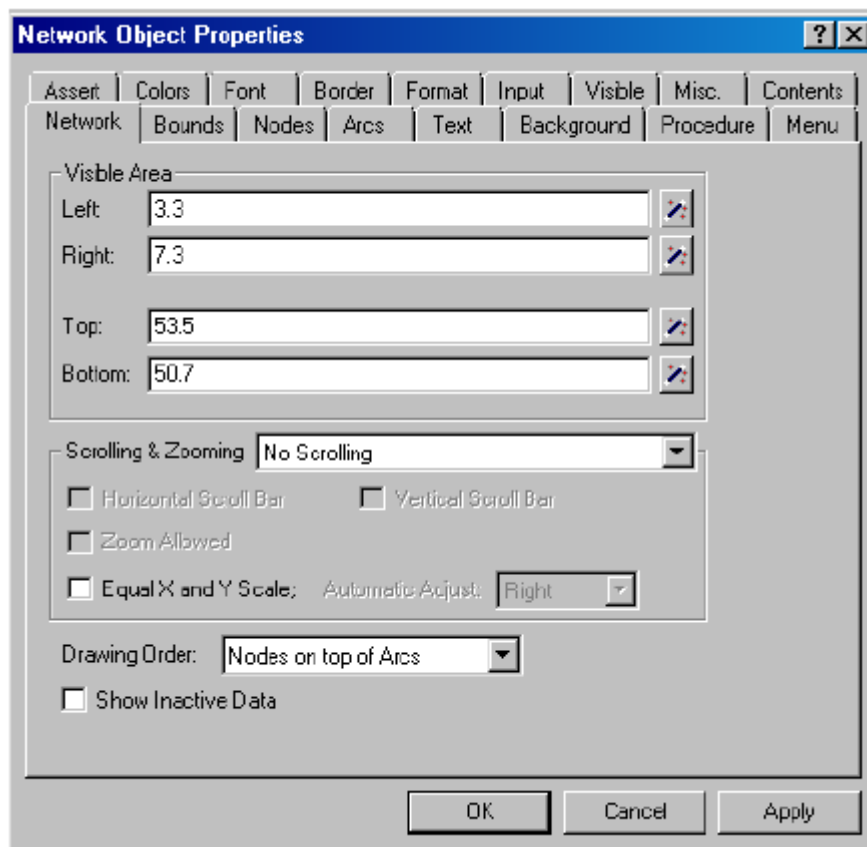



图 5.14 网络性质对话框

页面标题栏左边的星号表明页面补充工作尚未保存，请选择工具栏中‘Save Project’按钮  保存当前工作。


您现在可以通过点击工具栏的‘Page User Mode’按钮  调整页面到使用者模式。最终的网络对象如图 5.15 所示，注意图片上城市名字并非地图原本就有的，而是基于节点集的内容叠加而成的。






图 5.15 最终网络对象

6.1 模型结构

前文定义的‘初始模型树’主要是帮助初学者建立包含单一固定数据集的小模型，模型的全部声明放在同一声明段，然后在自初始化程序中输入初始数据，而处理数学规划指令被放入执行程序中，在这本内容更丰富的教程中，您将学会如何建立一个更完整的模型树。

当你试图建立一个更大的模型时，使用段落的概念是有益的，使用‘段落’可以让您更有效的组织管理模型，这种方法就是把模型中相似的部分放在一起。合理的模型组织能够让您和您的同事在模型的使用周期内更好的维护模型。本教程中，模型表示部分主要包含两个模型段：一个方便开发整个模型，其内容将在 4, 5 部分介绍，另一段落主要服务于用户界面，将在第 6 部分介绍。这些模型‘段落’会被细分成几个小的子段落，以体现其附加结构。本章中，首先细分第一个模型段。请按以下步骤创建两个主模型段：

- 首先在模型树中选择根节点‘Main Softdrink Planning’
- 点击工具栏中的‘New Section’按钮，在模型树中创建一个新的节点
- 该节点取名‘The Model’，然后按 Enter 键注册
- 再次点击工具栏中的‘New Section’按钮，创建第二个节点
- 第二个节点取名‘The User Interface’然后按 Enter 键结束

第一个主段落将被分成六个更小的子段落，首先请鼠标双击图标打开段落。打开段落，书的图标变为一本打开的书，假如你双击书名而非书的图标，则会进入段落的属性窗口，如果这样，请关闭此窗口，然后重新双击确保你打开段落。同创建主段落类似方式创建子段落，此时，创建的子段落结构如图 6.1 所示：

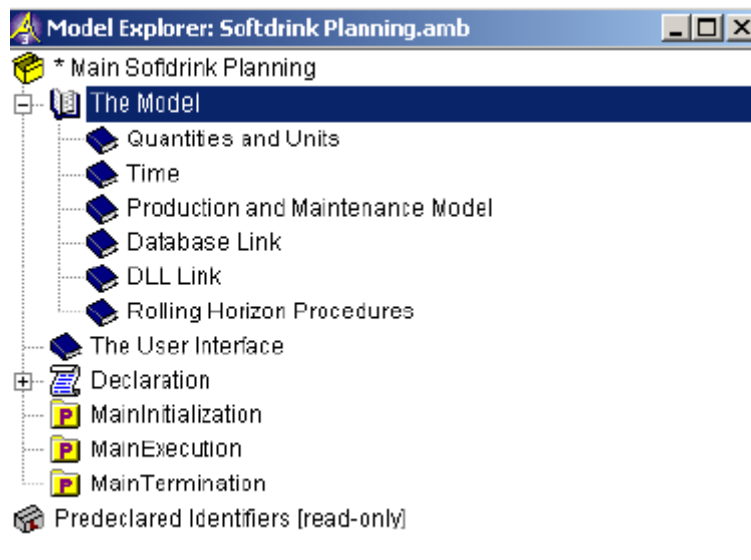



图 6.1 模型的段落结构




6.3 进入数量声明

以上全部段落结构完成后，既可以准备在‘Quantities and Units’段落下指定第一个声明段落。请按下列步骤创建声明段落：

- 双击图标  打开模型‘Quantities and Units’段落
- 点击‘New declaration’按钮创建新的声明段
- 输入‘Quantities and Units’作为新声明段落名称
- 按 Enter 键注册

开发应用程序时，通常开始都是对数量和单位进行声明，因为，完成模型所需参数和变量声明都需要单位。在本教程第三章，模型中用到公升和卡车容量作为体积单位，在 AIMMS 软件中，首先对体积进行数量声明，体积是一个标准的国际数量单位，也是目前 AIMMS 软件中的基本单位，其基本单位名称为立方米，而在此模型中，声明单位公升和卡车容量以代替立方米单位。

请按以下步骤声明体积数量

- 通过双击卷型图标  打开声明段落‘Quantity Declarations’
- 点击工具栏中‘Other’按钮 （或者按‘Insert’键）
- 在标识符对话框中选择数量类型按钮 ，然后按 OK 键确认
- 按照‘Press enter to select a SI Quantity’指示从一系列预定义的国际数量单位中进行挑选
- 选择‘SI Volume’数量，然后按 OK 键确认
- 如图 6.2 所示，选择第二个选项，然后
- 按 OK 键确认

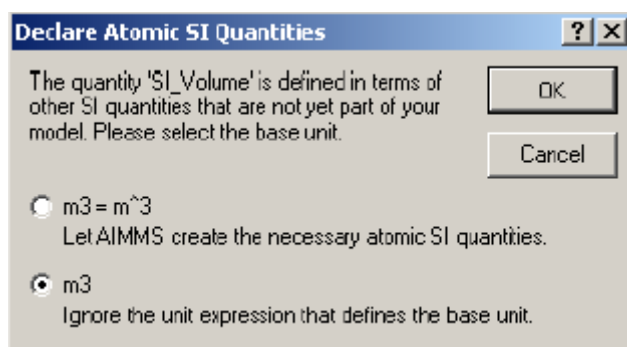


图 6.2 忽略单位表达对话框

现在打开‘SI Volume’属性窗口，输入单位[h1]和[TL]的单位转换因子，初始的‘SI Volume’的属性窗口如图 6.3 所示：

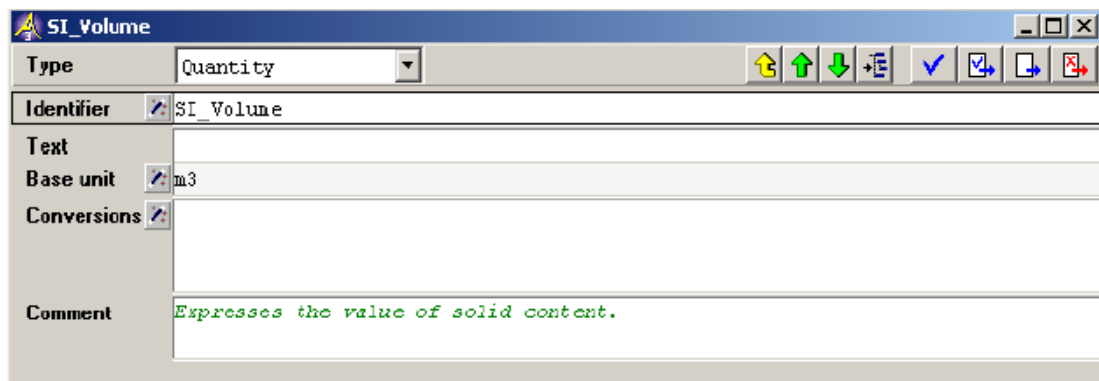




图 6.3 初始的 SI Volume 的属性窗口

然后按以下步骤指定第一个单位[h1]（公升）：

- 打开前文所说的数量‘SI Volume’属性窗口
- 点击‘Wizard’按钮进行属性转换
- 从‘Derived Units’列表框中选择‘L’（代表公升）
- 从‘Decimal Scaling’列表框中选择‘hecto’，然后
- 点击‘Transfer’按钮接受新单位‘h1’定义

所选的导出单位L和相应的十进制‘hecto’如图 6.4 所示：

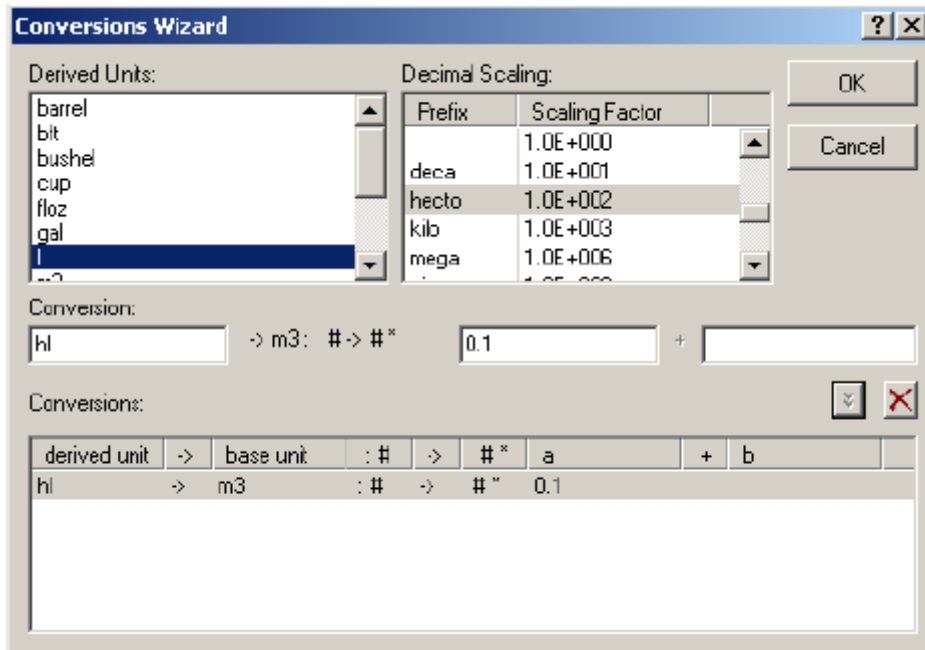




图 6.4 转换向导选择

现在准备输入第二个单位[TL]，每[TL]等于 12 立方米，注意[TL]是一个自定义的单位，此时，图 6.4 的两个列表框并不支持该单位定义，请按以下步骤执行此：

- 考虑标题为‘Conversion’（包含 hl）的编辑字段，将里面内容改为字母‘TL’（不加引号）
- 考虑右边的编辑字段（包含 0.1），将里面内容改为数字‘12’（不加引号）
- 和前面一样，点击‘Transfer’按钮接受新单位‘h1’的定义
- 按‘OK’键确认，完成单位[h1]和[TL]的指定

此时，属性窗口如图 6.5 所示，通过点击‘Check, Commit and Close’按钮确认 AIMMS 是否接受已完成的属性窗口，如果没有错误，AIMMS 将执行其内容然后关闭窗口。

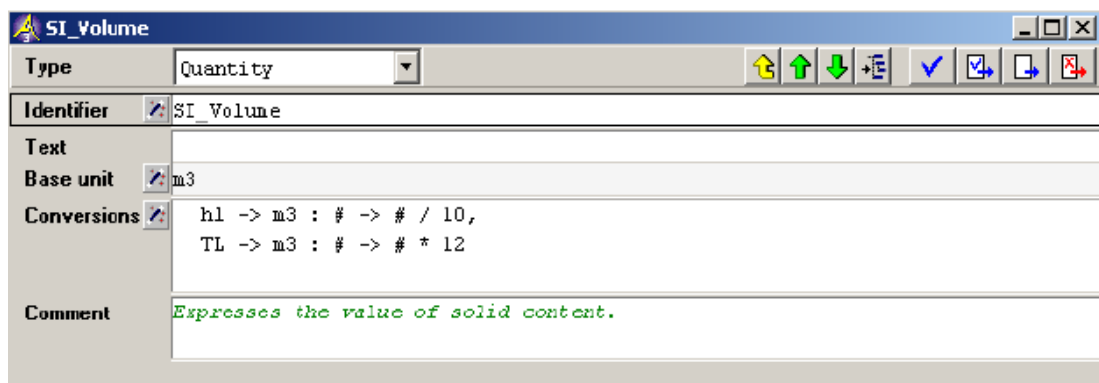



图 6.5 数量 SI Volume 的最终属性窗口

如果要表示金额,则需要声明货币单位,货币单位不是标准国际单位,需要单独被指定。本教程中,你仅需使用美元货币,无需美元与其他货币的换算。请按以下步骤声明货币单位:

- 声明货币单位
- 输入 '\$' (没有引号) 作为单位属性, 然后
- 按 'Check, Commit and Close' 按钮 

最后介绍单位 'SI quantity SI Time Duration', 该单位的基本单位默认为秒 (s)。然而, 在应用中, 基础单位天 (day) 更适合我们的模型, 通过工具栏的向导按钮将基本单位由秒 (s) 改为天 (day), 当 AIMMS 询问是否保存数据时, 选择 'No', 此外, 请使用转换向导制定天和周的换算关系。最终属性窗口如图 6.6 所示:

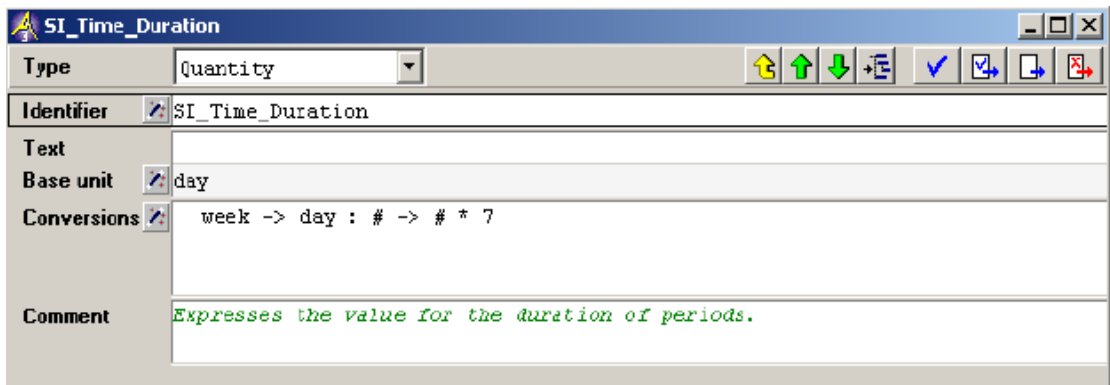


图 6.6 单位 SI Time Duration 的最终属性窗口

当前模型树如图 6.7 所示:

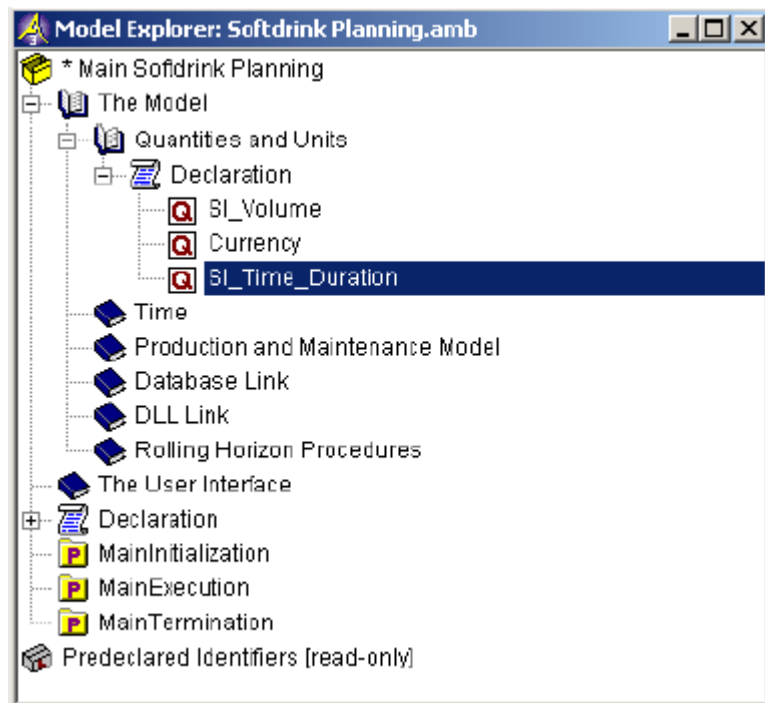



图 6.7 包含全部标识符的模型树

再次声明, 模型管理器中模型节点左端星号暗示补充内容尚未保存到硬盘, 请点击工具栏最左边的保存按钮  保存当前内容。

6.3 输入时间声明

AIMMS 基于模型中时间应用提供两种标识符，分别为日历和范围，日历和范围处理时间问题有其特定用途。本教程中，它们都将使用，而带索引的时间集合（时间表）会将这两种标识符联系起来。

学习本教程的经验告诉我们，如果想对高级概念有更清楚的认识，仅仅学习一次是不够的。

日历是具有单位长度的连续时间集合，它能覆盖从日历开始时刻到结束时刻的完整模型框架，我们也可以用一个带索引的时间代替日历时间，本教程中，将介绍天历和周历。

范围被用在数学规划上表示抽象时间周期集合，周期被分成三部分，也称三个时区，其中最重要部分是计划区间（planning interval）。计划区间之前时间称为过去（The Past），计划期间之后时间称为未来（The Beyond）。当变量和约束通过范围时间索引化后，AIMMS 将自动把变量和约束限制在包含时间区间的范围内。

时间表可以是带索引的集合，也可以是带索引的元素参数，他们将模型时间段连接到范围的区间或者日历的时间空档。基于时间表，AIMMS 函数可以将‘日历’时间聚合成‘范围’时间，同样的，也可以将范围时间分解成日历时间。图 6.8 阐述了范围和日历关系的时间表例子。

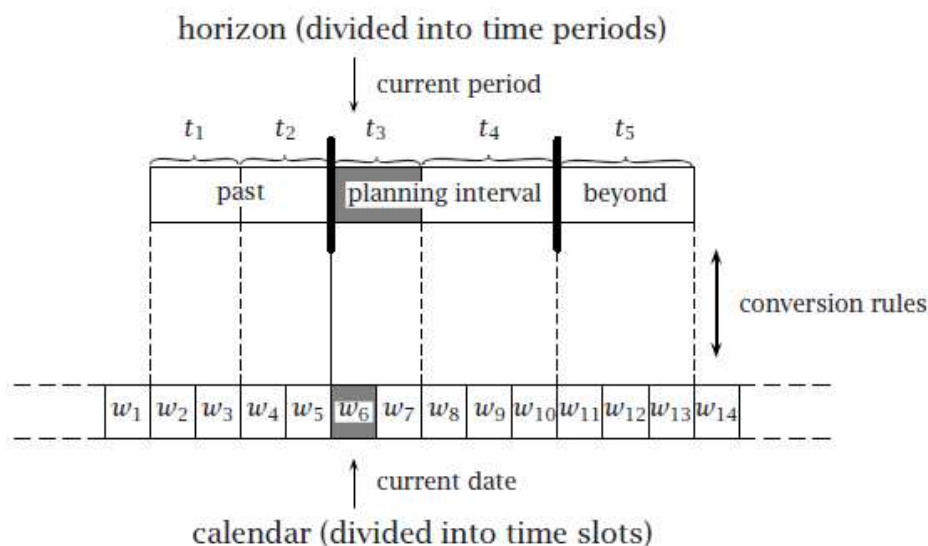


图 6.8 连接日历和范围

图 6.8 中例子相关联的时间表如图 6.9 所示，本例中，时间表命名为‘Timeslot To Period’。

TimeslotToPeriod	
TimeslotToPeriod(t_1)	{ w_2, w_3 }
TimeslotToPeriod(t_2)	{ w_4, w_5 }
TimeslotToPeriod(t_3)	{ w_6, w_7 }
TimeslotToPeriod(t_4)	{ w_8, w_9, w_{10} }
TimeslotToPeriod(t_5)	{ w_{11}, w_{12}, w_{13} }


图 6.9: 与图 6.8 相关的时间表

本教程中，为了能够把时间相关标识符放在一起，需要在时间模型段落下面创建两个单独的子声明段落。请按以下步骤执行：


- 在模型中，选择时间段落节点
- 创建一个新的声明段落 Period Declarations，然后
- 创建一个新的声明段落 Calendar Declarations

6.3.1 范围声明

首先声明段落 ‘Period Declarations’ 第一个参数 ‘NumberOfPeriods’，请按以下步骤执行：

- 打开声明段落 ‘Period Declarations’
- 点击工具栏中的 ‘New Parameter’ 按钮 ，在模型树中创建一个新的参数
- 命名新参数为 ‘NumberOfPeriods’，然后
- 按回车键记录名称

为了完成新参数 ‘NumberOfPeriods’ 声明，请打开其属性窗口，然后按以下步骤执行：



- 在整数区间输入 ‘{1..inf}’（没有引号）作为区间属性
- 点击 ‘Definition/Initial Data’ 属性前面的单选按钮 ‘Initial Data’
- 在 ‘Initial data’ 中输入数字 ‘10’（没有引号），然后
- 点击 ‘Check, Commit and Close’ 按钮  提交您的编辑文档

注意 AIMMS 中，整数区间总是被附波形括号，而连续区间附上方括号。

‘Range’ 属性能让 AIMMS 在系统运行期间完成区间检查，因为整数集 ‘{1..inf}’ 表示全体正整数集合，所以一旦将非整数，或值小于 1 的数分配给参数 ‘NumberOfPeriods’，AIMMS 将自动报告错误。

第二个参数 ‘NumberOfPeriodsInPlanningInterval’ 可以按同样方式进行声明，输入 ‘{1..inf}’ 作为 ‘Range’ 属性，输入 ‘8’（没有引号）作为 ‘Initial data’ 属性。

您可以按照下列步骤进行范围声明：

- 在工具栏中点击 ‘Other’ 按钮 
- 点击范围键 ，然后按 ‘OK’ 键
- 命名 ‘Periods’，然后
- 按 Enter 键记录名称

接下来，打开属性窗口，输入索引和当前时间属性：

- 再次按回车键打开 ‘Periods’ 的属性窗口
- 移动光标到 ‘Index’ 属性旁边的空白编辑区域，输入字母 ‘t’（没有引号），然后
- 输入 ‘period-01’（没有引号）作为 ‘Current period’ 属性

接下来考虑区间长度属性，使用 AIMMS 中非常方便的 ‘文件名完成’ 功能，避免重复输入长标识符名称。

- 在 ‘Interval length’ 属性旁边的空白编辑区域输入字母 ‘N’
- 按 Ctrl-Spacebar 组合键，AIMMS 自动弹出全部标识符列表，选择字母

- ‘N’ 扩展（如图 6.10 所示），然后
- 选择 ‘NumberOfPeriodsInPlanningInterval’ 作为标识符名称，然后按回车键

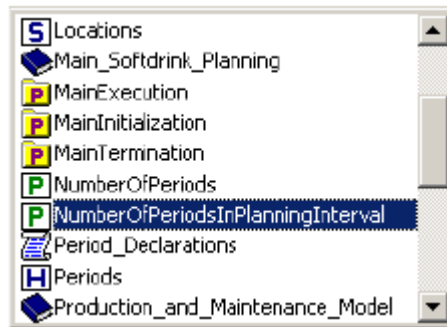


图 6.10 ‘文件名完成’ 弹出框

考虑图 6.11，完成定义的属性，您可以再次使用文件名完成功能，选择 ‘NumberOfPeriods’ 作为函数 ‘ElementRange’ 第二个自变量：

- 按图 6.11 所示输入定义
- 点击 ‘Check, Commit and Close’ 按钮 ，提交编辑内容

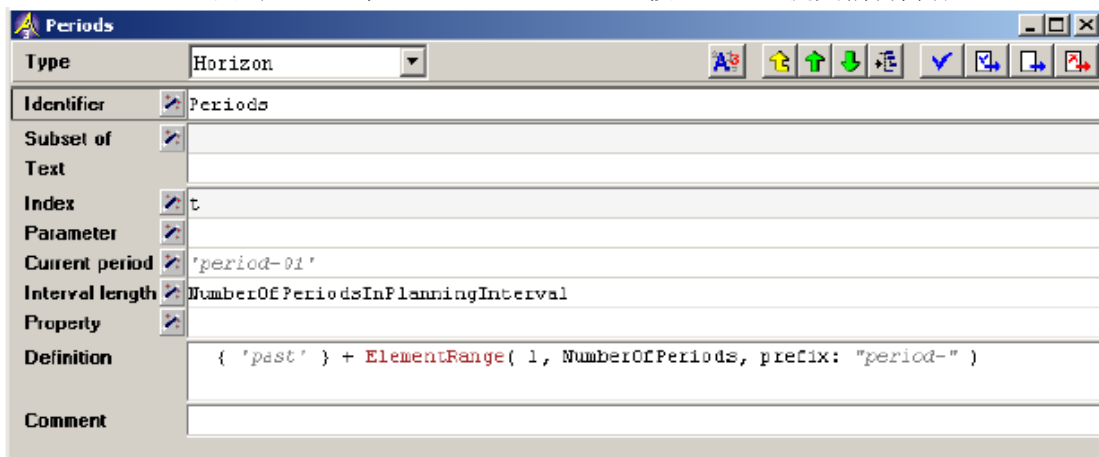


图 6.11 ‘Periods’ 的最终属性框

‘文件名完成’ 功能常被用来处理未完成的标识符名称，除此之外，你也可以拖拽模型树中的标识符名称到编辑区域，这两个功能可以帮你避免文件名的输入错误，确保名称的一致性并加快您的工作。

对于那些整数值的集合，函数 ‘ElementRange’ 允许读者动态地创建或改变其内容，本教程中，这些元素分别是：‘period-01’，‘period-02’ 等等，一直到参数 ‘NumberOfPeriods’ 的值。除了字母 ‘from’ 和 ‘to’ 可能在形式参数名的前面，这两个参数是固定在前面的，剩下的参数位置可自由选择，只要不是错误命令，形式参数一定在其前面。

输入函数名后，进入开括号（或者移动鼠标指针在函数名前摆动），AIMMS 自动弹出快速信息提示窗口如图 6.12 所示，信息提示窗口显示 ‘ElementRange’ 函数的参数信息。如果想关闭信息提示窗口，您可以进入闭括号（或者通过移动光标使其位于参数列表外）

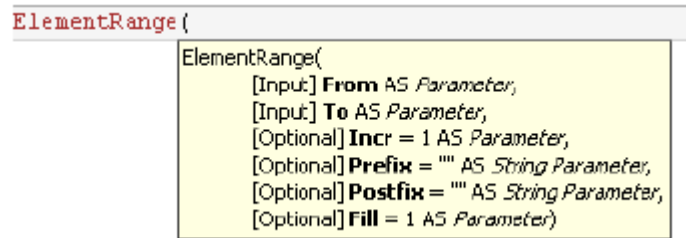


图 6.12: ‘ElementRange’ 函数信息提示窗口

AIMMS 中，读者可以快速获取函数参数的类型和顺序的相关信息，或者从帮助文件中获取某一文件，请按以下步骤打开 AIMMS 的函数参考文本：

- 移动鼠标到关键字 ‘ElementRange’ 文本光标位置
- 右击弹出菜单中选择 ‘Help on’ 命令，然后
- 在 ‘Help on’ 子目录中选择 ‘ElementRange’ 条目（如图 6.13 所示）

此时，‘Acrobats’ PDF 查看器会在相应页面打开函数参考文本

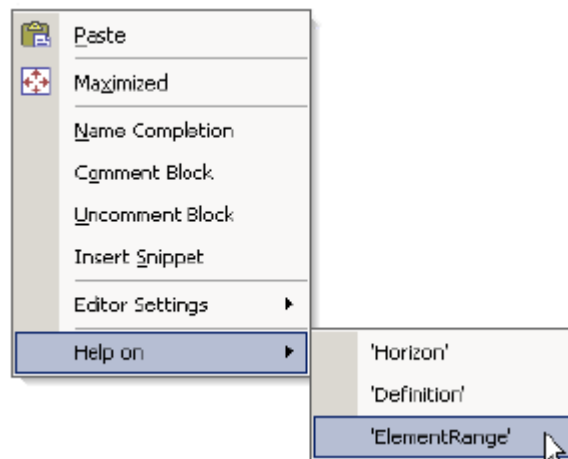





图 6.13 右击弹出菜单栏


余下 4 个期间声明涉及 3 个数值参数：阶段理想天数，阶段理想周数，阶段实际天数（反映周末和官方节日），和所谓的表示计划区间第一阶段的元素参数。最后一个并非数值，而是集合 ‘periods’ 中元素。



周理想天数是 7 天，但由于周末和官方节日存在，周的实际天数小于 7，您可以按以下步骤对参数 ‘DesiredNumberOfDaysInPeriod’ 进行声明：

- 直接在范围 ‘periods’ 下方嵌入新参数
- 将新参数命名为 ‘DesiredNumberOfDaysInPeriod(t)’，然后按回车键
- 打开属性窗口
- 输入数字 ‘7’（没有引号）作为其 ‘Definition’ 属性，然后
- 按 ‘Check, Commit and Close’ 按钮 ，提交编辑内容

因为参数 ‘DesiredNumberOfWeeksInPeriod’ 与参数 ‘DesiredNumberOfDaysInPeriod’ 相似，通过复制参数 DesiredNumberOfDaysInPeriod 完成标识符的声明，请按以下步骤执行：

- 在模型中选择标识符 DesiredNumberOfWeeksInPeriod
- 点击工具栏中的 ‘copy’ 按钮 （Ctrl-C 组合键）
- 点击工具栏中的 ‘Paste’ 按钮 （或者，按 Ctrl-V 组合键）
- 按 F2 键，将名称 CopyDesiredNumberOfDaysIn-Periods(t) 改为 DesiredNumberOfWeeksInPeriod(t)

- 按回车键确认修改
 - 按回车键打开属性窗口
 - 在属性定义中，将数字‘7’改为‘1’（没有引号），然后
 - 点击‘Check, Commit and Close’按钮，提交编辑内容
- 一旦改变模型树中某个标识符名称，AIMMS 将相应改变与此标识符相关的全部引用再声明一个带索引的参数 ActualNumberOfDaysInPeriod(t)，用以表示天数。您可以按以下步骤执行：

- 嵌入一新参数
- 输入新参数名称‘ActualNumberOfDaysInPeriod(t)’，然后按回车键
- 打开属性窗口，点击‘Unit’属性栏中‘Wizard’按钮
- 选择‘SI Time Duration’作为数量，‘day’作为单位
- 然后按‘OK’键
- 输入‘takes into account the weekends and the official holidays’（不加引号）作为‘Contents’属性，然后
- 点击‘Check, Commit and Close’按钮，提交编辑内容

最终属性窗口如图 6.14 所示：

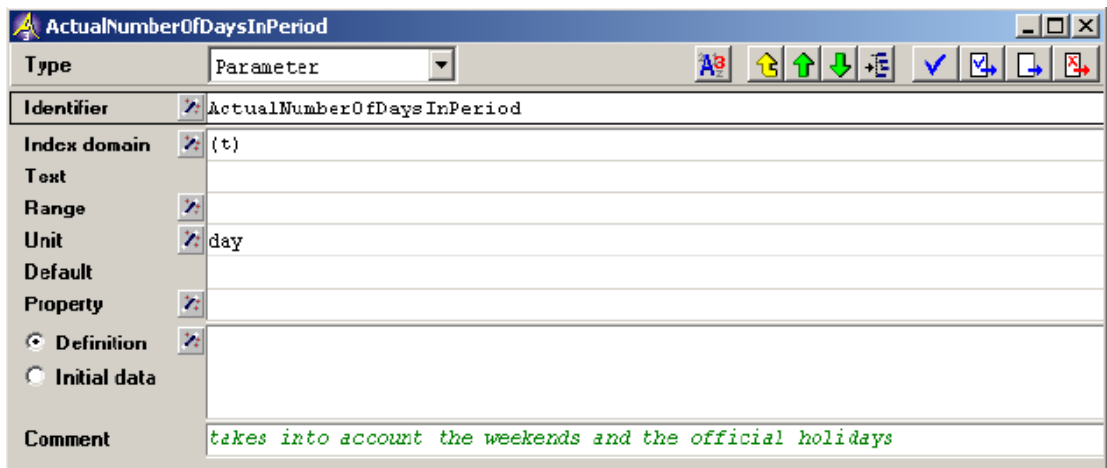



图 6.14：参数 ActualNumberOfDaysInPeriod 的属性窗口

通过对计划期间的第一阶段声明独立参数，代替之前使用的元素‘period-01’，实现对模型和数据之间的分离，请按以下步骤执行：

- 点击工具栏中‘New’按钮
- 选择元素参数类型图标，然后点击‘OK’按钮
- 输入元素参数的名称‘FirstPeriodInPlanningInterval’然后按回车键注册

按以下步骤完成相应的属性窗口：

- 按回车键打开属性窗口
- 使用‘Range’向导，选择范围集合‘Periods’
- 输入‘first(t / t in Periods.Planning)’（没有引号）作为定义，然后
- 点击‘Check, Commit and Close’按钮，提交编辑内容

此时，模型树如图 6.15 所示：

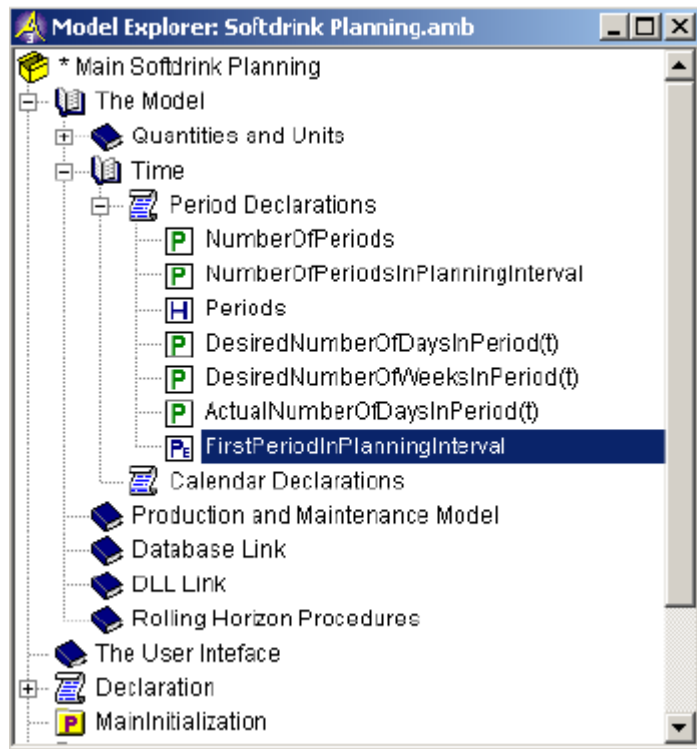




图 6.15 模型树中全部 ‘Periods’ 声明

6.3.2 日历声明

引入字符串参数，读者可以动态的修改日历的始末时间，再次实现模型和数据的分离，您可以按以下步骤对第一个参数进行声明：


- 打开 ‘Calendar Declarations’ 声明段落
- 点击工具栏中的 ‘Other’ 按钮 
- 点击字符串参数类型图标 ，然后按 OK 键
- 输入字符串参数名称 ‘BeginDateOfCalendar’，然后按回车键记录名称


重复上面后三步操作，完成第二个参数 ‘EndDateOfCalendar’ 声明。

按以下步骤完成属性形式：

- 选择字符串参数 ‘BeginDateOfCalendar’
- 按回车键，打开其属性窗口
- 在开始时间的定义栏中输入字符串 “2000-07-01”（带引号），然后
- 点击 ‘Check, Commit and Close’ 按钮，或者按 *Ctrl-Enter* 组合键，提交编辑内容

重复上面步骤完成字符串参数 ‘EndDateOfCalendar’，在定义栏中输入 “2001-06-30”（带引号）。在 AIMMS 中，要求使用日期格式 (yyyy-mm-dd) 表示开始时间和结束时间。对于时隙格式，使用 ‘Timeslot format’ 属性栏进行专门指定，您可以按以下步骤声明日历天数：

- 点击工具栏中 ‘Other’ 按钮 

- 点击日历类型图标, 然后按 ‘OK’ 按钮
- 输入名称 ‘days’, 然后
- 按回车键记录名称

打开日历的属性形式窗口, 使用向导按钮完成属性域 (如图 6.16 所示), 完成 ‘Begin date’ 和 ‘End date’ 属性后, 从弹出菜单栏中执行 ‘Select String Parameter’ 命令, 然后选择相应的字符串参数。

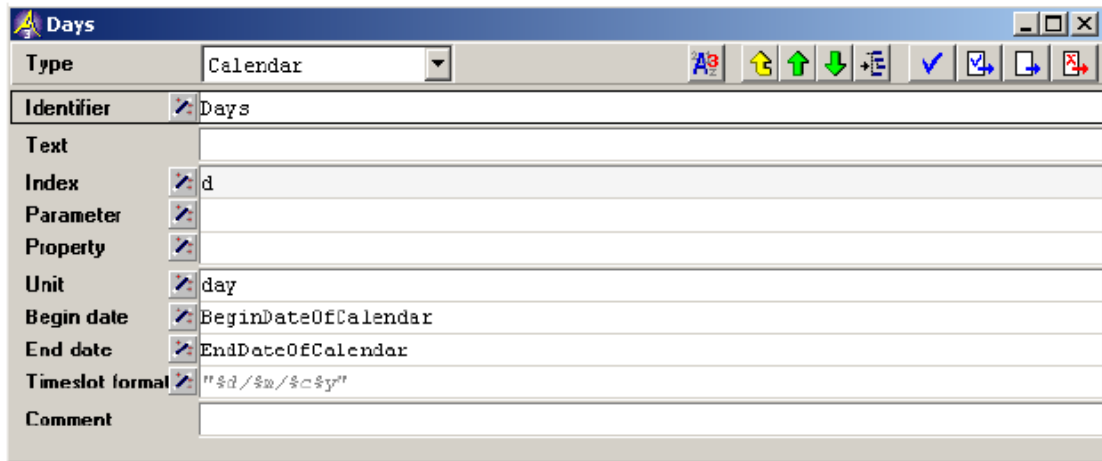


图 6.16: 日历 ‘days’ 最终属性形式

使用向导功能完成 ‘Timeslot format’ 属性后, 从弹出的菜单栏中选择 ‘静态字符串’ (Select Static String), 然后 AIMMS 打开 ‘时隙格式’ (Timeslot format) 向导框构建相应的时隙格式, 通过向导, 不仅可以从一系列基础格式 (包含与您的电脑中区域设置相关的时间格式) 中选择合适的格式属性, 而且也可以自定义格式, 相关操作通过继续阅读本教程即可了解。本教程中, 可选的时隙格式如图 6.17 所示:

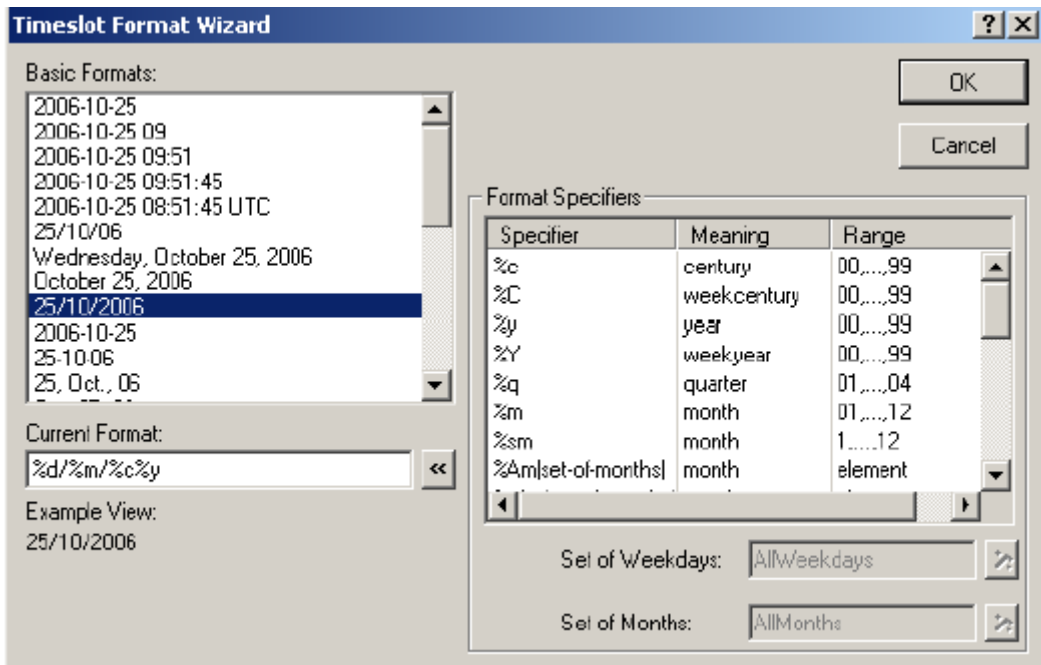


图 6.17: ‘Timeslot format’ 向导

日历天数中几个常用子集的使用将贯彻本教程的模型始终, 基于当前所学, 您应学会了如何输入这些集合。注意如果声明这些子集, 必须通过使用向导 ‘Subset of wizard’ 完

成输入（如图 6.18 所示），而非手动输入。

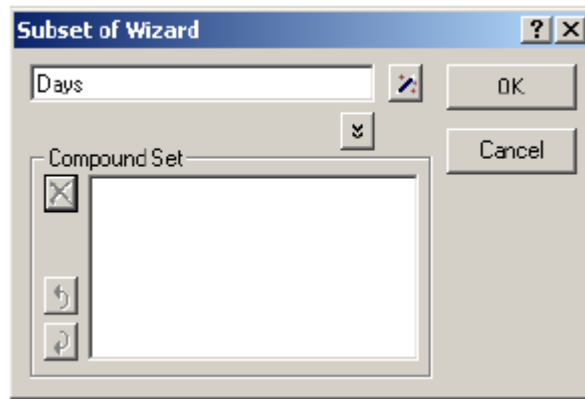


图 6.18: 子集向导

子集的名称含义是不言而喻的，创建连接范围 ‘periods’ 和日历 ‘days’ 的时间表后，子集 ‘Mondays’ 就能发挥重要作用，子集 ‘Mondays’ 常被定义为一个函数参数，在 AIMMS 中一旦模型时间窗口是星期一，意味着范围 ‘Periods’ 进入一个新的阶段。请在日历声明段落输入如下 5 个子集：

SET:

identifier : WeekendDays

subset of : Days

definition : { d | TimeslotCharacteristic(d, 'weekday') > 5 }

SET:

identifier : OfficialHolidays

subset of : Days

SET:

identifier : InactiveDays

subset of : Days

definition : WeekendDays + OfficialHolidays

SET:

identifier : Mondays

subset of : Days

definition : { d | TimeslotCharacteristic(d, 'weekday') = 1 }

SET:

identifier : DaysInPeriod

index domain : t

subset of : Days

Timeslot

预定义函数 ‘TimeslotCharacteristic’ 决定了时隙的数值，用以取代 ‘weeks’ 中 ‘days’ 表示。在集合 ‘WeekendDays’ 定义属性栏中，选取周中天数大于 ‘5’（因为周末的存在）元素。同样地，在集合 ‘Mondays’ 定义属性栏中，函数将选择所有星期一（数值等于 1）作为区间分割天。

此时，因为日历的开头和结尾已被定义为字符串常数，整个日历即被定义。同样地，子集 ‘WeekendDays’ 也全部被定义，请按以下步骤查看其内容：

- 模型中选择集合 ‘WeekendDays’，然后
- 鼠标右击弹出对话框中选择 ‘Data’ 命令（如图 6.19 所示）

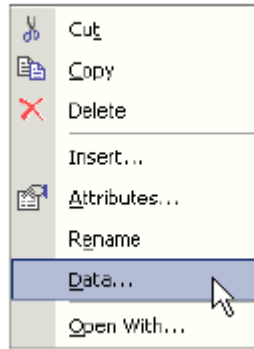


图 6.19: 鼠标右击弹出对话框

AIMMS 中，相应数据页面如图 6.20 所示，界面左边显示的是集合 ‘WeekendDays’ 中元素，界面右边是子集 ‘Days’ 中元素。

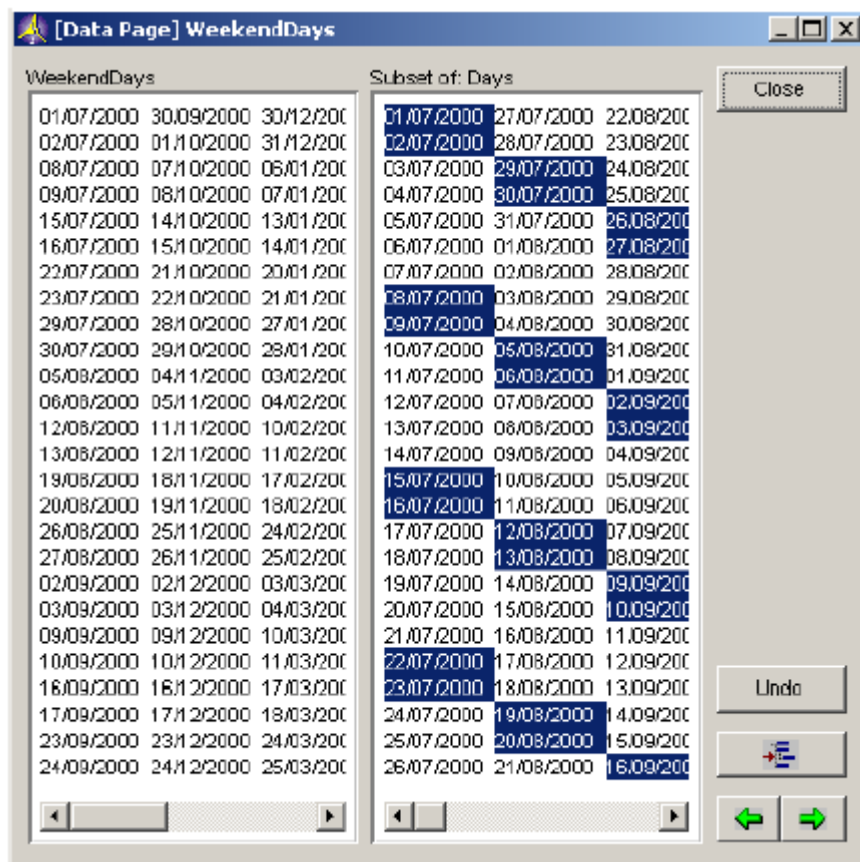


图 6.20: 集合 WeekendDays 的数据界面

除日历之外，也可以创建周历以及相应的子集。基于当前所学内容，您可以声明周历，教程建议您手动指定 ‘Timeslot format’ 属性，因为周历的格式并不是预定义的，完成后的 ‘Weeks’ 属性窗口如图 6.21 所示：

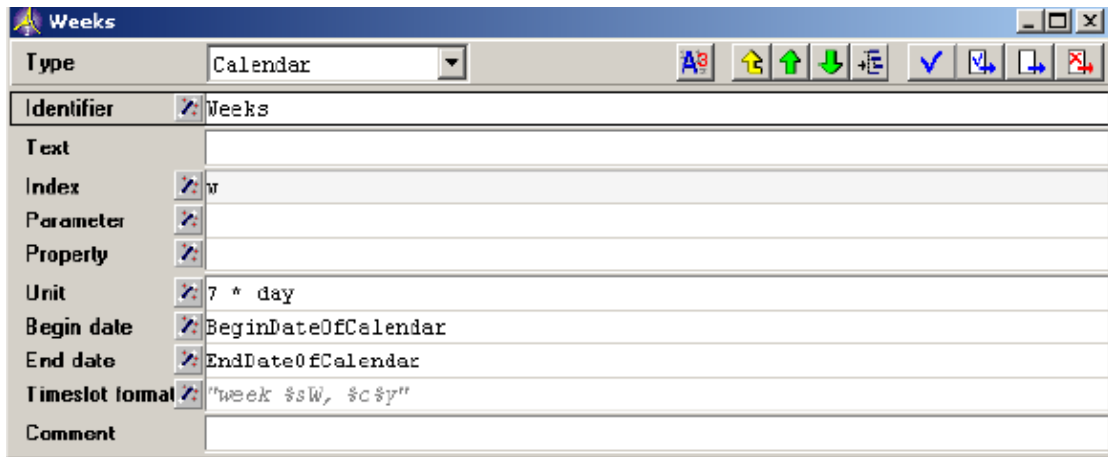


图 6.21: ‘Weeks’ 的最终属性窗口

此时访问 ‘Weeks’ 数据，将会弹出警告说明，警告内容显示：‘对周历而言，起开始时间并非一周的第一天（周一），从而限制周历的使用’。然而教程项目中，这些限制不会导致任何问题，为了防止警告窗口再次弹出，请按以下步骤关闭选项 ‘Warning calendar week begin’：

- 打开 ‘Setting’ 菜单，执行 ‘Project Options’ 命令
- 在选项树中选择文件夹 ‘AIMMS - Progress, errors & warnings - Warnings - Compilation’（如图 6.22 所示）
- 在最右边窗口点击 ‘Warning calendar week begin’ 选项
- 选择 ‘Off’ 项
- 点击 ‘AIMMS Options’ 对话框中的 ‘Apply’ 按钮
- 按 ‘OK’ 按钮完成

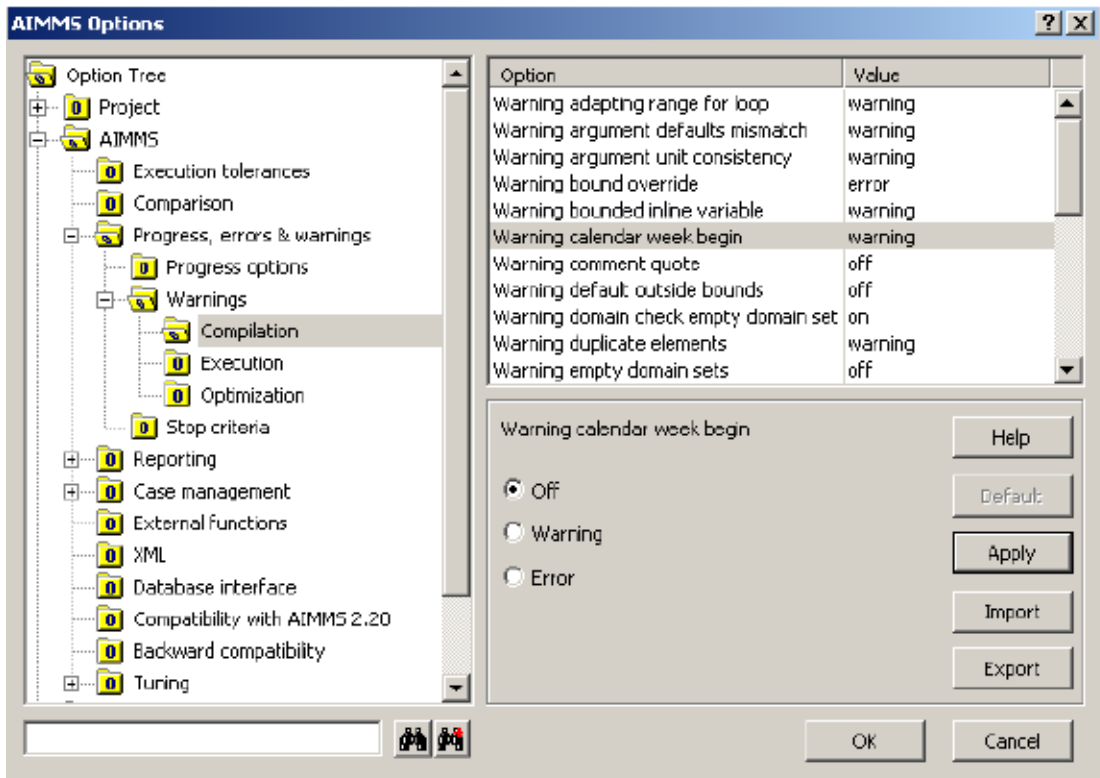


图 6.22: ‘AIMMS Options’ 对话框

如前文所述，输入下面这些与‘Weeks’相关的子集和元素参数

```
SET:  
  identifier : InactiveWeeks  
  subset of : Weeks  
ELEMENT PARAMETER:  
  identifier : WeekInPeriod  
  index domain : t  
  range : Weeks
```

如果了解天数和周数之间关系，创建一个包含天数和周数且带索引的元素参数，对于日历中每一天，周历中都有某一周与之对应。请输入以下声明：

```
ELEMENT PARAMETER:  
  identifier : DayToWeek  
  index domain : d  
  range : Weeks  
  definition : first( w | TimeslotCharacteristic(w, ' week' ) =  
                    TimeslotCharacteristic(d, ' week' )  
                    and  
                    TimeslotCharacteristic(w, ' year' ) =  
                    TimeslotCharacteristic(d, ' year' ) )
```

通过使用函数‘TimeslotCharacteristic’，非常直观的核实第‘w’周（其范围从第1周到第5周）周数是否等于第‘d’天所在周数，年数可以同样方式核实。

下面日历相关的标识符后面需要用到，现在输入：

```
ELEMENT PARAMETER:  
  identifier : LastWeekInCalendar  
  range : Weeks  
  definition : last(Weeks)
```

模型树中包含与日历相关的部分如图 6.23 所示：

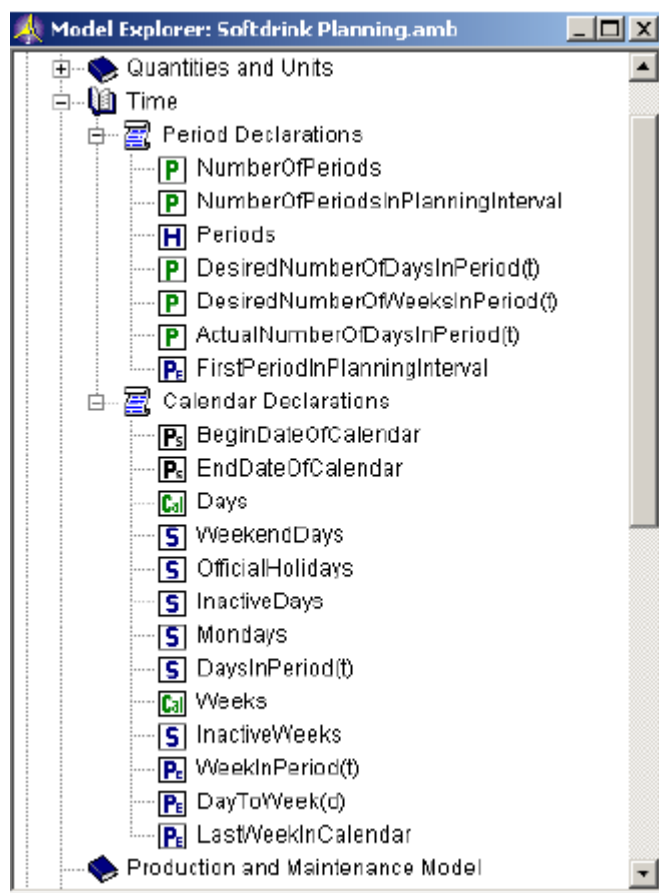


图 6.23：模型树中全部日历相关声明

第七章

生产和维护模型

在这一章,我们将输入与第三章描述的数学模型相关的全部标识符。对于大部分现实模型,标识符的数量相当庞大,可以通过声明额外的标识符段落改进模型树。

7.1 模型结构

首先在段落‘Production and Maintenance Model’中输入下面子段落:

- Location Declarations
- Scenario Declarations
- Production Declarations
- Supply and Demand Declarations
- Maintenance and Vacation Declarations
- Cost Declarations
- Mathematical Program Declarations

完成后的模型树段落如图 7.1 所示:

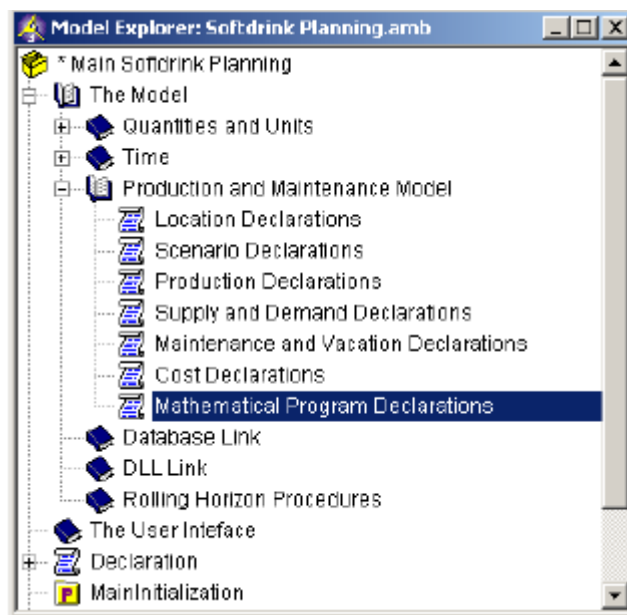




图 7.1: 模型树中添加的 7 个子声明段

本章大部分声明都提供了简洁的文本格式,他们与前面章节的属性格式密切相关。基于当前所学内容,完成其属性形式取代详细的声明说明。

和第四章一样,在模型树的‘Production and Maintenance Model’中导入模型段‘Production and Maintenance Model.amb’,可避免输入本章中声明的部分或者全部。如果选择导入文件选项,建议首先检查本章剩余部分所列出的声明。

7.2 网络拓扑

第五章中，您已声明位置集合以及相应的 X 坐标和 Y 坐标。现在移动这些标识符到模型树中新的位置，具体操作如下：

- 在模型树的结尾创建新的声明段 ‘Declaration’
- 按住 shift 键，点击鼠标左键并移动鼠标选择全部三个标识符
- 点击工具栏中的剪切按钮 
- 选择并打开 ‘Location Declarations’ 段落，然后
- 点击工具栏中的复制按钮 

此外，您也可以直接拖动三个标识符到新的位置。

在 ‘Location Declarations’ 段中，声明集合 ‘Factories and Centers’ 作为位置集合的子集。

SET:

```
identifier : Factories
subset of  : Locations
index     : f
```

SET:

```
identifier : Centers
subset of  : Locations
index     : c
```

7.3 需求情况

在段落 ‘Scenario Declarations’ 中增加下面两个标识符，他们用来设置需求情况。注意当某集合有固定数量的已知元素时，请在 ‘Definition’ 属性框中输入这些元素作为数据（请看集合 ‘Scenarios’ ）

SET:

```
identifier : Scenarios
index     : s
definition : data { pessimistic, expected, optimistic }
```

PARAMETER:

```
identifier : Probability
index domain : s
definition : data { pessimistic : 0.25, expected : 0.50, optimistic : 0.25 }
```

7.4 生产

AIMMS 的一个有趣功能是可以指定一个全局索引的定义域条件，如下面最后三个声明所示。在本例中，AIMMS 在应用中仅考虑 (f, p) 组合，注意到运算符号 ‘|’ 用来表示 ‘such that’，运算符 Ord(p) 则反馈元素 p 在其定义集合 ‘ProductionLines’ 的原始位置。

打开子段落 ‘Production Declarations’，阅读前文，然后输入下面声明：

```

SET:
  identifier : ProductionLines
  index     : p
  definition : data { line-01 .. line-04 }

PARAMETER:
  identifier : NumberOfProductionLines
  index domain : f

SET:
  identifier : FactoryProductionLines
  index domain : f
  subset of : ProductionLines
  definition : { p | Ord(p) <= NumberOfProductionLines(f) }

PARAMETER:
  identifier : DeteriorationLevel
  index domain : (f,p) | p in FactoryProductionLines(f)

PARAMETER:
  identifier : DeteriorationLevelAtStartOfCalendar
  index domain : (f,p) | p in FactoryProductionLines(f)

PARAMETER:
  identifier : MaximumDeteriorationLevel
  index domain : (f,p) | p in FactoryProductionLines(f)

```



‘Unit’ 向导仅能从期望单位和已输入单位表达式中选择一个作为 ‘Unit’ 属性。因此声明下面两个参数中首个时，手动输入 ‘Unit’ 属性 [h1/day]，当输入第二个声明时，您可以使用 ‘Unit’ 向导功能，然后选择 ‘Implicit Quantities’ 。

```

PARAMETER:
  identifier : ProductionLineLevelAtStartOfCalendar
  index domain : (f,p) | p in FactoryProductionLines(f)
  unit       : h1/day

PARAMETER:
  identifier : MaximumProductionLineLevel
  index domain : (f,p) | p in FactoryProductionLines(f)
  unit       : h1/day

```

完成下面列表声明输入后，AIMMS 仍不能编译参数 ‘PotentialProduction’ 定义，该定义引用了三个尚未声明的标识符 LineInMaintenance，DropDueToVacation 和 IsVacationPeriod，此时使用 ‘Commit and close’ 按钮  代替 ‘Check, commit and close’ 按钮 ，AIMMS 就不会出现错误警告，而指令 AIMMS 编译模型树则会提示错误—缺失标识符，后面讲声明者三个标识符。

```

PARAMETER
  identifier : PotentialProduction
  index domain : (f,p,t) | p in FactoryProductionLines(f)
  unit       : h1
  definition : ActualNumberOfDaysInPeriod(t) *
              ( 1 - LineInMaintenance(f,p,t) ) *
              ( 1 - DropDueToVacation * IsVacationPeriod(f,t) ) *
              MaximumProductionLineLevel(f,p)

```

到目前为止，输入的全部声明如图 7.2 所示：

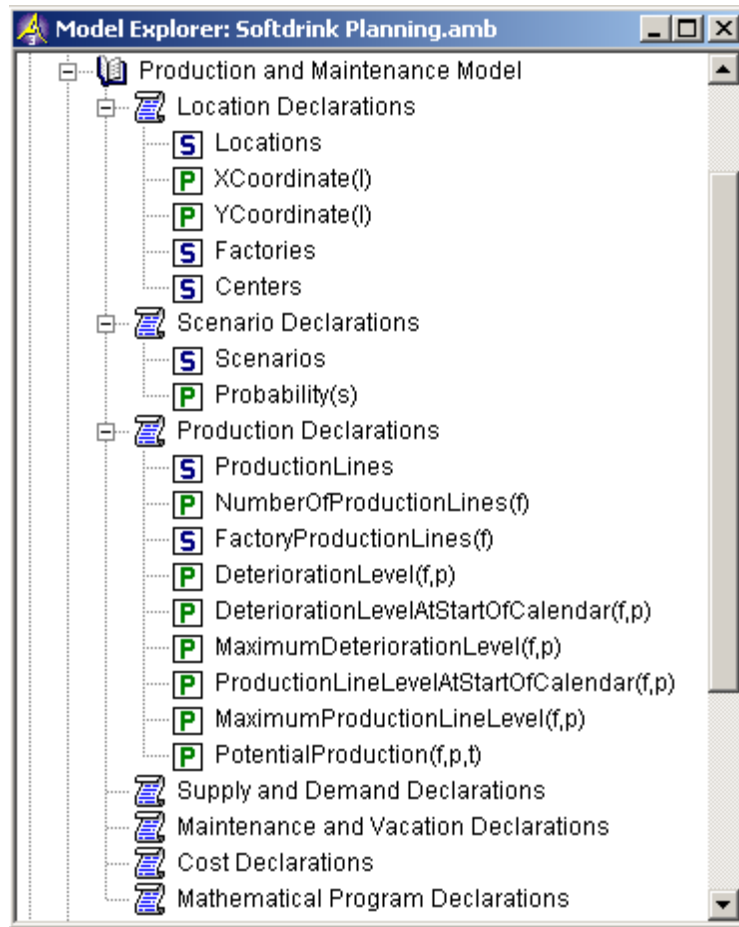


图 7.2：位置，情况和生产声明

7.5 供应和需求

使用 ‘Unit’ 向导，‘Range’ 向导和名称完成功能，在模型树的合适位置输入以下需求和供应声明：

```
PARAMETER:  
  identifier : Demand  
  index domain : (c,t,s)  
  unit : hl
```

```
PARAMETER:  
  identifier : MinimumStock  
  index domain : 1  
  unit : hl
```

```
PARAMETER:  
  identifier : MaximumStock  
  index domain : 1  
  unit : hl
```

```
PARAMETER:
  identifier : StockAtStartOfCalendar
  index domain : 1
  unit : h1
```

```
PARAMETER:
  identifier : MaximumTransportCapacity
  index domain : f
  unit : TL
```

7.6 维修和假期

如第二章所说，关于维修计划的大部分计算指令可以放在数学规划外部。在此只需对生产线处于维修的状态的情况进行声明。请在模型树中使用‘Maintenance and Vacation Declarations’声明段。

```
PARAMETER:
  identifier : LineInMaintenance
  index domain : (f,p,t)
  range : binary
```

从工厂的管理历史可以知道，在某几周，工厂大部分员工处于休假状态。在这期间，工厂产能降为最大产能的 40%，请声明下面两个参数：

```
SET:
  identifier : VacationWeeks
  index domain : f
  subset of : Weeks

PARAMETER:
  identifier : DropDueToVacation
  initial data : 0.4

PARAMETER:
  identifier : IsVacationPeriod
  index domain : (f,t)
  range : binary
  definition : if ( WeekInPeriod(t) in VacationWeeks(f) )
               then 1
               else 0
               endif
```

此时，编译整个模型，因为 7.4 节缺失的三个标识符此时都已被声明。执行‘Run’菜单中的‘Compile ALL’命令，完成模型编译，也可以按 F5 键完成模型编译。最后请忽视数据初始化的警告。

7.7 成本

每种情况总成本分为四个部分，每部分都包含单位成本声明，全部成本等于所有情况的总成本加权，请在模型树 ‘Cost Declarations’ 段落输入下面声明：

```
PARAMETER:  
  identifier : UnitTransportCost  
  index domain : (f,c)  
  unit : $/TL  
  
PARAMETER:  
  identifier : FixedCostDueToLevelChange  
  unit : $  
  initial data : 25000  
  
PARAMETER:  
  identifier : UnitStockCost  
  index domain : 1  
  unit : $/h1  
  
PARAMETER:  
  identifier : UnitProductionCost  
  index domain : f  
  unit : $/h1
```

全部供应和需求、维修和假期、成本的声明如图 7.3 所示：

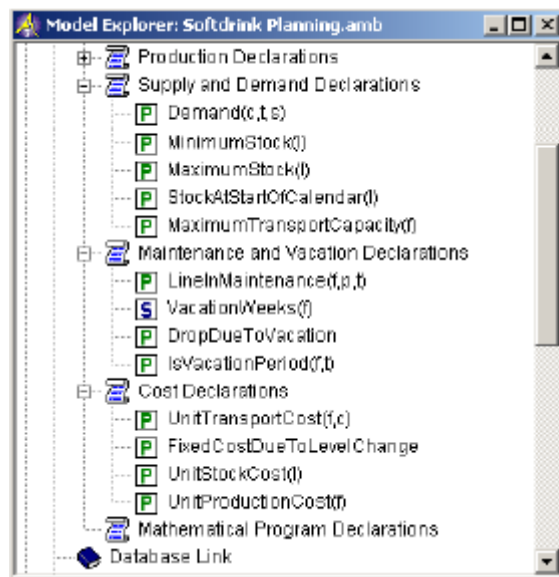


图 7.3 全部供应和需求，维修和假期，成本声明

7.8 优化模型

本教程中，数学规划的模型构造需要下面 7 个变量，请在模型树 ‘Mathematical Program Declarations’ 声明段中输入他们。

```
VARIABLE:
  identifier : ProductionLineInUse
  index domain : (f,p,t) | p in FactoryProductionLines(f)
  range       : binary

VARIABLE:
  identifier : Production
  index domain : (f,t)
  range      : nonnegative
  unit      : h1
  definition : sum[ p, PotentialProduction(f,p,t) *
                  ProductionLineInUse(f,p,t) ]

VARIABLE:
  identifier : ProductionLineLevelChange
  index domain : (f,p,t) | p in FactoryProductionLines(f)
  range      : [0, 1]

VARIABLE:
  identifier : Transport
  index domain : (f,c,t,s) | UnitTransportCost(f,c)
  range      : [0, MaximumStock(c)]
  unit      : TL

VARIABLE:
  identifier : Stock
  index domain : (l,t,s)

  range      : [MinimumStock(l), MaximumStock(l)]
  unit      : h1
  definition : Stock(l,t-1,s) + Production(l,t) +
              sum[ f, Transport(f,l,t,s) ] -
              sum[ c, Transport(l,c,t,s) ] - Demand(l,t,s)

VARIABLE:
  identifier : TotalScenarioCost
  index domain : s
  unit      : $
  definition : sum[ (f,p,t), FixedCostDueToLevelChange *
                  ProductionLineLevelChange(f,p,t) ]
              sum[ (f,t), UnitProductionCost(f) *
                  Production(f,t) ]
              sum[ (l,t), UnitStockCost(l) *
                  Stock(l,t,s) ]
              sum[ (f,c,t), UnitTransportCost(f,c) *
                  Transport(f,c,t,s) ]

VARIABLE:
  identifier : TotalCost
  unit      : $
  definition : sum[ s, Probability(s) * TotalScenarioCost(s) ]
```

注意其中四个变量本身已有定义，运行相应的数学规划模型时，AIMMS 会把定义作为约束条件。

构造数学规划模型余下的三个变量约束如下所示：

```
CONSTRAINT:
  identifier : ChangeWhenIncrease
  index domain : (f,p,t) | p in FactoryProductionLines(f)
  definition : ProductionLineLevelChange(f,p,t) >=
              ProductionLineInUse(f,p,t) - ProductionLineInUse(f,p,t-1)

CONSTRAINT:
  identifier : ChangeWhenDecrease
  index domain : (f,p,t) | p in FactoryProductionLines(f)
  definition : ProductionLineLevelChange(f,p,t) >=
              ProductionLineInUse(f,p,t-1) - ProductionLineInUse(f,p,t)

CONSTRAINT:
  identifier : RestrictTransportCapacity
  index domain : (f,t,s)
  unit : TL
  definition : sum[ c, Transport(f,c,t,s) ] <= MaximumTransportCapacity(f)
```

考虑模型运行的需要，与目标、优化方向和模型类型相应的变量与约束集合都需要在 AIMMS 数学规划中被指定。假如不指定变量集或约束集，AIMMS 将假设全部模型变量和模型约束都已包含在数学规划中，如图 7.4 所示，请使用‘Objective’、‘Direction’和‘Type’向导对数学规划‘LeastCostPlan’进行声明。

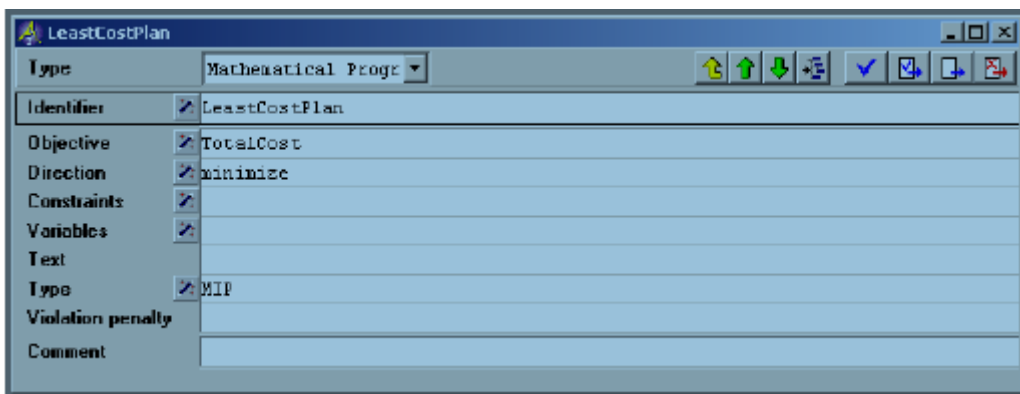


图 7.4：数学规划的属性栏

数学规划声明段中的全部变量与约束如图 7.5 所示：

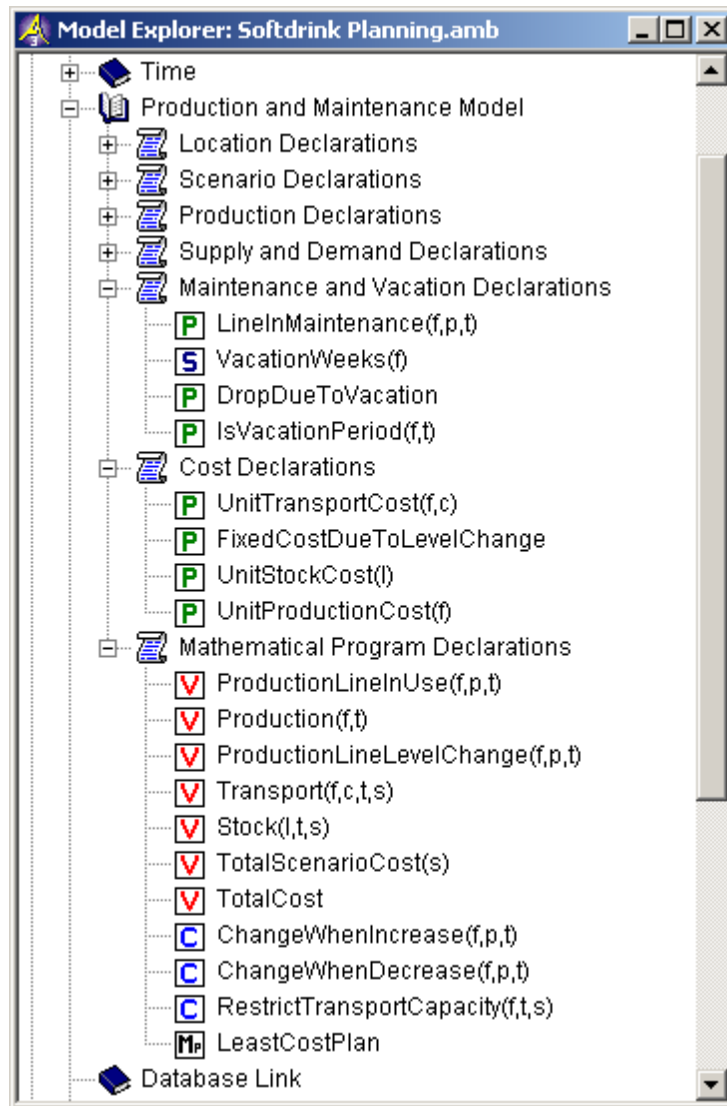


图 7.5: 模型树

第三部分

模型程序和函数

第8章

数据库链接

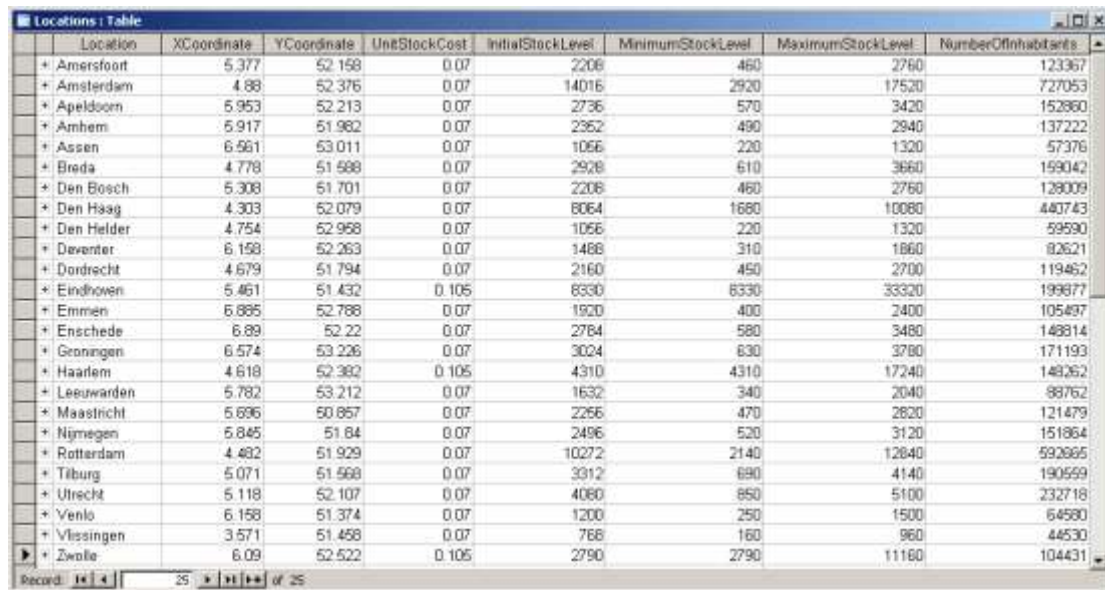
从本章中您将了解到，使用‘点—击数据框交互’功能直接连接模型和数据库，此外还将阐述输入SQL子程序的可能性。

按照本章中步骤实践之后，如果认为自己需要了解更多有关数据库链接知识，请参看《语言参考手册》中章节“数据库间的对话”。

8.1 数据库表

AIMMS与数据库间的连接依托于ODBC（开放式数据库连接）标准，或OLE DB标准。任一商业数据包至少支持其中一种标准，本教程所使用的MS Access数据库程序包也不例外。在这里，我们将基于ODBC标准连接AIMMS和MS Access。

数据库基本模块是由行列构成的数据库表格，在每一个数据库表格中，一列或多列被设为主键列，并按照其定义生成剩余列中数据，通过每一行主键值识别改行。图8.1表格中第一列为主键列，然后通过位置名称识别每一行。



Location	XCoordinate	YCoordinate	UnitStockCost	InitialStockLevel	MinimumStockLevel	MaximumStockLevel	NumberOfInhabitants
Amersfoort	5.377	52.158	0.07	2206	460	2760	123367
Amsterdam	4.88	52.376	0.07	14016	2920	17520	727053
Apeldoorn	5.953	52.213	0.07	2736	570	3420	152860
Arnhem	5.917	51.982	0.07	2952	490	2940	137222
Assen	6.561	53.011	0.07	1056	220	1320	57376
Breda	4.778	51.988	0.07	2928	610	3660	159042
Den Bosch	5.308	51.701	0.07	2206	460	2760	128009
Den Haag	4.303	52.079	0.07	8064	1680	10080	440743
Den Helder	4.754	52.958	0.07	1056	220	1320	59590
Deventer	6.158	52.263	0.07	1488	310	1860	82621
Dordrecht	4.679	51.794	0.07	2160	450	2700	119462
Eindhoven	5.461	51.432	0.105	8330	8330	33320	199677
Emmen	6.885	52.788	0.07	1920	400	2400	105497
Enschede	6.89	52.22	0.07	2784	580	3480	148914
Geoningen	6.574	53.226	0.07	3024	630	3780	171193
Haarlem	4.618	52.382	0.105	4310	4310	17240	148262
Leeuwarden	5.782	53.212	0.07	1632	340	2040	88762
Maastricht	5.696	50.857	0.07	2256	470	2820	121479
Nijmegen	5.845	51.84	0.07	2496	520	3120	151864
Rotterdam	4.482	51.929	0.07	10272	2140	12840	592665
Tilburg	5.071	51.968	0.07	3312	690	4140	190559
Utrecht	5.118	52.107	0.07	4080	850	5100	232718
Venlo	6.158	51.374	0.07	1200	250	1500	64580
Vlissingen	3.571	51.458	0.07	768	160	960	44530
Zwolle	6.09	52.522	0.105	2790	2790	11160	104431



表8.1: 表格“Locations”内容

本教程中数据库包含四个数据库表格：第一张表格数据应用于工厂和配送中心（如调配和库存水平）；第二张提供工厂配置所需的相关数据（如生产能力和成本）；第三张是历史数据表（随时间变化的需求量），用于启动周期滚动过程；最后一张表格包含生产线配置的相关数据（如生产线产能）。

8.1.1 输入首张数据库表声明

你可以使用‘database table’标识符声明访问AIMMS的外部数据库，指定你想进入的数据‘ODBC数据源’名称，或者你想读写的外部数据库表格名称。

首先声明AIMMS首张数据表，请按下面步骤执行：

- 在模型树‘Database Link’段中创建新的声明段，命名‘Database Declarations’
- 打开该声明段
- 点击工具栏‘Other. . .’按钮
- ‘Select Type of Identifier’对话框中选择数据库表格按钮，创建新的数据库表‘identifier’
- 命名‘LocationTable’

本教程提供‘Softdrink Factory Planning.mdb’的MS Access数据库文件，接下来执行下面操作将其链接到AIMMS。

- 激活数据库表‘LocationTable’属性列表中‘Data source’向导
- 在弹出菜单中选择‘Select File Data Source. . .’命令
- 从‘Data’子目录中选择文件‘Softdrink Planning.dsn’
- 按‘Save’按钮保存。

值得注意的是，您必须使用适合您AIMMS结构的Microsoft Access驱动程序。如果您使用32位版本的AIMMS，您必须在系统上安装32位的ODBC驱动程序。同样地，64位版本的AIMMS应该对应于64位ODBC驱动程序。其中有一种复杂情形：如果您想要安装任一版本的ODBC驱动程序，那么必须确保您已经安装了相应的Microsoft Office；更为准确地说，若您没有安装相应结构的Microsoft Office组件，那么驱动程序是不可能成功运行的。例如，系统上只要安装了至少一个32位的Microsoft Office组件，即不可能安装64位ODBC驱动程序。

创建数据源后，然后从中选择表格，请执行以下简单步骤：

- 激活‘Table name’向导，
- 从弹出菜单中选中‘Select Table/Query Name. . .’命令，
- 选择‘Locations’
- 按下OK按钮

如果您之前没有使用过外部数据库，那么您可能会对将外部数据库表导入数据库中比较感兴趣。为此，启动MS Access，然后检查数据库表“Locations”设计视图，如表8.2。

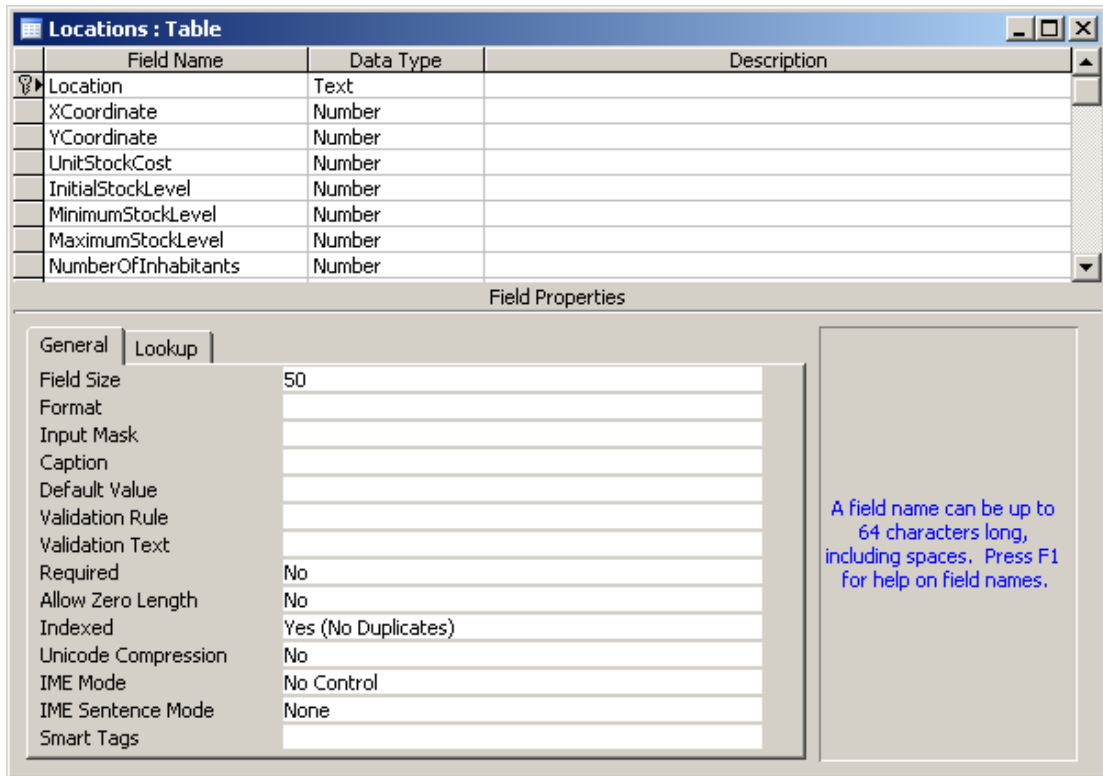




表8.2: 在表 ‘Locations’ 的MS Access设计视图

一般而言，外部数据库表与AIMMS中 ‘identifiers’ 对应命名习惯并不相同，因此，有必要采用映射将外部数据库表栏中项目与AIMMS中的 ‘identifiers’ 对应起来。例如，可对AIMMS中identifier1索引与数据 ‘Location’ 栏指定如下映射：

- 激活 ‘Mapping’ 向导
- 从 ‘Data Column’ 下拉列表中选择列 ‘Location’（参见表8.3）
- 按下 ‘Wizard’ 按钮 ，选择索引 ‘1’ 作为为 ‘AIMMS Identifier’
- 按下 ‘Transfer’ 按钮 ，将指定映射关系填入 ‘Mappings’ 列表中
- 按下OK按钮。

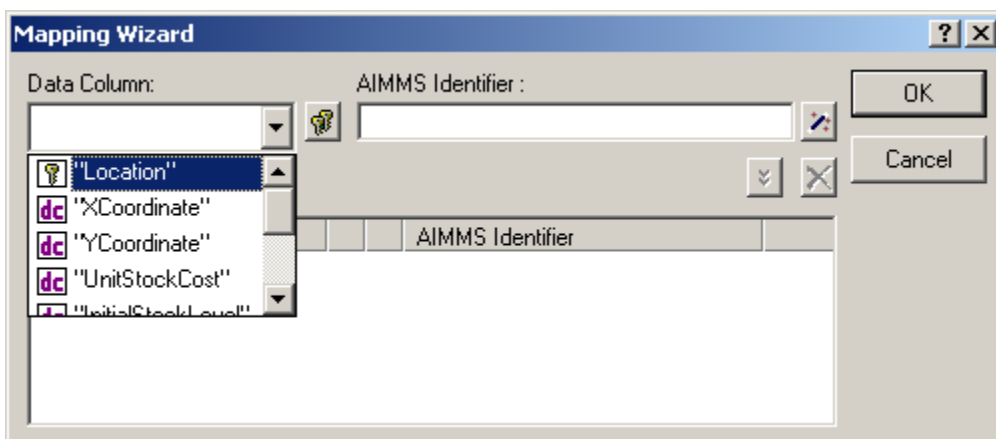


图8.3: ‘Mapping’ 向导

参照图8.4，通过前文所述向导完成映射关系的建立。

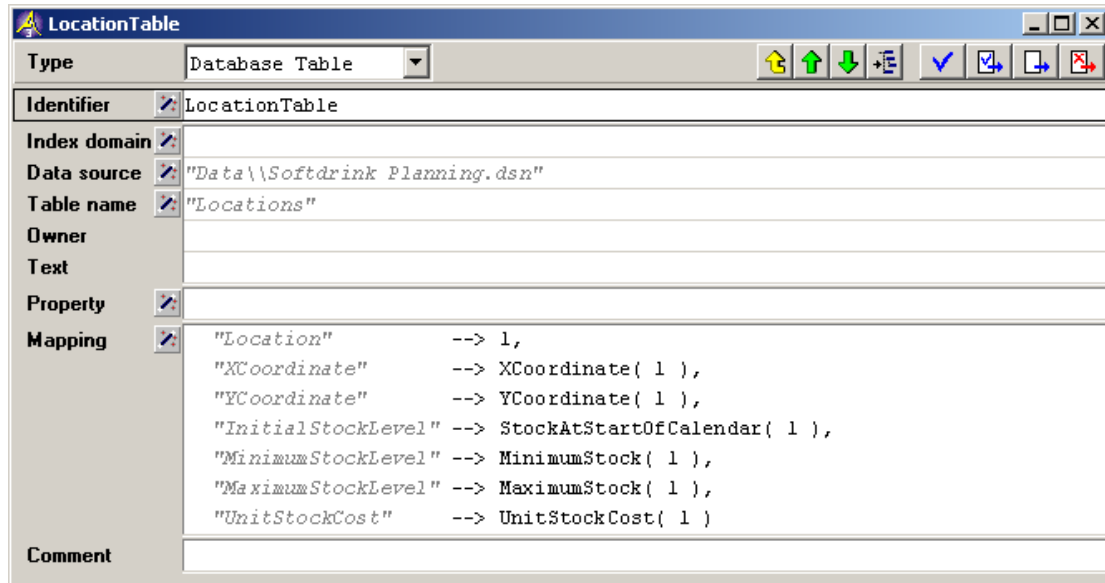


图8.4: 数据表“Locations”属性形式

8.1.2 输入附加数据库表声明

按照上节所述完成了第一个数据库表声明后,然后完成剩余外部数据库表使其同样适用于AIMMS。输入相应声明之前,需要补充两个模型参数以方便存储从数据库读取每周的需求数据。

PARAMETER:

```

  identifier   : WeeklyDemand
  index domain : (c,w,s)
  unit        : hl

```

PARAMETER:

```

  identifier   : TotalWeeklyDemand
  index domain : (w,s)
  unit        : hl

```

首先对模型树中补充的三个数据库表格标识符: ‘FactoryTable’、‘CenterTable’ 和 ‘ProductionLineTable’ (位于参数 ‘TotalWeeklyDemand’ 下一级) 进行声明。然后参考下表列出的属性描述,使用Data source向导,Table name向导和Mapping向导分别填写相应的属性形式。

DATABASE TABLE:
 identifier : FactoryTable
 data source : "Data\\Softdrink Planning.dsn"
 table name : "Factories"
 mapping : "Factory" |--> f,
 "UnitProductionCost" --> UnitProductionCost(f),
 "MaximumTransportCapacity" --> MaximumTransportCapacity(f)

DATABASE TABLE:
 identifier : CenterTable
 data source : "Data\\Softdrink Planning.dsn"
 table name : "Centers"
 mapping : "Center" --> c,
 "Date" --> w,
 "Scenario" --> s,
 "Demand" --> WeeklyDemand(c, w, s)

DATABASE TABLE:
 identifier : ProductionLineTable
 data source : "Data\\Softdrink Planning.dsn"
 table name : "ProductionLines"
 mapping :
 "Factory" --> f,
 "ProductionLine" --> p,
 "InitialUsageCount" --> DeteriorationLevelAtStartOfCalendar(f, p),
 "InitialProductionLevel" --> ProductionLineLevelAtStartOfCalendar(f, p),
 "MaximumProductionLevel" --> MaximumProductionLineLevel(f, p),
 "MaximumUsageCount" --> MaximumDeteriorationLevel(f, p)

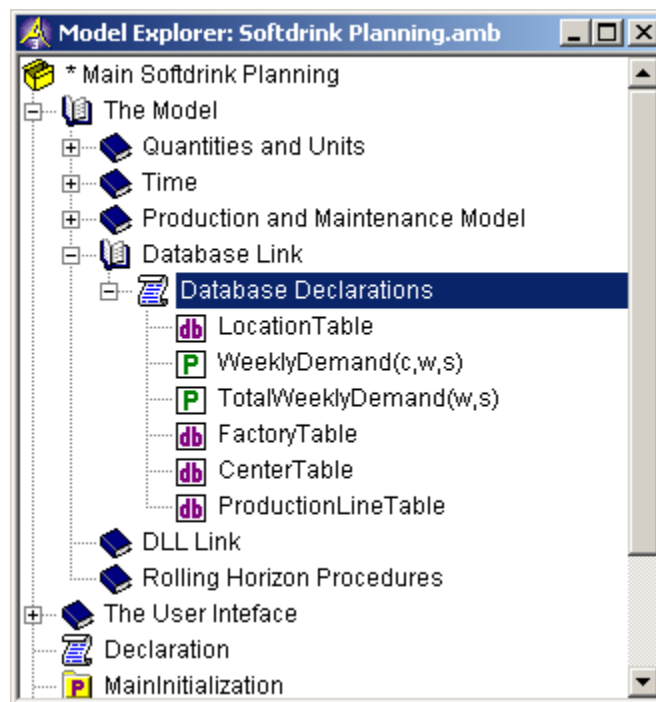


图8.5: 此时模型树图中数据库段落

8.2 数据库程序

相比标准的数据库表接口中的数据连接，当您从数据库表中传送数据，或者传送数据到另一数据库表中时，您可能需要更精密的控制，而在AIMMS中，可通过编写并执行SQL（结构化查询语言）语句，或提供调用数据库中已有程序的访问路径，实现对于数据连接的控制。




8.2.1 SQL查询

访问尚未存储在数据库表中的数据，例如，数据库表ProductionLines，其两个主键列分别为“Factory”和“ProductionLine”，显然在此数据库表中，没有对应于每条生产线数量条目。然而该信息却使用SQL从数据库中获得。

```
ELECT Factory, COUNT(ProductionLine) AS LineCount  
FROM ProductionLines GROUP BY Factory
```

暂时查询会在数据库临时创建了一张新表，该表包含两个主键列，其一为“Factory”，另一个为“LineCount”，从而实现前面要求。

为了在AIMMS中实现本次查询，创建名为“NumberOfProductionLinesQuery”的数据库程序。请执行以下步骤：

- 通过双击滚动图标关闭“Database Declarations”的声明部分
- 按下工具栏中“Other. . .”按钮
- 从“Select Type of Node”（选择节点类型）对话框中选择数据库程序（见图8.6），然后按OK按钮
- 输入该数据库程序名称“NumberOfProductionLinesQuery”
- 按下回车键完成命名注册

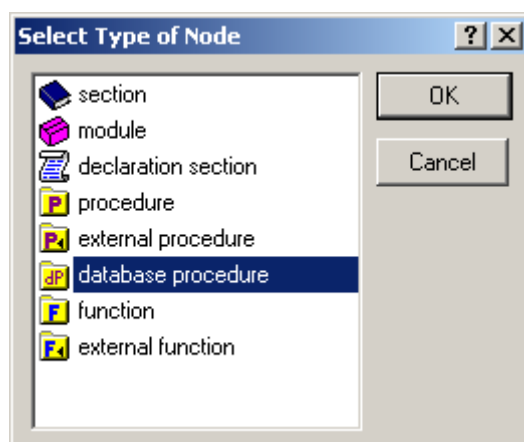


图8.6: “Select Type of Node”（选择节点类型）对话框

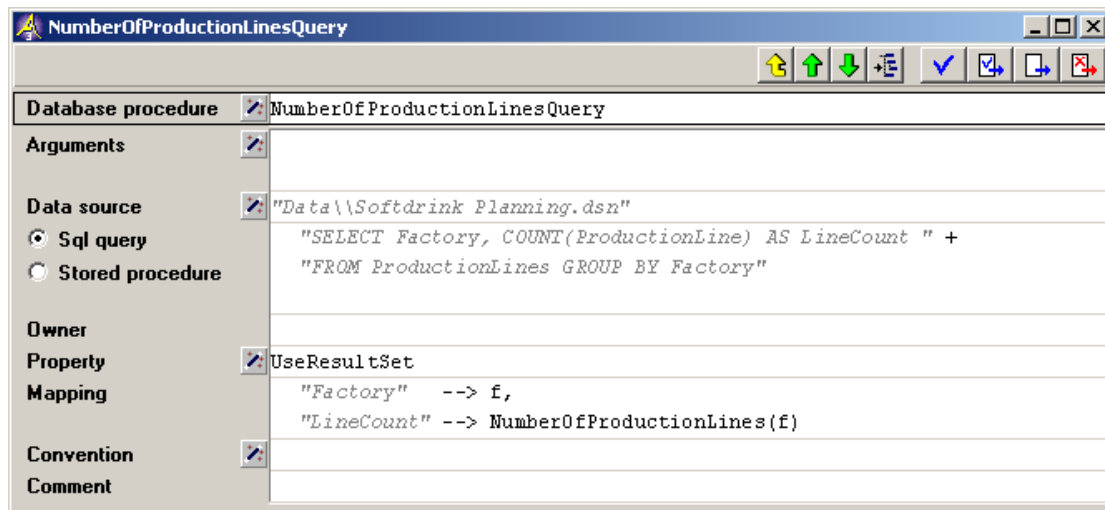



图8.7：数据库程序执行SQL命令

8.2.2 存储程序

在上个小节中，把你创建的 SQL 查询语句放在一个 AIMMS 数据库程序中。在这一小节，考虑已经存在数据库的查询语句，而且你还将可以进入 AIMMS 数据库程序。

一个储存的程序可以有一个或多个变量，而且设置 AIMMS 数据库程序的这些变量很直接。但是，在本教程，我们使用命名为 TotalDemand 和 AllCenters 的储存程序，而且这些程序刚好没有变量。

要声明你的第二个数据库程序，请执行以下操作：

- 在模型树里插入一个新的数据库程序，并指定 'TotalDemandQuery' 作为它的名字
- 打开它的属性形式
- 使用 Data source 向导，选择 'Softdrink Planning.dsn' 作为它的 Data Source 属性
- 按 Stored procedure 前面的单选按钮
- 启动 Stored procedure 向导
- 在弹出的菜单中选择 Select Stored Procedure Name... 指令
- 选择 'TotalDemand' 作为 Stored procedure 属性
- 完成如图 8.8 所示的属性形式，然后
- 使用 Select Stored Procedure Name... 按钮 

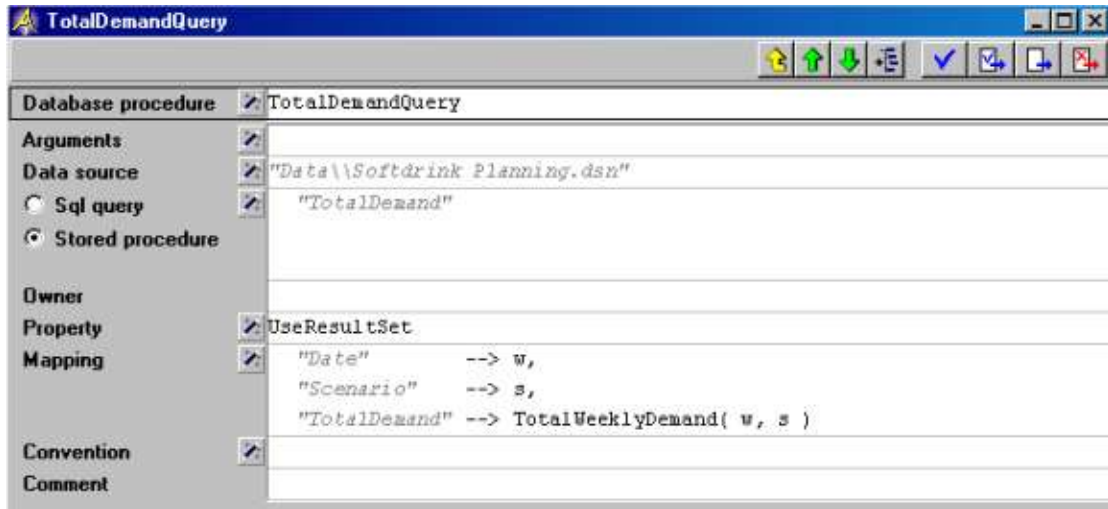


图 8.8: 数据库程序 TotalDemandQuery 的完整属性形式

声明命名为 ‘AllCentersQuery’ 的第三个数据库程序，请执行上述提及的类似步骤。只是这一次选择 ‘AllCenters’ 作为 **Stored procedure** 属性。完整的属性形式应该如图 8.9 所示。

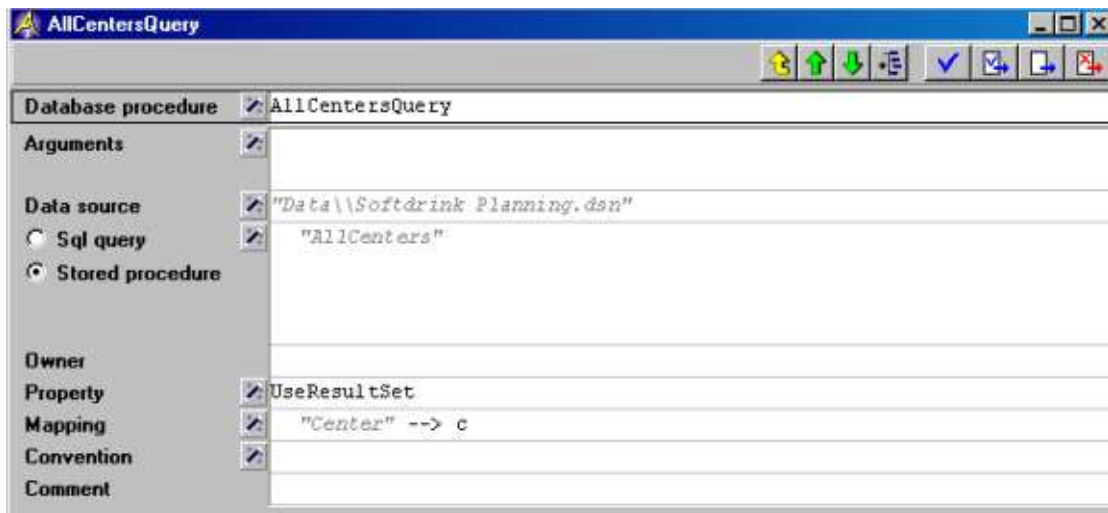


图 8.9: 数据库程序 AllCentersQuery 的完整属性形式

描述数据库链接的模型树部分如图 8.10 所示。

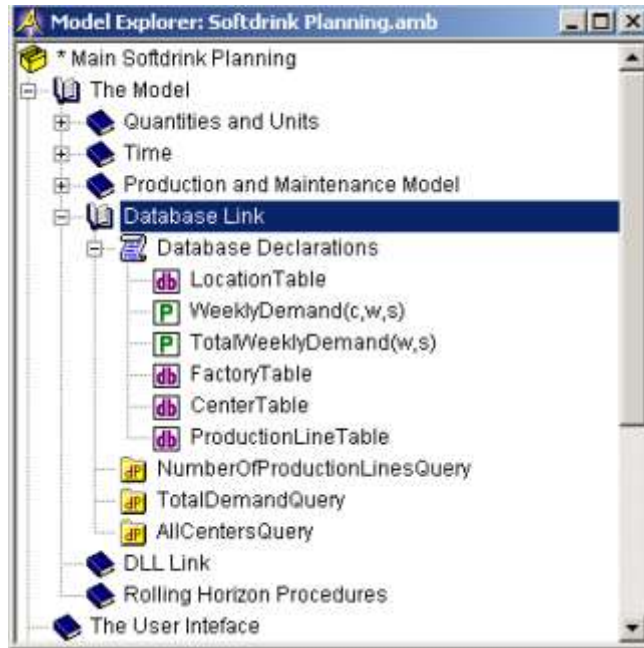


图 8.10: 显示所有数据库识别符和程序的中间模型树

第九章

函数与程序

在上一章，你学习了数据库程序。在这一章，你将开发几个 AIMMS 程序以读取数据和控制整个周期滚动过程。而且，你还将处理从 AIMMS 里面调用的外部函数。

9.1 读取数据库

在 AIMMS 中同时读取所有的数据库表格是很容易的。比如，考虑上一章声明的数据库表格 LocationTable。输入以下语句：



```
read from table LocationTable;
```

是让 AIMMS 读取所有相应的数据库表格中设置的 Mapping 属性的标识符。我们也可选择要读取的数据库表格中设置的 Mapping 属性的标识符。例如，以下的语句

```
read XCoordinate, YCoordinate from table LocationTable;
```

只读取 XCoordinate 和 YCoordinate 的数据。

现在你需要创建一个名为 ReadFromDatabase 的程序，并用以下方式把它放置在模型树的 Database Declarations 节点和 NumberOfProductionLinesQuery 节点中间。

- 选择模型树的 Database Declarations 部分
- 如果打开的话，通过点击图标前面的负号标志  关掉这一节
- 按工具栏的 New Procedure 按钮 
- 输入 ‘ReadFromDatabase’ 作为程序名称，然后
- 按 Enter 键，注册这个名字

双击程序 ReadFromDatabase 名，打开属性形式，如图 9.1 完成 Body 属性。注意，两个数据库程序 NumberOfProductionLinesQuery 和 TotalDemandQuery 都生成数据库的临时表格，而且对 AIMMS 而言，每个程序的名字似乎和临时表格的名字是一样的。

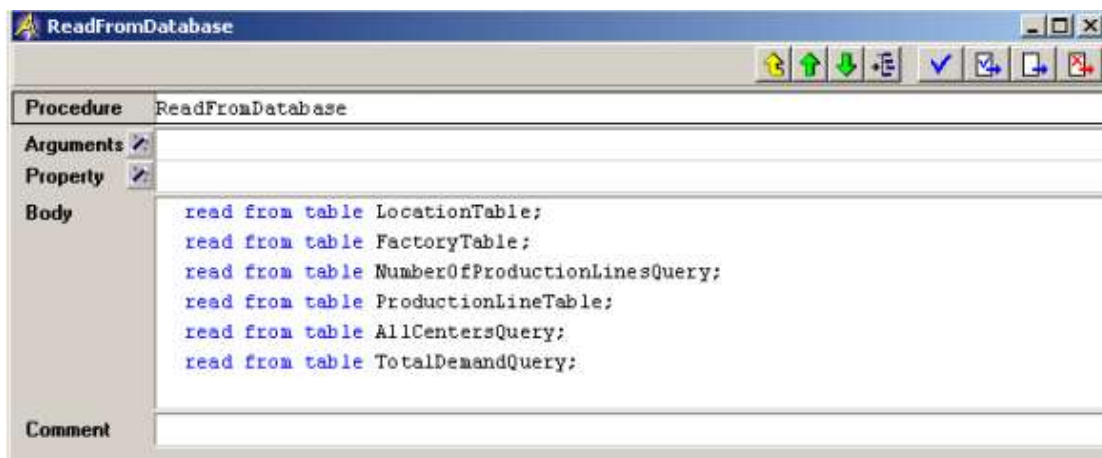



图 9.1: 程序 ‘ReadFromDatabase’

完成了程序 ‘ReadFromDatabase’ 的 Body 属性之后，使用 Check, commit and close 按钮 。现在你可以通过执行以下操作，运行这个程序：

- 选择模型树的程序 ReadFromDatabase
- 使用鼠标右键的弹出菜单（见图 9.2）选择 Run Procedure 命令

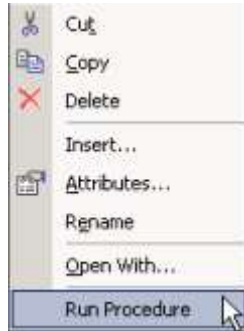



图 9.2: 程序 ‘ReadFromDatabase’ 的鼠标右键菜单

执行程序 ‘ReadFromDatabase’ 后，你可能想看比如参数 MaximumProductionLineLevel 的内容。在你浏览它的数据之前，你需要在模型树中找到这个参数。你可以通过以下方式找到它：

- 按工具栏的 Find 按钮 
- 使用名称完成工具（见图 9.3）输入 ‘MaximumProductionLineLevel’
- 按 Declaration...按钮

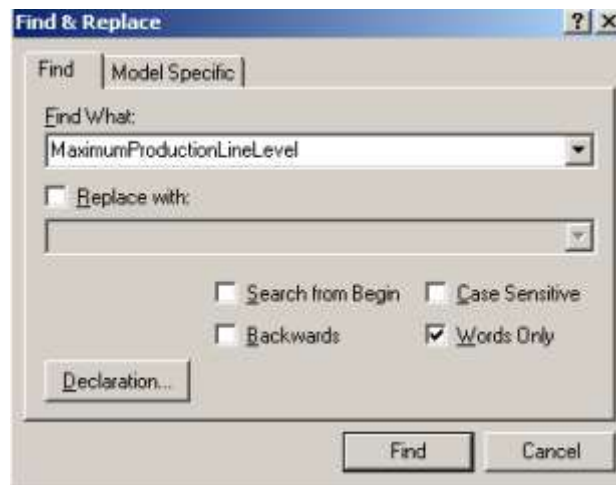


图 19.3: Find & Replace 对话框

然后，执行以下两个步骤，打开参数 MaximumProductionLineLevel 的数据页面：

- 按鼠标右键按钮，启动弹出菜单，然后
- 选择 **Data...**命令

现在你的电脑上的数据页面应该如图 9.4 所示。

p	line-01	line-02	line-03	line-04
f				
Eindhoven	450	550	550	600
Haarlem	450	500	550	600
Zwolle	450	550	550	600

图 9.4: MaximumProductionLineLevel 的数据页面

9.2 外部的 DLL 函数

在这一节，你将链接一个名为“External Routines.dll”的外部 Dynamic Link Library (DLL)到你的 AIMMS 模型。这个 DLL 里有一个命名为 DLLUnitTransportCost 的函数，它基于某个工厂和某个配送中心的距离，决定单位运输成本。写你自己的 DLL 教程不是本教程的范围之内。但是，The Language Reference 的第 11 和 30 章进一步阐述了 DLL 和相关的 AIMMS 编程界面的使用。“External Routines.dll”的源代码已经复制到你的项目的“DLL”子目录。

DLL “External Routines.dll”输出以下函数。

```
double DLLUnitTransportCost( char *from_name, char *to_name )
```

函数的两个输入变量代表计算单位运输成本的两个地方的名字的字符串。

在 AIMMS 应用中使用每个外部 DLL 函数，首先必须在 AIMMS 声明一个相应的外部函数。在本教程，这个外部函数命名为 ExternalUnitTransportCost，而且和它的外部函数有相同数量的变量。

要声明外部函数，你需要执行以下步骤：

- 打开 DLL Link 模型部分
- 按工具栏的 Other...按钮
- 从 Select Type of Node (见图 9.5) 对话框选择外部函数
- 指定 ‘ExternalUnitTransportCost(factory, center)’ 作为函数名，然后
- 按 Enter 键，注册这个名字

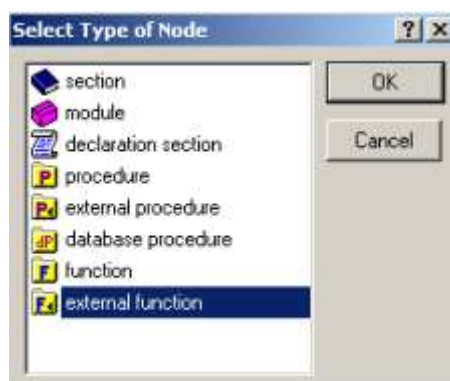


图 9.5: Select Node Type 对话框

接下来，AIMMS 会自动打开 Arguments 向导，如图 9.6 所示。

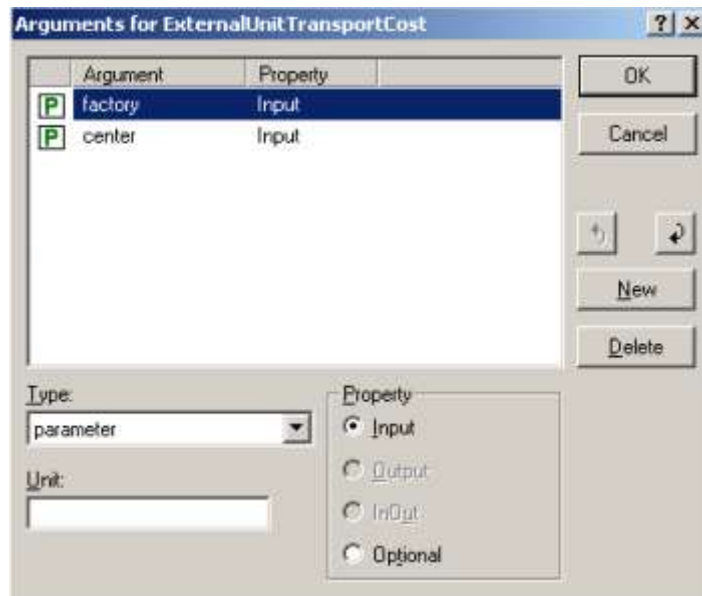


图 9.6: Arguments 向导

要完成 Arguments 向导，请执行以下步骤：

- 把目前选择的变量类型 factory 改至 ‘element parameter’，
- 选择 Factories 作为它的 Range 属性
- 然后点击第二个变量 center
- 把它的类型改至 ‘element parameter’，
- 选择 Centers 作为它的 Range 属性，然后
- 按 OK 按钮

完成了 Arguments 向导后，AIMMS 会声明两个输入变量作为局部元素参数。你可以验证 AIMMS 确实已经把在新声明节的这些局部参数放在模型树（见图 9.7）的 ExternalUnitTransportCost 节点。



图 9.7: 模型树的完整 DLL 部分

使用向导可以直接地完成外部函数的 DLL name 和 Return type 属性，如图 9.8 所示。

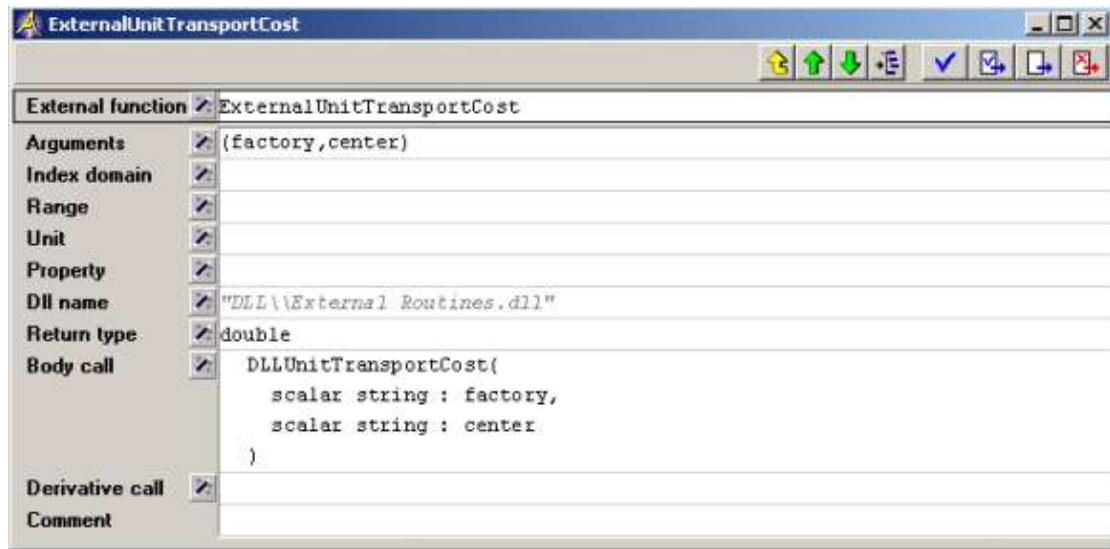




图 9.8：外部函数 ExternalUnitTransportCost 的属性形式

Body call 属性指定 AIMMS 和 DLL 的函数变量的实际链接。广泛的 Body call 向导支持建立该链接的几个选择，如图 9.9 所示。在 Body call 向导（见图 9.9），你需要执行以下操作：

- 选择 ‘Scalar’ 转换类型
- 按向导按钮 ，选择元素参数 factory 作为实际变量
- 设置外部数据类型为 ‘String’
- 按 Add 按钮
- 选择 ‘Scalar’ 转换类型
- 按向导按钮 ，选择元素参数 center 作为实际变量
- 设置外部数据类型为 ‘String’
- 按 Add 按钮
- 按 OK 按钮

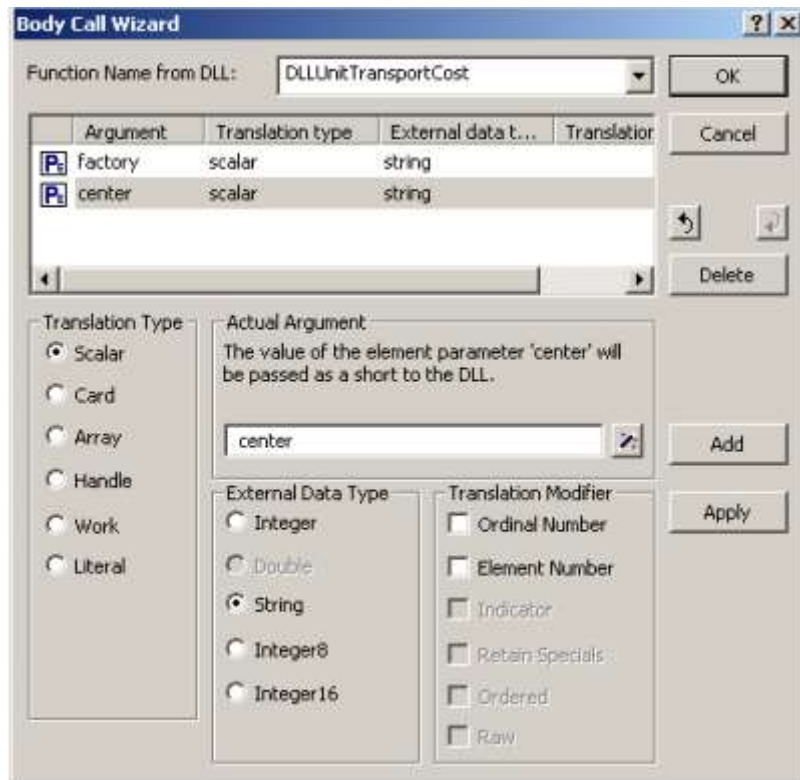


图 9.9: Body call 向导

9.3 设置周期滚动

在这一节你将设置需要描述周期滚动过程的所有程序。一旦你执行了这个过程的一个步骤，描述整个过程就变得简单。在适当的数据初始化之后，你就可以运行周期滚动程序的完整集。

如图 9.10 所示，这一节分为四个小节。你应该把这些小节添加到你的模型树。

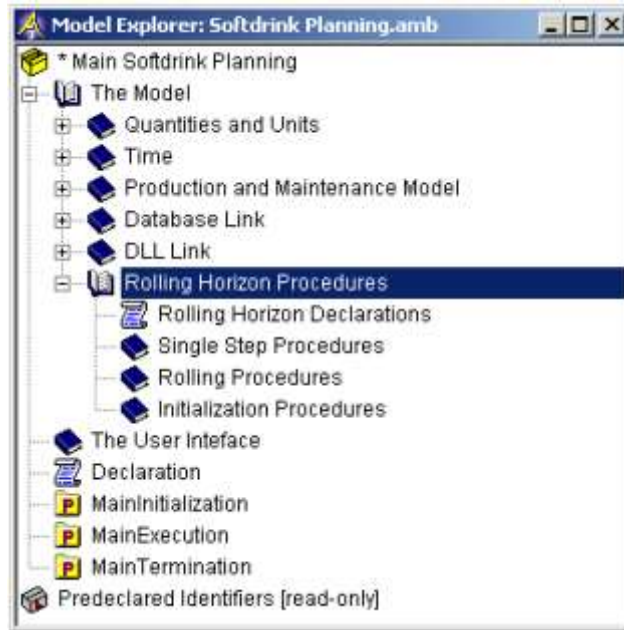


图 9.10: Rolling Horizon Procedures 部分的结构

9.3.1 周期滚动声明

有几个标识符在周期滚动过程中起到作用。它们的名称都是不言自明的，而且它们的内容设置如下。你将在下一个小节中见到，这些标识符用来组成时间表，而这些时间表把周期滚动模型的抽象时期链接到两个日历的具体的天和星期。

在这个阶段，你需要在 Rolling Horizon Declarations 中输入以下声明。

ELEMENT PARAMETER:

identifier : FirstDayInPlanningInterval

range : Days

SET:

identifier : WeeksInPlanningInterval

subset of : Weeks

definition : union[t, WeekInPeriod(t)]

ELEMENT PARAMETER:

identifier : FirstWeekInPlanningInterval

range : Weeks

definition : DayToWeek(FirstDayInPlanningInterval)

ELEMENT PARAMETER:

identifier : LastWeekInPlanningInterval

range : Weeks

definition : last(WeeksInPlanningInterval)

PARAMETER:

```
identifier : LengthDominatesNotActive
index domain : t
```

下一小节讨论的程序 CreateTimeTable 输入过程需要名为 LengthDominatesNotActive 的标识符。当这个标识符的值为默认值 0 时，由于分隔符位置的存在，任何时期的理想长度可能都无法实现。在本教程的例子中，这个参数确实是 0。因此，时间表 DaysInPeriod 将确认每个时期从周一（分隔符位置）开始。尽管每个时期的理想长度已经设为 7 天，但由于周末和官方假日（所谓的非工作时间），它的实际长度会缩短。

除了这五个标识符外，你还需要输入两个注册标识符。这两个标识符是用来储存总体维护和行用法规划。在 Rolling Horizon Declarations 节后面添加以下的两个参数：

PARAMETER:

```
identifier : OverallMaintenancePlanning
index domain : (f,p,w) | p in FactoryProductionLines(f)
```

PARAMETER:

```
identifier : OverallLineUsagePlanning
index domain : (f,p,w) | p in FactoryProductionLines(f)
```

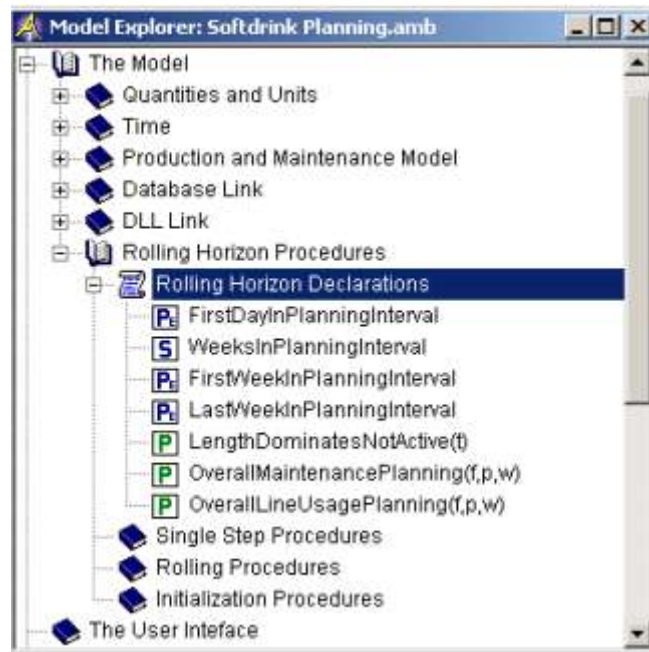


图 9.11: 模型树的 Rolling Horizon Declarations 节

9.3.2 单步骤程序

周期滚动决策过程的一个步骤可以分为如图 9.12 所示的几个程序。本小节将讨论每个程序的实行。相应地完成你的模型树，但是在输入有两个命名为 iw 和 ip 的变量的程序 RegisterInOverallPlanning 时，请遵循下一段的指示。

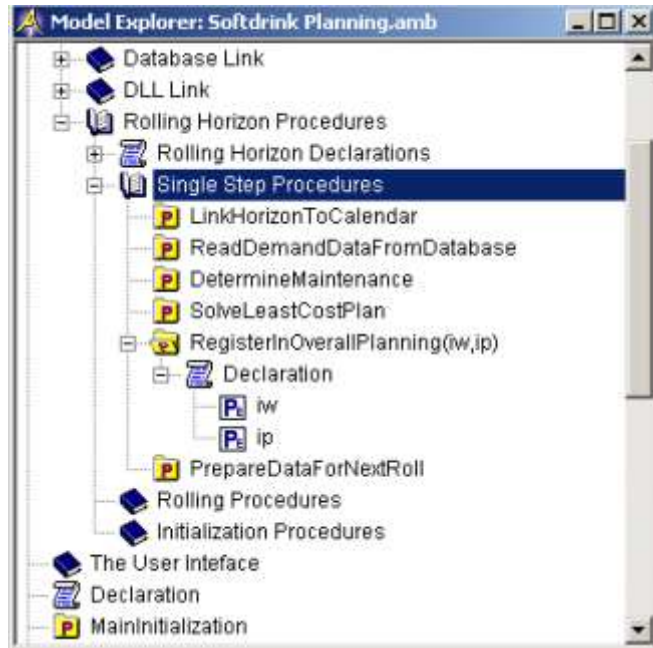


图 9.12: 需要指定一个步骤的程序

当你输入有两个变量的程序 RegisterInOverallPlanning(iw, ip), AIMMS 将自动打开向导。要完成这个 iw (指某一周) 和 ip (指某个时期) 的 Argument 向导, 你应该执行以下步骤:

- 把现在选择的变量 iw 类型改至 ‘element parameter’
- 选择 Weeks 作为它的 Range 属性
- 选择 ‘Input’ 作为它的 Property 属性
- 然后点击第二个变量 ip, 改变目标
- 把它的类型改至 ‘element parameter’
- 选择 Periods 作为它的 Range 属性, 然后
- 选择 ‘Input’ 作为它的 Property 属性

现在, Argument 向导应该和图 9.13 所示的一样。

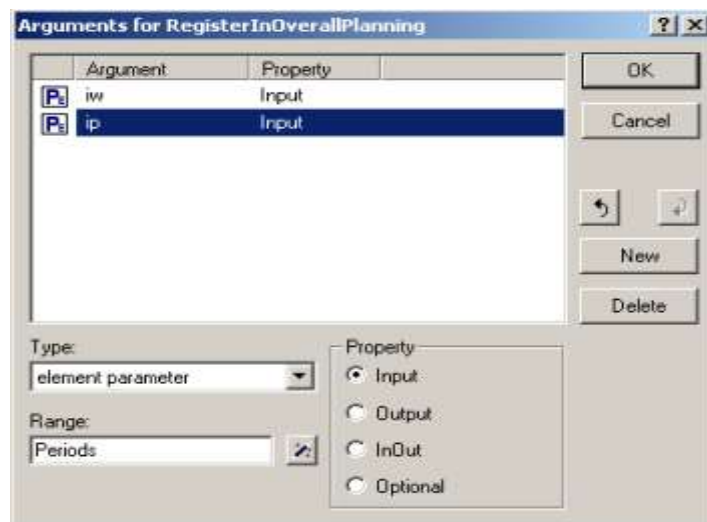


图 9.13: Argument 向导

时间表是一个索引集或索引元素参数，代表范围中时期和日历中时隙的映射。当时期可以包含几个时间位置时，例如在时间表 DaysInPeriod，它便是一个索引集。当存在每个时期和每个时间位置的一一映射，例如在时间表 WeekInPeriod，它也可以是索引元素参数。

事先定义的程序 CreateTimeTable 的 quick info tip 窗口如图 9.14 所示。

```
CreateTimeTable(  
  [Output] TimeTable AS Set,  
  [Input] CurrentTimeSlot AS Element Parameter,  
  [Input] CurrentPeriod AS Element Parameter,  
  [Input] PeriodLength AS Parameter,  
  [Input] LengthDominates AS Parameter,  
  [Input] InactiveTimeSlots AS Set,  
  [Input] DelimiterSlots AS Set)
```

图 9.14: 函数 CreateTimeTable 的 quick info tip 窗口


你通过变量可以控制时间表的内容。若想进一步了解，请参考 The Language Reference 手册的 29.4 节。

移至程序 LinkHorizonToCalendar 的 Body 属性，并输入以下语句：

```
CreateTimeTable(  
  TimeTable : DaysInPeriod,  
  CurrentTimeSlot : FirstDayInPlanningInterval,  
  CurrentPeriod : FirstPeriodInPlanningInterval,  
  PeriodLength : DesiredNumberOfDaysInPeriod,  
  LengthDominates : LengthDominatesNotActive,  
  InactiveTimeSlots : InactiveDays,  
  DelimiterSlots : Mondays );  
ActualNumberOfDaysInPeriod(t) := (card(DaysInPeriod(t))) [day];  
CreateTimeTable(  
  TimeTable : WeekInPeriod,  
  CurrentTimeSlot : FirstWeekInPlanningInterval,  
  CurrentPeriod : FirstPeriodInPlanningInterval,  
  PeriodLength : DesiredNumberOfWeeksInPeriod,  
  LengthDominates : LengthDominatesNotActive,  
  InactiveTimeSlots : InactiveWeeks,  
  DelimiterSlots : Weeks );
```

注意，当调用 CreateTimeTable 时，变量的前面是它们的变量名，如图 9.14 所示。在 AIMMS，在函数调用中，变量名的使用的可选择的。在上述 Body 属性中，变量名的使用时为了增加可读性。

为了保证上述赋值语句的单位一致性，无单位的表达式 card(DaysInPeriod(t)) 被赋予单位 [day]。这种单位设置需要圆括号直接的表达式是封闭的。

你可以使用工具栏的 Maximize 按钮 ，暂时增加 Body 属性（或其他任何多重线属性）的大小以方便输入。当你完成了这个属性，再次按 Maximize 按钮就可以恢复到原来的大小。时间表 DaysInPeriod 包含一周的工作日，而且明确排除了非工作时间如周末和官方假日。创建这种时间表的唯一原因就是决定需要制定正确生产水平的参数

ActualNumberOf-DaysInPeriod。

要浏览 DaysInPeriod 时间表的内容，首先你需要初始化元素参数 FirstDayIn-PlanningInterval。其他输入参数已经初始化了的。执行以下步骤：

- 选择模型树的程序 LinkHorizonToCalendar
- 按 Enter 键，打开属性表
- 把文本光标放在 Body 属性的字符串 ‘FirstDayInPlanningInterval’ 的某处
- 按鼠标右键按钮，启动弹出菜单
- 选择 Data...命令
- 点击 Data 页面的等号的空白右边处
- 指定 ‘03/07/2000’（没有引号）作为数据页面的值，然后
- 按 Close 按钮

你可以重新打开页面验证 AIMMS 以及接收你的输入值。如果你的输入格式是不正确的，AIMMS 会用默认的空白字符串代替你的输入。

现在你可以运行程序并查看适合的数据页面，这样就可以浏览时间表 DaysInPeriod 的内容。

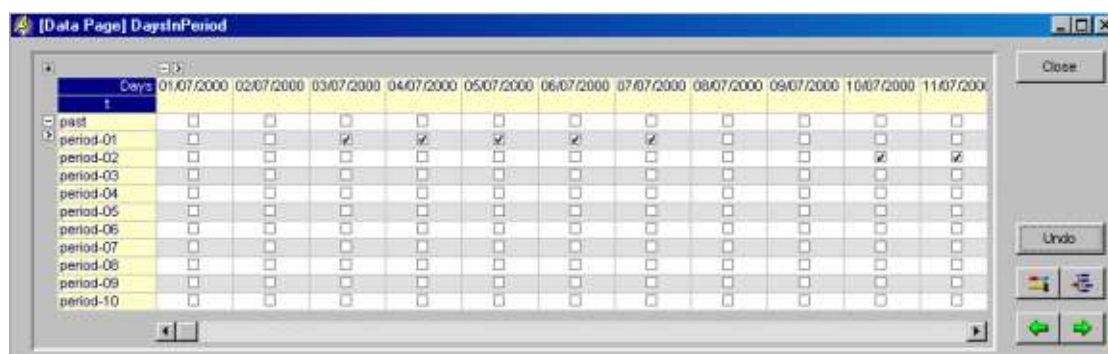
- 把文本光标放在 Procedure 属性的字符串 ‘LinkHorizonToCalendar’ 的某处
- 按鼠标右键按钮，启动弹出菜单，然后
- 选择 Run Procedure 命令

由于目前的默认值足够本教程里的使用，你可以忽略所有的初始化警告。请关闭 Errors/Warnings 窗口并继续。

接下来构建对应于如图 9.15 所示的时间表 DaysInPeriod 的数据页面，并执行以下步骤：

- 把文本光标放在 Body 属性的字符串 ‘DaysInPeriod’ 的某处
- 按鼠标右键按钮，再次启动弹出菜单，然后
- 选择 Data...命令

注意，由于周末被排除在外，每个时期刚好包括五天。数据页面的默认格式要求你水平滚动滚动条。你可以通过按 Change View 按钮，选择不同的视图，并选择，例如，‘Sparse List’ 作为对象类型。



The screenshot shows a data page titled '[Data Page] DaysInPeriod'. It features a grid with columns representing dates from 01/07/2000 to 11/07/2000 and rows representing different periods from 'past' to 'period-10'. Each cell in the grid contains a checkbox. The 'past' row has all checkboxes unchecked. The 'period-01' row has checkboxes checked for dates 03/07/2000, 04/07/2000, 05/07/2000, 06/07/2000, and 07/07/2000. The 'period-02' row has checkboxes checked for dates 09/07/2000 and 10/07/2000. All other rows have all checkboxes unchecked. The interface includes a 'Close' button at the top right, an 'Undo' button, and a 'Change View' button at the bottom right.

Days	01/07/2000	02/07/2000	03/07/2000	04/07/2000	05/07/2000	06/07/2000	07/07/2000	08/07/2000	09/07/2000	10/07/2000	11/07/2000
past	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
period-01	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
period-02	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
period-03	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
period-04	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
period-05	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
period-06	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
period-07	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
period-08	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
period-09	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
period-10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

图 9.15：以天数为基础的时间表的数据页面

本教程的每周日历跨越了几乎为一年的周期。但是，总体的周期滚动程序的每一个步骤的规划范围却只是星期的一个小的子集。这就是为什么程序 ReadDemandDataFromDatabase

被引入到限制某个时期加载到内存的需求数据的总量。

在周期滚动过程的每个后续步骤之前,推荐的做法是先清空和旧的规划间隔相关的每周需求数据。这个可以通过在程序 ReadDemandDataFromDatabase. 的 Body 属性中输入以下语句。

```
empty WeeklyDemand;  
read WeeklyDemand(c,w,s) from table CenterTable  
filtering w in WeeksInPlanningInterval;  
Demand(c,t,s) := WeeklyDemand(c,WeekInPeriod(t),s);
```

注意每周的需求只针对那些在目前规划间隔的星期读取。使用时间表 WeekInPeriod, 每周的需求分配到待解决的数学程序所需要的时期需求。

参数 DeteriorationLevel 记录对于每个工厂和生产线的组合,自从上次生产线维修经过的时间。假设所有生产线对于整个规划期都是处于使用状态,要估计某个生产线什么时候处于维修状态是简单的。

如果有超过一位的候补者,你应该只维持一个生产线,并延迟其他候补者的维修到下个时期。最终的结果储存在为每一个工厂、生产线和时期声明的参数 LineInMaintenance。这个参数是使用中(见参数 PotentialProduction 的定义)的生产线的生产水平的其中一个决定因素。

在指定程序 DetermineMaintenance 的 Body 属性前,你需要在程序节点 DetermineMaintenance 的新声明节声明下列两个局部标识符。

```
ELEMENT PARAMETER:  
identifier : EstimatedMaintenancePeriod  
index domain : (f,p)  
range : Periods  
SET:  
identifier : LinesInMaintenanceInFirstPeriod  
index domain : f  
subset of : ProductionLines
```

图 9.16 显示程序 DetermineMaintenance 的局部声明节。

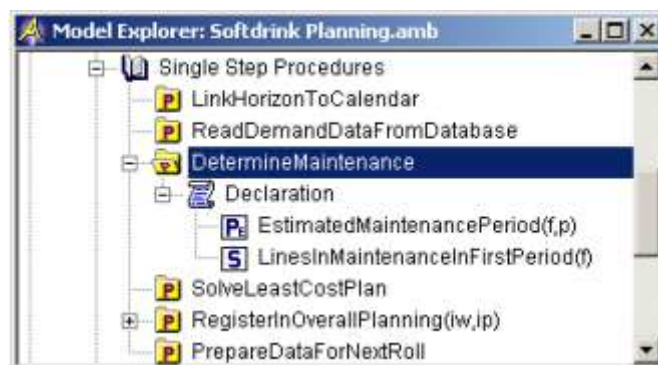


图 9.16: 程序 DetermineMaintenance 的局部声明

以下 AIMMS 中的声明已经在之前段落中讨论过。请在程序 DetermineMaintenance 的 Body 属性中输入它们。

```

EstimatedMaintenancePeriod(f, p) :=
Element(
    Periods,
    max(
        MaximumDeteriorationLevel(f, p)
        -Floor(DeteriorationLevel(f, p) + 2, 2)
    )
);
LinesInMaintenanceInFirstPeriod(f) :=
{ p | EstimatedMaintenancePeriod(f, p) = FirstPeriodInPlanningInterval };
EstimatedMaintenancePeriod( (f, p) |
Ord(p, LinesInMaintenanceInFirstPeriod(f)) >= 2 ) += 1;
empty LineInMaintenance;
LineInMaintenance((f, p, EstimatedMaintenancePeriod(f, p)) |
EstimatedMaintenancePeriod(f, p) in Periods.Planning ) := 1;

```

完成了前三个单步骤程序后，你现在可以输入解决单步骤数学程序的指令。请在程序 SolveLeastCostPlan 的 Body 属性中输入以下语句。

```

solve LeastCostPlan;
halt with "Least cost mathematical program is not optimal.\nCheck "
+ "input data for infeasibilities.
when ( LeastCostPlan.ProgramStatus <> ' Optimal' );

```

注意，第二个语句阐述 halt 语句在 AIMMS 的使用。一旦程序停止，它提供了一个两行的消息，正如特殊符号 ‘\n’ 所示。通过使用 Body 属性的+符号，你可以把一个单引号的字符串划分为几块。在 halt 语句的条件 when 部分，存在利用点符号（见 The Language Reference 的 15.2 节）对数学程序即 the program stus 的引用。

在单步骤数学程序的解决之后，和第一个时期相关的结果保持为 ‘definite’（有限）。在本教程，只有维护的总规划和生产线的总规划得以保存。总规划以日历周的形式记录下来，也即是说时期数据必须转换为周数据。这种转换是通过以下两个增加到程序 RegisterInOverallPlanning 的 Body 属性的语句实现的。

```

OverallMaintenancePlanning(f, p, iw) := LineInMaintenance(f, p, ip);
OverallLineUsagePlanning(f, p, iw) := ProductionLineInUse(f, p, ip);

```

一旦总体规划记录之后，剩下的就是为下一步骤准备几个数据项目。首先，规划间隔的第一天必须向前移动七天至下周一。然后目前的第一时期库存和生产解决数据必须变成历史数据。最后，所有生产线的退化水平必须适当地调整。所有的这些分配任务都在程序 PrepareDataForNextRoll 的以下 Body 属性中实现。

```

FirstDayInPlanningInterval += 7;
Stock(l, ' past' , s) :=
Stock(l, FirstPeriodInPlanningInterval, s);
ProductionLineInUse(f, p, ' past' ) :=
ProductionLineInUse(f, p, FirstPeriodInPlanningInterval);

```



```

DeteriorationLevel(f, p) +=
0.75 * ProductionLineInUse(f, p, FirstPeriodInPlanningInterval) + 0.25;
DeteriorationLevel( (f, p) |
LineInMaintenance(f, p, FirstPeriodInPlanningInterval) ) := 0;

```

注意，一个生产线的退化水平以 1 更新，表示生产线在规划间隔的第一个时期使用。否则的话，退化水平只增加了 0.25，表示生产线在那一周是空闲的。当然，如果生产线在第一个时期处于维修状态，它的退化水平便重置为 0。

9.3.3 滚动程序

我们可以考虑两个周期滚动程序。其中之一包含了需要执行周期滚动过程的所有程序。你可以使用相应的鼠标右键有次序地执行这个程序，并检验发现的结果。第二个程序一次性执行所有剩下的单步骤。请更新你的树结构的 Rolling Procedures 节，如图 9.17 所示。

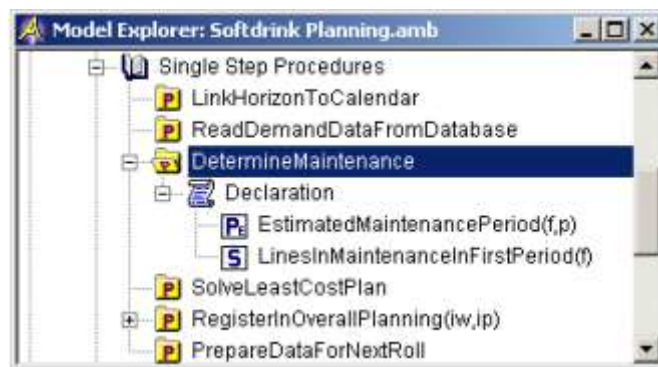


图 9.17: Rolling Procedures 部分的结构

以下序列的语句执行周期滚动过程的一个单步骤。请在程序 RollHorizonOnce 的 Body 属性中输入它们。注意，每一个语句都是对前面小节中开发的程序的调用。

```

LinkHorizonToCalendar;
ReadDemandDatafromDatabase;
DetermineMaintenance;
SolveLeastCostPlan;
RegisterInOverallPlanning(FirstWeekInPlanningInterval, FirstPeriodInPlanningInterval);
PrepareDataForNextRoll;

```

以下程序完成了由元素参数 FirstWeekInPlanningInterval 决定的日历现在时间开始的周期滚动过程。在下一小节，你将遇到可以让你从日历的起到开始周期滚动过程的程序。请在程序 RollHorizonToEnd 的 Body 属性中输入以下语句。

```

while ( LastWeekInPlanningInterval < LastWeekInCalendar ) do

```

```

RollHorizonOnce;
endwhile;
for (t | t > FirstPeriodInPlanningInterval) do
RegisterInOverallPlanning(WeekInPeriod(t), t);
endfor;

```

注意, 最终的规划间隔的维护和行使用规划不只是通过程序 RollHorizonOnce 记录第一个时期, 而且还通过语句的执行记录剩下的时期。

9.3.4 初始化程序

我们考虑三种初始化程序。第一种是系统提供的程序 MainInitialization, 它随着每次项目启动而执行。另外两种已经嵌入在 MainInitialization, 但也可以分别调用。请更新你的树结构, 如图 9.18 所示。确认不要创建一个 MainInitialization 程序, 因为已经有一个存在你的模型中。只需把它从模型树的后面移至理想的位置(使用剪切-黏贴或者 AIMMS 的拖放工具)。



图 9.18: Initialization Procedures 节的结构

在程序 InitializeLengthOfPlanningInterval 中, 我们设置周期滚动的两个重要的参数。它们的值决定了周期滚动过程的一个步骤考虑的时间。如果你想考虑不同的规划间隔, 你可以改变这些值。请把以下语句输入到 Body 属性中。

```

NumberOfPeriods := 10;
NumberOfPeriodsInPlanningInterval := 8;

```

程序 MovePlanningIntervalToStartOfCalendar 首先清空任何已存的总体维护和生产线使用规划, 然后把日历开端的所有初始值赋给适合的变量和参数。任何时候都可以调用这个程序, 导致任何激活的周期滚动程序在日历开端启动, 请把以下语句输入到 Body 属性。

```

empty OverallMaintenancePlanning, OverallLineUsagePlanning;

```

```

Stock(l, ' past' , s) := StockAtStartOfCalendar(l);
ProductionLineInUse(f, p, ' past' ) := 1 onlyif
ProductionLineLevelAtStartOfCalendar(f, p);
DeteriorationLevel(f, p) := DeteriorationLevelAtStartOfCalendar(f, p);
FirstDayInPlanningInterval := first( Mondays );
WeekInPeriod(t) := Element( Weeks, Ord(t) );

```

AIMMS 在每次运行的开始执行程序 MainInitialization, 它是读取数据、初始化各种参数和启动初始化你的模型数据的其他程序的自然开始点。在本教程, 程序 MainInitialization 从数据库表格读取所有问题数据。唯一的例外是需求数据, 它每次从程序 RollHorizonOnce 的目前规划范围中读取一节。然后, 通过调用 9.2 节开发的外部程序可以获得单位运输成本。最后, 通过调用上述描述的两个程序, 可以完成周期滚动程序所需要的数据初始化。请用以下语句代替 MainInitialization 程序的内容。

```

ReadFromDatabase;
UnitTransportCost(f, c) := (ExternalUnitTransportCost(f, c)) [$/TL];
InitializeLengthOfPlanningInterval;
MovePlanningIntervalToStartOfCalendar;
empty LengthDominatesNotActive, InactiveWeeks;

```

注意单位[\$/TL]归于外部函数的输出。这就要求你把圆括号放在函数调用的旁边, 如上述所示。

9.4 运行模型

如上述所示, 当项目打开时, 你在 MainInitialization 程序输入的语句便开始执行。尽管你可以使用鼠标右键的 Run Procedure 命令来直接运行这个程序, 你也可以通过先关闭项目, 然后再打开它来试验默认的操作。要这样做, 请执行以下步骤, 关闭你的项目:

- 从 AIMMS 的 File 菜单选择 Close Project 命令
- 在关闭项目之前, 被问到编译你的模型时, 回答 ' No'
- 在问你是否想储存你的数据 (见图 9.19) 时, 回答 ' No'
- 回答 ' Yes', 储存更改的项目

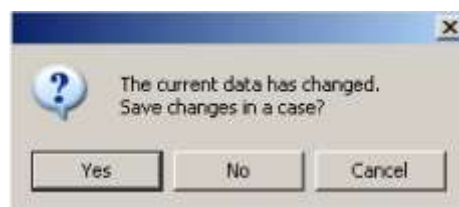


图 9.19: Save Changes 对话框

打开一个你刚关闭的项目是很简单的。AIMMS 保存最近五个打开的项目的记录。只需从 AIMMS 的 Start Page 显示的项目列表中选择 ' Softdrink Planning'。或者, 你可以从 File

菜单（见图 9.20）中选择最近的项目。

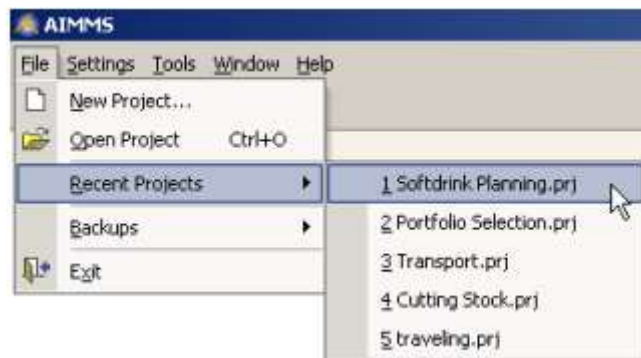



图 9.20: AIMMS 的 File 菜单

AIMMS 窗口的标题栏左边的星号表明你对项目的最近更改尚未保存到硬盘。要保存你的更改，请按工具栏左边的第二个位置的 Save Project 按钮 。

现在你可以检验从日历开端开始的周期滚动过程。要运行程序 RollHorizonOnce，你应该执行以下步骤：

- 选择模型树的程序 RollHorizonOnce，然后
- 在鼠标右键菜单中选择 Run Procedure 命令（见图 9.21）

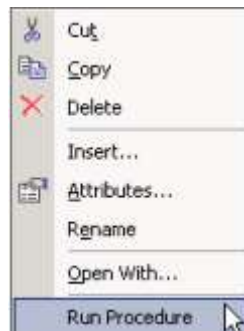


图 9.21: 程序 RollHorizonOnce 的鼠标右键菜单

Progress 窗口让你监测数学程序的生成和解决过程中，AIMMS 和求解工具的进度。通过按 Ctrl-P 组合键，如图 9.22 所示的 Progress 窗口就会出现。一旦发现了解决方案，AIMMS 就会再次发出数据未初始化的警告。在教程的现在阶段，我们可以忽略这些警告。

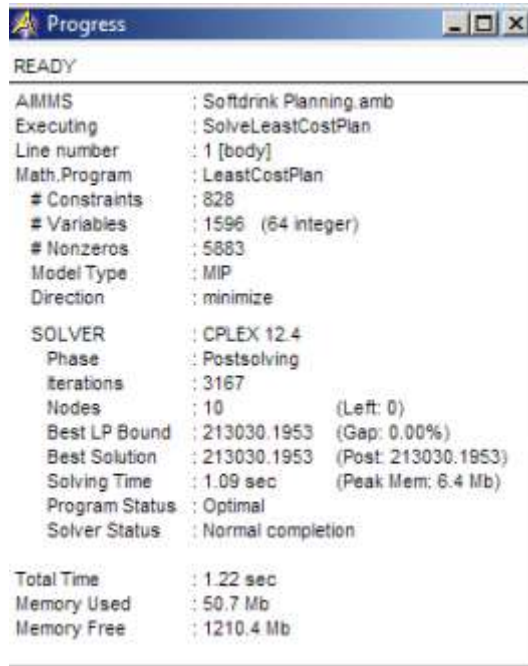


图 9.22: Progress 窗口

一旦程序 RollHorizonOnce 完成,你可以浏览结果。例如,你可以打开和变量 TotalCost 相关的数据页面,并比较它的值和图 9.22 的 Progress 窗口的值。类似的,你可以检查任何决策变量的值。例如,变量 Production 的最优值如图 9.23 所示。

t	period-01	period-02	period-03	period-04	period-05	period-06	period-07	period-08
Eindhoven	3000	3000	3000	3000	3000	3000	3000	3000
Haarlem	7750	7750	7750	4650	3150	4650	4650	4650
Zwolle	2750	2750	1650	1650	1650	1650	1650	1650

图 9.23: 变量 Production 的数据页面

AIMMS 默认在数据透视表显示非标量数据。对于变量和限制条件,当存在的话,额外的信息(例如边际值,基本状况)都会显示在数据透视表。注意,基本状况显示在变量 Production 的数据页面中。

在本教程的这个阶段,你已经到达了一个重大里程碑,因为我们已经完成了周期滚动应用的完整模型描述。在本教程的下一部分,你将专注于为这个应用的终端用户构建图形用户界面。

第四部分

构建用户终端界面

第十章

页面和模板的管理

在下面章节，您将通过使用页面管理器构建用户终端界面的整体结构。并且通过使用模板管理器指定用户终端界面类型。在本章的结尾，您将创建一个启动页面。该页面包含其他全部页面的参考目录。

设计一个有效的用户终端界面是一个反复的过程，它需要与终端用户进行互动。第十二章使用者向导包含了一些基本的设计原则。然而在本教程中，不需要任何重新设计既可构建特定的终端界面。

10.1 页面管理器

在 AIMMS 中，页面相当于终端用户可视化的信息窗口。通过页面管理器管理页面，允许您以树状方式组织全部终端用户窗口。在页面树中，页面组织定义了终端界面的导航结构，相对于页面树中的特定页面，其他页面位置定义了页面之间关系，譬如：上层页面，下层页面，下一页，前一页。您可以通过类似按钮和菜单使用导航架构。图 10.1 介绍了终端应用中常用的导航架构。

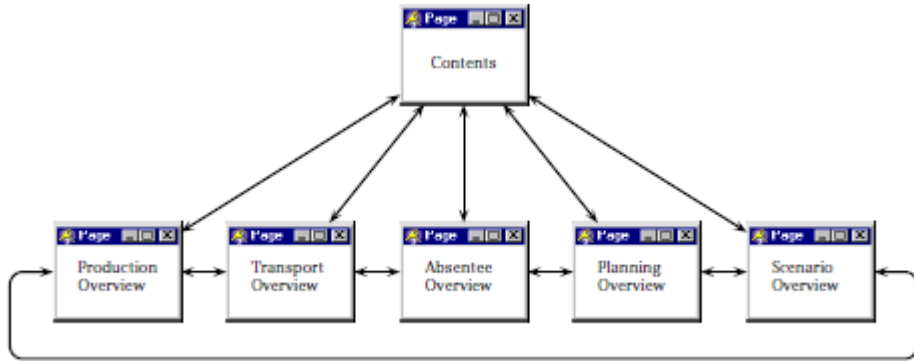



图 10.1 应用中的导航架构

为创建一个理想的页面结构，首先从 AIMMS 工具菜单栏中打开页面管理器，或者按 F9 键。打开的页面树如图 10.2 所示。注意到第五章所创建的试用页面会自动补充道页面管理器中，如果您之前储存了已修改的数据页面，则一个名为 ‘All Data Pages’ 的当前页面也会被补充道页面管理器，里面包含了储存的数据页面。




图 10.2 包含一个页面的页面管理器

在第五章中，您已创建了一个新的页面：

- 在工具栏中点击新页面按钮 ，创建一个新的页面。或者按 Insert 键。
- 为新页面命名 ‘Contents’，然后
- 按 Enter 注册页面

您可以按以下步骤在页面 Contents 中创建一个下层页面：

- 双击页面 Contents 图标打开页面 Contents
- 在工具栏中点击新页面按钮 ，创建一个新的页面
- 为新页面命名 ‘Production Overview’
- 按 Enter 注册页面

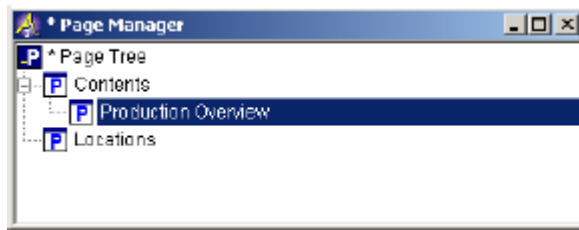


图 10.3 中间段（此时）页面

如图 10.4，完成页面树的构建。

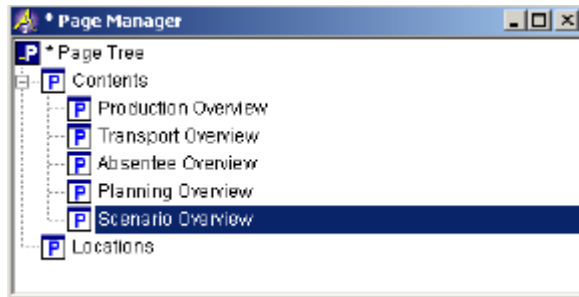



图 10.4 最终的导航结构

标题栏左上角的星号表明改变后的项目尚未保存到硬盘，点击工具栏中 ‘Save Project’ 按钮保存项目, 或者按 Ctrl+S 组合键。

六个页面中每一个目的内容如下所示：

- 目录：创建目录页面作为其他页面的导航
- 生产概要：生产概要页面包含了计划期间的最优生产水平以及维修计划表
- 运输概要：运输概要页面包含了计划期间工厂和配送中心的最优运输值以及相应的库存水平
- 缺席概要：缺席概要页面提供了一个交互型功能以方便指定节日和假期
- 计划概要：在所考虑的日历期间部分，计划概要页面将显示全部生产和维修计划
- 情况概要：情况概要页面将显示数据库中不同情况的需求数据

10.2 模板管理

使用模板管理器，您能确保所有终端界面同一尺寸并且具有相同的外观。您可以通过创建所谓的模板页面实现这种效果，该页面定义了常用的用户界面的页面属性和对象。这些模板页面可以内嵌到模板页面树中，此外，您需要在模板页面下合理安排用户界面作为子页面，以至模板页面中的对象能在用户界面中显示。

用户界面从模板页面中继承的页面对象和页面属性如下所示：

- 背景颜色和背景地图
- 商标 / 标志

- 导航按钮
- 页面菜单栏和工具栏
- 页眉和页脚区域
- 页面尺寸和调整行为


本教程的练习提供了一个关于背景颜色的模板和一个共享导航按钮的模板。

为创建一个理想的页面模板，首先从 AIMMS 工具菜单栏中打开页面管理器，或者按 F9 键。初始的模板树如图 10.5 所示，注意到初始模板树包含之前在页面管理器中已创建的全部页面。




图 10.5: 包含初始模板的模板管理器

接下来，你需要为背景颜色和导航按钮分别创建一个页面模板：

- 在模板树中选择根节点
- 点击工具栏中的新模板按钮 
- 为新模板命名背景地图，然后
- 按 Enter 注册模板

如图 10.6 所示，在第一个模板下层创建第二个页面模板：

- 双击背景地图模板的图标打开模板
- 点击工具栏中的新模板按钮 
- 为新模板命名导航按钮，然后
- 按 Enter 注册模板

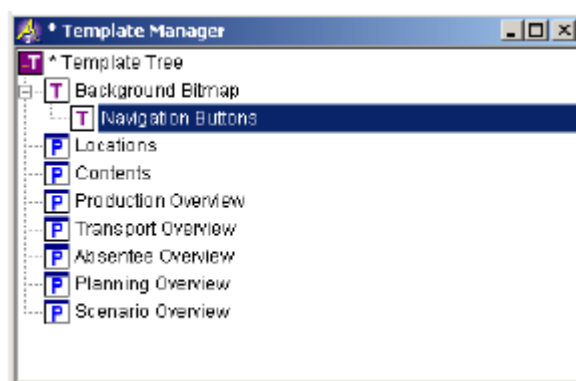


图 10.6: 包含模板树的模板管理器

页面管理器中创建的 6 个页面将自动出现在模板管理器。如图 10.7 所示，您可以通过移动目录页面使其继承地图背景。

- 在模板树中选择目录页面，然后
- 拖到该页面到背景地图模板

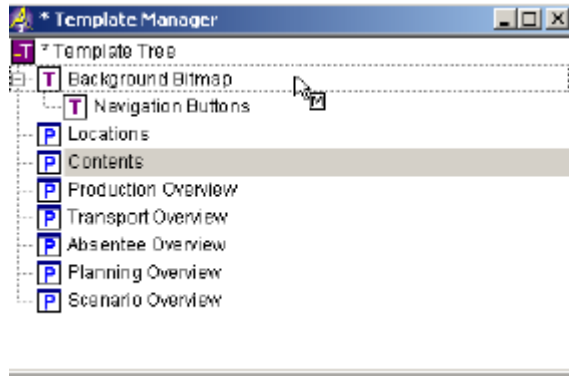


图 10.7：移动目录页时的模板管理器

接下来，如图 10.8 所示，请移动剩下的 5 个概要页面使他们继承地图背景和导航按钮：

- 双击导航按钮模板的图标打开模板
- 按住 SHIFT 键，移动鼠标选择模板树中全部 5 个概述页面，然后
- 拖到选中的页面到导航按钮模板下

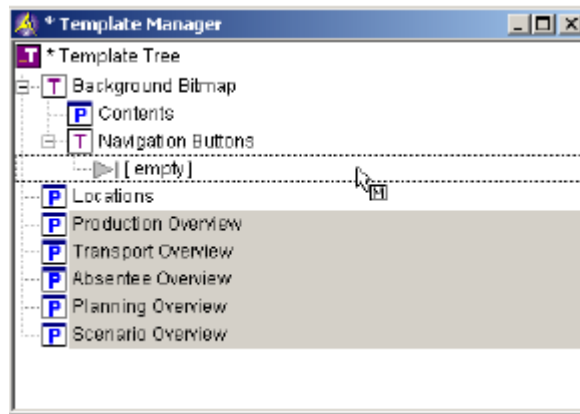


图 10.8：移动概述页面时的模板管理器

最终的模板树如图 10.9 所示：

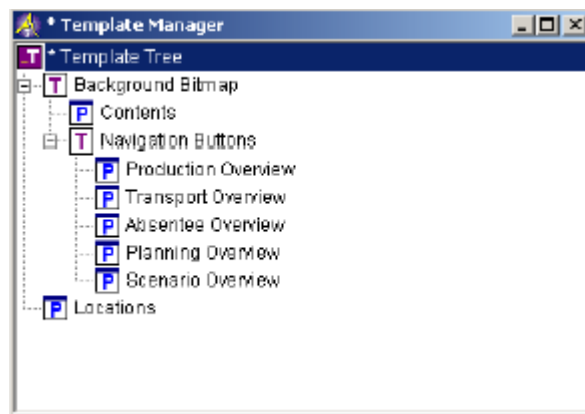



图 10.9 移动页面后的模板管理器

背景地图模板为全部终端用户界面提供了统一的界面，您可以按一下方式指定模板：

- 在模板树中选择 **Background Bitmap** 模板
- 点击工具栏中的 ‘Open in Edit Mode’ 按钮，打开模板

- 在 Object 目录中选择 ‘Picture’ 命令
- 移动鼠标光标到模板的左上角
- 按住鼠标左键，拖动鼠标光标到模板的右下角，然后
- 放开鼠标

此时，您需要完成 Picture Properties 对话框：

- 点击文件名编辑区域的右边的向导按钮 
- 在鼠标右击自动弹出的目录中，选择 ‘Select File Name. . .’ 命令
- 选择地图文件 ‘Bitmaps\Background.bmp’
- 点击 Open 按钮
- 选择 ‘Fill with multiple bitmaps’ 显示选项，然后
- 按 Enter 键结束

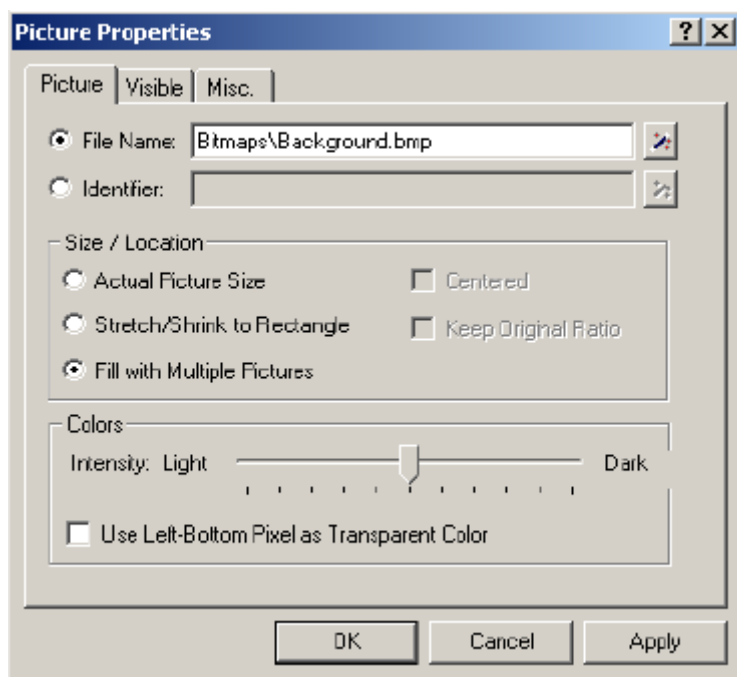
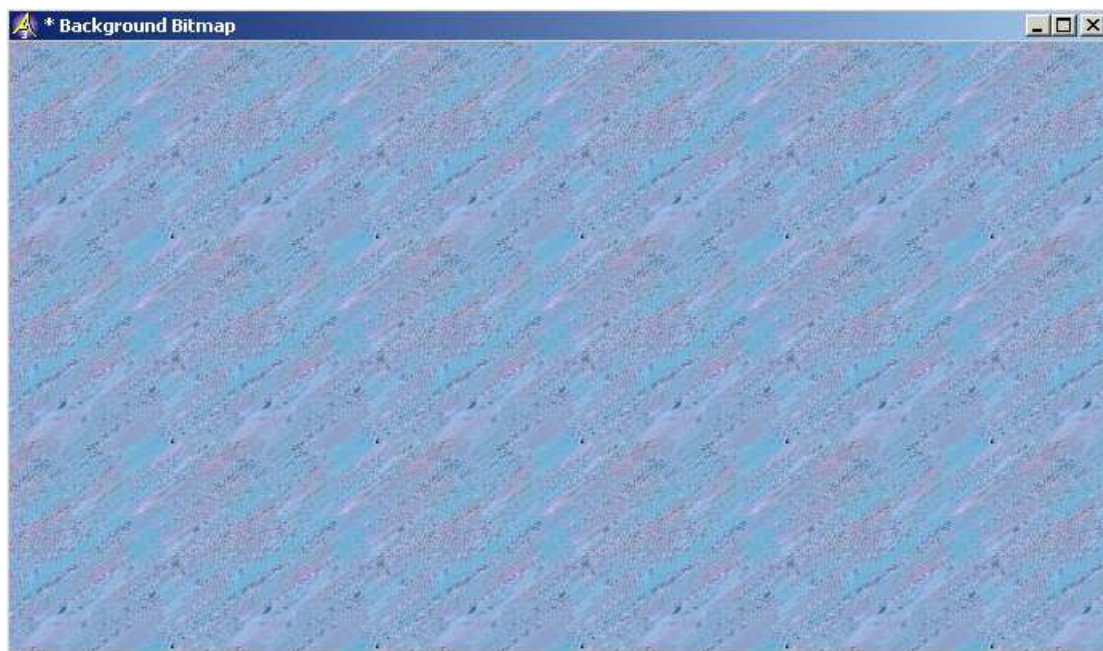



图 10.10: 图片属性对话框


通过选择选项 ‘Fill with Multiple Pictures’，如图 10.10 对话框所示，通过指令 AIMMS 复制文件夹 ‘Background.bmp’ 中小的位图，此时，屏幕充满了如图 10.11 所示的蓝色图案。



模板界面标题框中左上角的星号表明项目中增加的内容尚未保存, 点击工具栏中保存项目按钮  保存工作内容。

您可以按以下方式检查背景位图模板是否正确继承:


- 按 F9 键, 打开页面管理器, 然后
- 例如: 通过双击 Production Overview 页面打开它

‘Production Overview’ 页面与 ‘Background Template’ 页面类似, 一旦证实已被继承, 可以通过点击页面右上角的交叉十字按钮  关闭界面。

第二个模板为概述页面的向导按钮提供了专门的创建区域。以下三个按钮方便您访问:

- 下一页
- 前一页, 和
- 目录页

创建一个按钮, 允许通过单击该按钮进入下一页

- 在 Edit 模式中打开导航按钮模板
- 按下工具栏中按钮 New Button 
- 使用鼠标在页面的右下角拖拽出小的矩形
- 在按钮属性对话框中选择 “Bitmap Button” 选项
- 在鼠标右击弹出的目录中, 使用向导选择选择 Select File Name. . . 命令
- 选择文件夹 ‘Bitmaps\Button Next.bmp’
- 点击 OK 按钮

接下来, 再次打开按钮属性对话框, 如果 10.12 所示, 完成选项卡的操作

- 选择 ‘Actions’ 选项卡
- 选择 ‘Goto Page’ 选项
- 点击 Add 按钮, 系统默认选择 ‘Go to Previous Page’ 选项
- 选择 ‘Next Page’ 选项
- 按 Apply 按钮得到新的 ‘Go to Next Page’, 然后
- 点击 OK 按钮

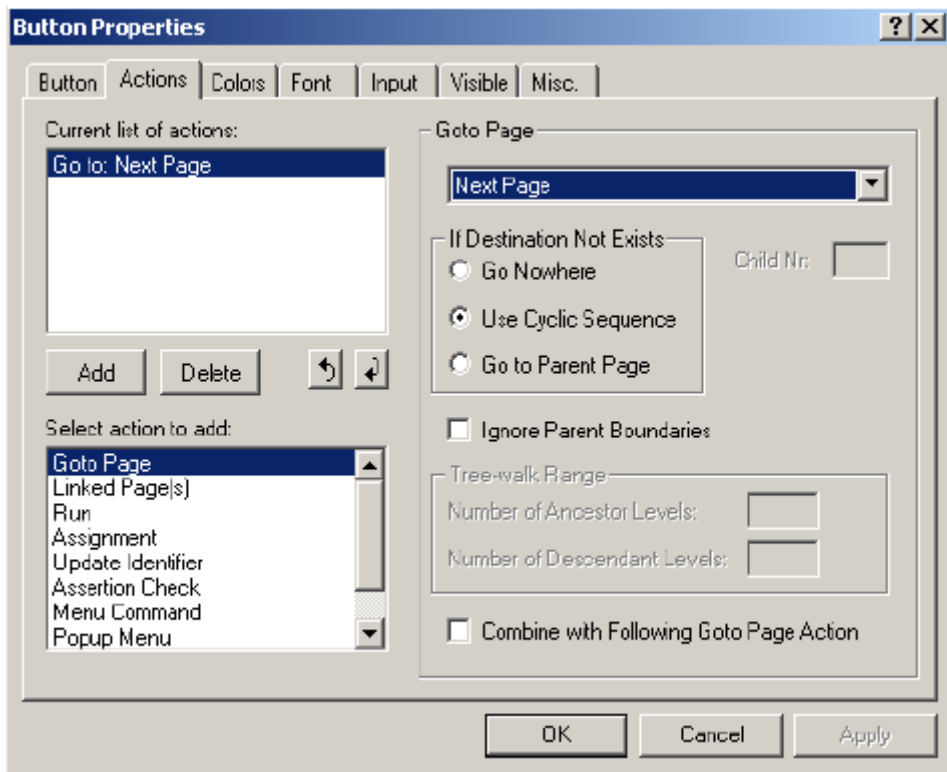


图 10.12: Button Properties 对话框



在您的屏幕上您应该看到一个按钮包含一个灰色的小盒子，点击工具栏左边的 Page User Mode 按钮 ，灰色盒子变成位图，并带有指向右边的箭头，再次点击工具栏左边 Page Edit Mode 按钮 ，回到目标编辑模式然后创建剩余两个按钮，如图 10.13 所示。左箭头按钮上的位图与文件 ‘Bitmaps\Button Prev.bmp’ 相对应，该按钮表示 ‘转到前一页’ 动作，剩下的按钮与文件 ‘Bitmaps\Button Up.bmp’ 相对应，表示 ‘Go to Parent Page’ 动作。与前文所述类似，仍然可以通过改变进入 User 模式检查这三个按钮。




图 10.13: 页面模板的三个按钮

10.3 目录页

在页面管理器中，从页面的层次结构二样，目录页是上层页面。利用该页面可以引用 5 个概述页面。为此，AIMMS 提供了所谓的导航对象，根据页面管理器的页面结构，可以动态的更改导航对象的目录。

您可以执行以下步骤，在目录页中创建一个新的导航对象：

- 打开 Contents 页面
- 确保页面处于 Edit 模式
- 点击工具栏中的 New Navigation Object 按钮 
- 使用鼠标在页面的中间拖拽出一个矩形，然后
- 点击 OK 按钮

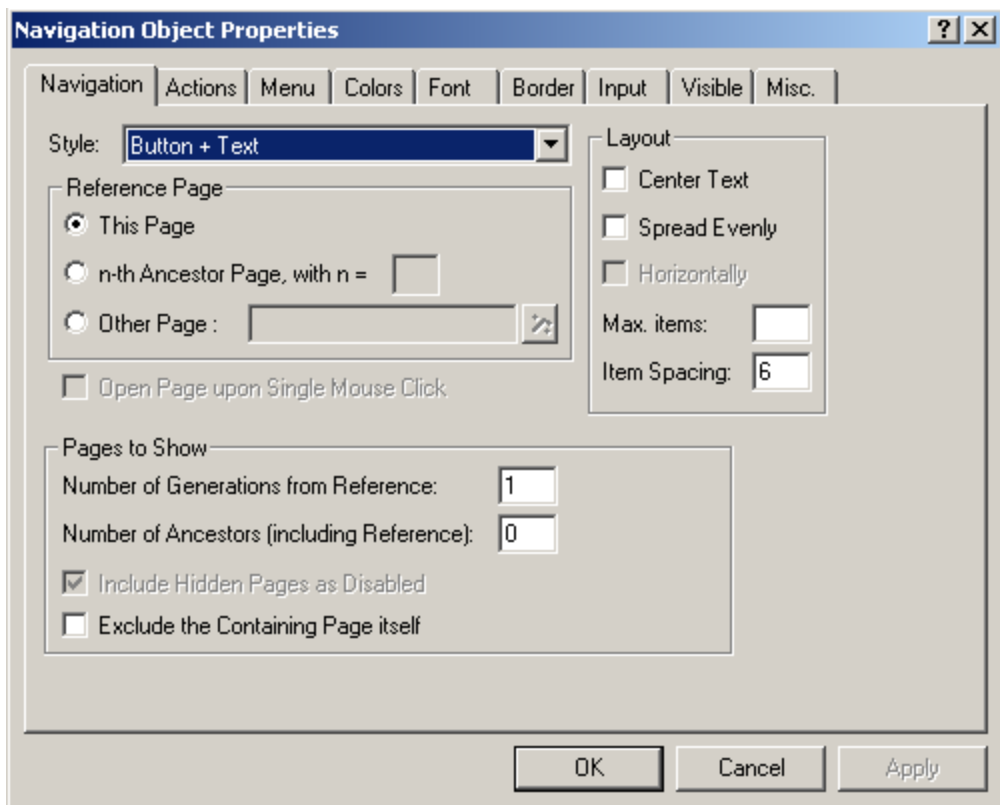



图 10.14: Navigation Object Properties 对话框

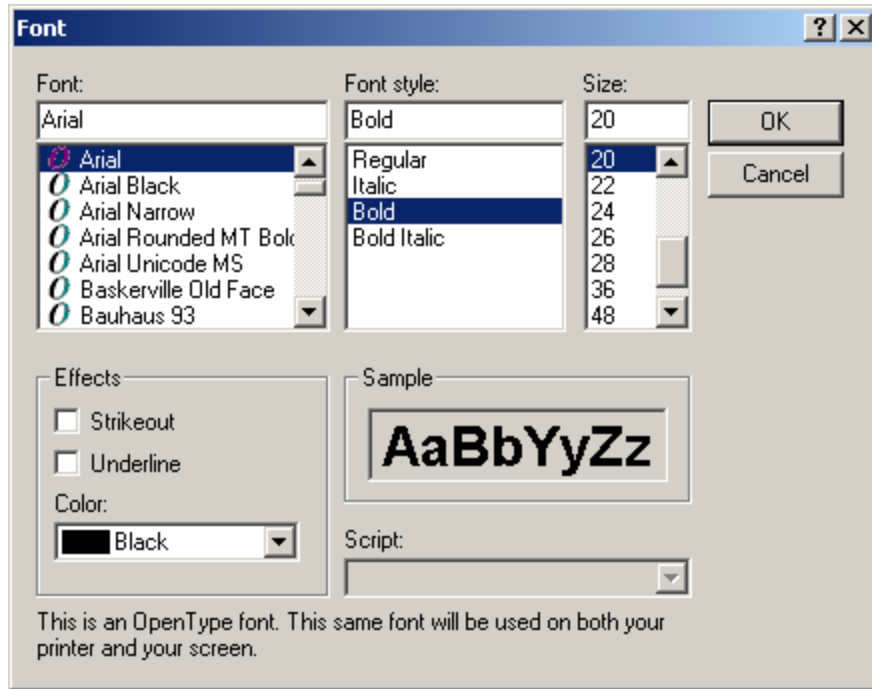
导航对象属性对话框的默认设置入托 10.14 所示，如果选择默认设置，仅当前参考页面的子页面将被显示。如果想调整导航对象的目录，可以通过改变参数“Number of Generations from Reference”或参数“Number of Ancestors（包括参考）”。

如果考虑到导航对象中默认的字体过小，如果想改变字体字号，请打开 Navigation Properties 对话框，右击鼠标选择‘Properties. . .’命令，或点击工具栏中‘Properties’按钮 。一旦进入对对话框，请按以下步骤执行：

- 选择 Font 键
- 点击 Add 按钮
- 在‘Font Style’中选择‘Bold’选项
- 在‘Font Size’中选择‘20’
- 点击 OK 按钮
- 指定‘Navigation Object Font’作为新字体名称，然后

- 按 OK 按钮

字体选择页面如图 10.15 所示, 屏幕中导航对象应是可视的。



10.15: Font 对话框

图 10.16 显示如何设置前景颜色为深蓝色, 请按以下步骤执行:

- 重新打开 Navigation Properties 对话框
- 选择 Colors 键
- 在背景颜色的下拉列表中选择 'Transparent'
- 选择 'User' 作为前景颜色提供者
- 设置前景颜色为深蓝色, 然后
- 按 OK 键

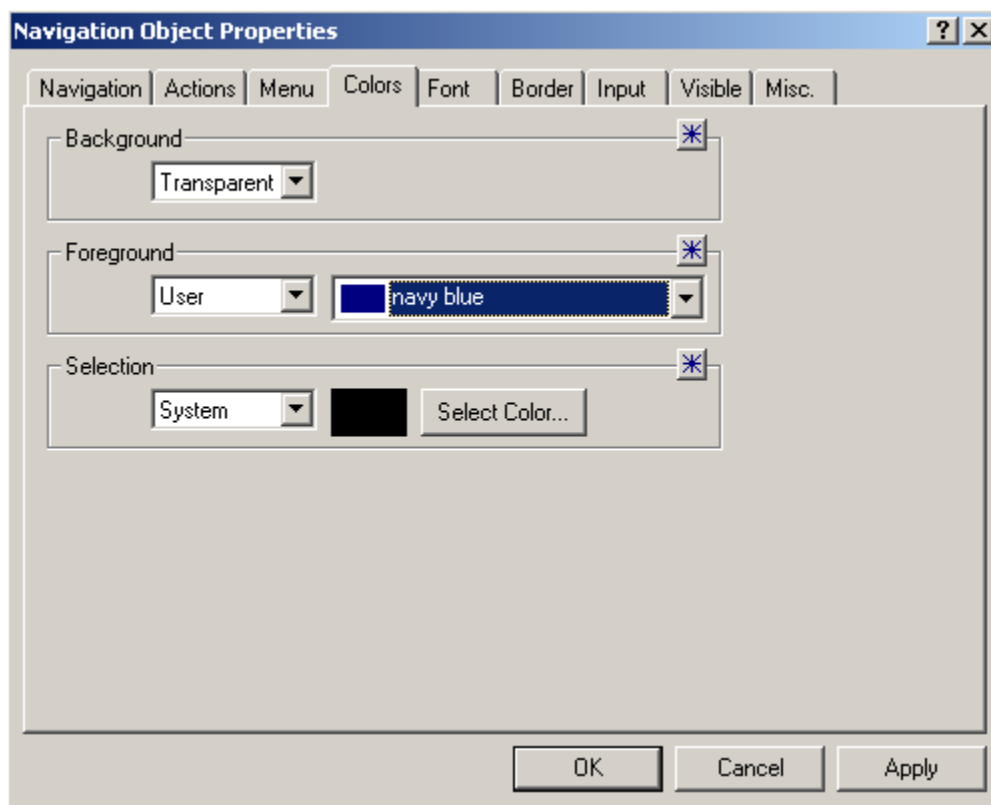


图 10.16: Navigation Properties 对话框中 Colors 项


在许多应用中,可能需要在页面上添加 LOGO,在本教程中,您可以按以下步骤使用 AIMMS 的 LOGO 功能。

- 在编辑模式中打开 Contents 页
- 在 Object 菜单栏中选择 Picture 命令
- 使用鼠标,在页面的右上角拖拽出一个矩形
- 在 'File Name' 编辑域的右边选择向导按钮 
- 在右击弹出菜单栏中执行 'Select File Name' 命令
- 在 Picture Properties 对话框中选择文件 'Bitmaps\AIMMS Logo.bmp'
- 点击 Open 按钮,回到 Picture Properties 对话框
- 按 OK 键

此时 Contents 页图像如图 10.17 所示:



图 10.17: Contents 页

只要点击了 Page User Mode 按钮 ，您就可以点击目录页中的创建的 5 个按钮，然后 AIMMS 会自动打开相应的子页面。您可以使用 ‘Previous’，‘Next’ 或 ‘Up’ 按钮进入另一页面。

在 AIMMS 中，您可以指定一个启动页面，当底层应用程序打开后，启动页面会自动显示。如果您想让目录页作为应用程序的默认启动页，请执行以下步骤：

- 从 Settings 菜单栏中选择 Project Options 命令
- 如图 10.18 所示，设置 ‘Startup page’，然后
- 按 OK 键

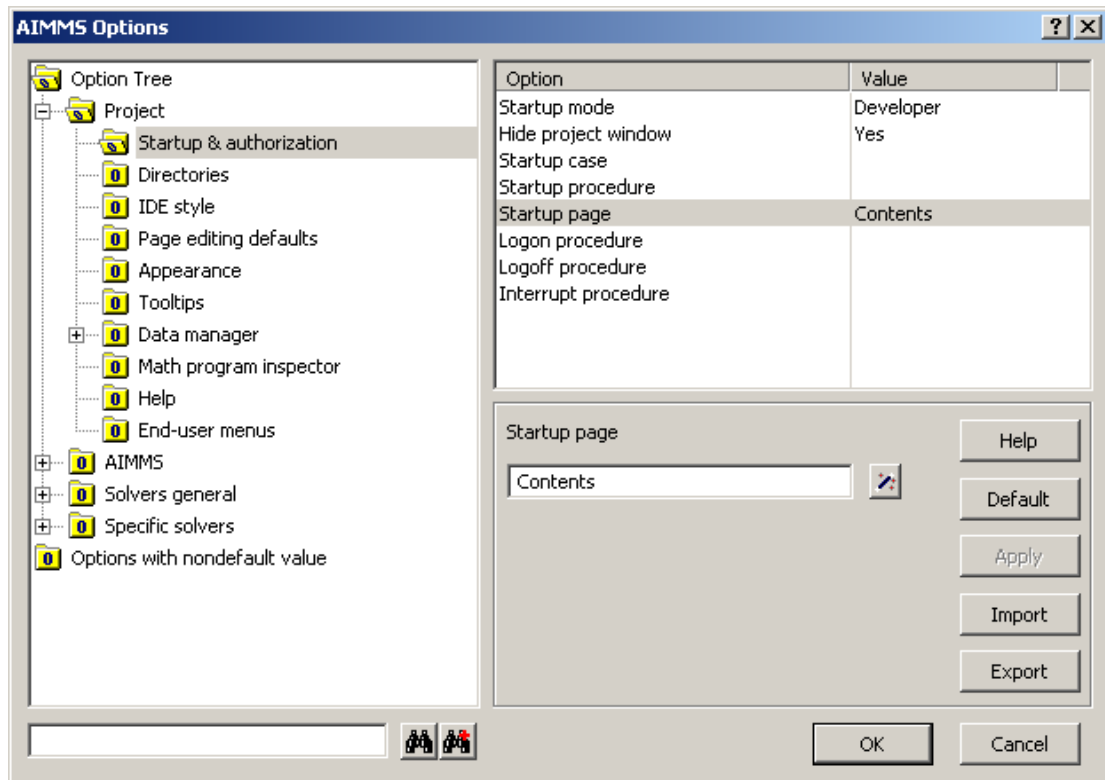



图 10.18: Aimms Options 对话框

AIMMS 窗口中标题栏的左上角星号表示最近项目改变的内容尚未保存到硬盘中，点击工具栏的保存项目按钮保存

保存项目之后关闭，然后重新打开该项目，确认 Contents 页是否自动显示，关闭和重启项目已经被这一过程在第九章结尾已详细讨论过。

生产和运输概述

本章将创建两个终端用户界面，以体现在一个滚动周期过程中，模型解的周期性变化。界面一是生产概述界面，里面主要有整个计划期间，每一阶段的最优的生产和维修时间表。界面二是运输概述界面，主要提供工厂到配送中心的最优运输模式，此外还包括每个位置的储存概述。

11.1 扩展模型树

建立专业的用户界面后，自然需要引进额外的标识符去支持此界面。例如：当某一数值低于门阙值时，定义在预定义集合‘AllColors’的元素参数可以用来改变数值的颜色，或者引入一个参数控制 Gantt 图的滚动机制，此外，也可以引入标识符控制某一对象是否出现，

对于 Page Manager 中已引入的 5 个终端用户概述页面，额外需要在模型树中引入相应的 5 个段落。引入的全部新的特殊页面标识符将被插入到合适的段落中，更新后的树结构如图 11.1 所示：

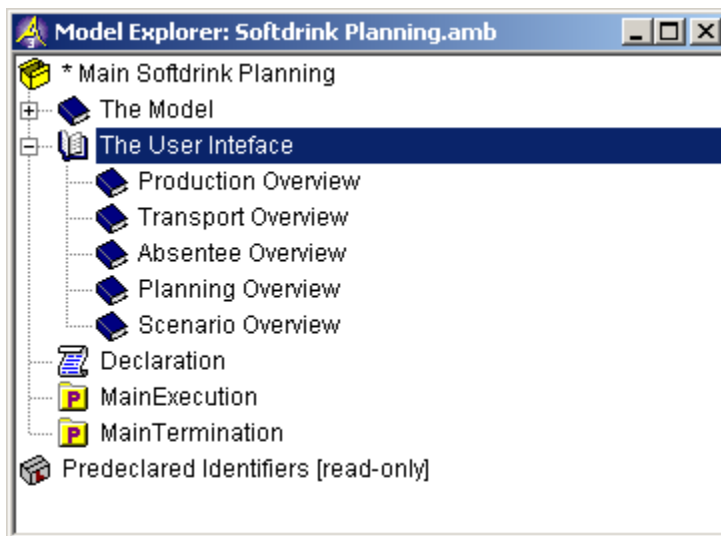


图 11.1：用户界面的细分图

11.2 生产概述页面

本节中，将构建的整个页面如图 11.2 所示，每个页面对象放在单独小节处理

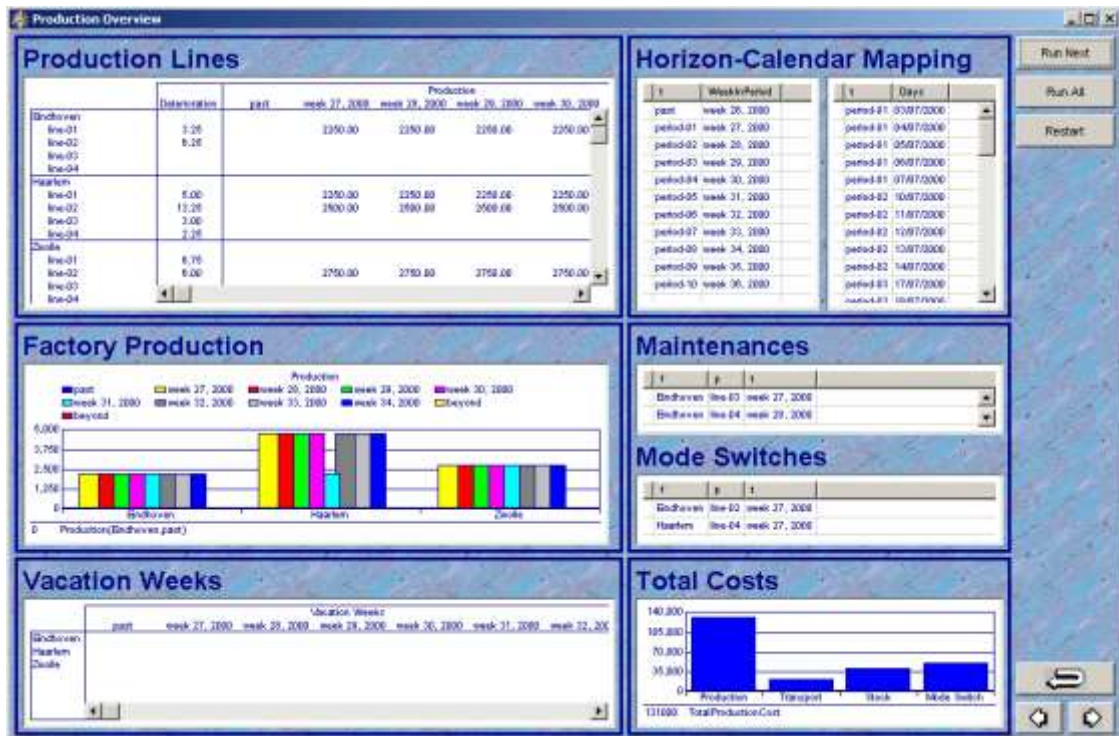




图 11.2 完成后的生产概述页面

11.2.1 执行按钮

增加第一个执行按钮，该按钮被设计用来执行周期滚动过程中的一个单步，可以允许逐步跟踪模型运作过程。请按以下步骤创建‘Run Next’按钮

- 在 Edit 模式打开 Production Overview 页面
- 点击工具栏中的按钮 New Button 
- 使用鼠标在页面的右上角拖拽出小的矩形
- 在‘Title’编辑区域中输入“Run Next”（有引号）
- 按下 Actions 选项
- 选择‘Run’选项
- 点击 Add 按钮
- 选择‘Procedure’选项（注意不是‘Page Procedure’选项）
- 使用向导按钮  选择程序‘RollHorizonOnce’
- 按下 Finish 按钮，然后
- 按 OK 键

增加第二个执行按钮，该按钮主要执行整个周期滚动过程中从当前点前进的动作，重复前文创建‘Run All’按钮的步骤，注意这里选择了程序‘RollHorizonToEnd’。

增加第三个执行按钮‘Restart’，该按钮具有激活程序‘MovePlanningInterval-ToStartOfCalendar’功能。在执行该程序之前，您可以使用前面两个执行按钮的一个区执行周期滚动过程的部分和全体。为方便起见，避免与前文一样重复过程去创建该按钮，您可以按下面步骤使用 copy and paste 功能：



- 在 Edit 模式中，选择 Run All 按钮并点击
- 点击工具栏中的复制按钮
- 点击工具栏中的黏贴按钮（鼠标光标将如图 11.3 发生改变）
- 使用鼠标光标移动新按钮位于 Run All 按钮之下
- 点击鼠标左键确定新按钮的位置
- 双击鼠标左键，打开新按钮的 Button Properties 对话框，然后
- 适当修改新按钮的性质



图 11.3 点击 Paste 按钮后鼠标光标变化

11.2.2 生产线表


生产概述页面的第一张表格包含三个标识符，分别是：

- 每个工厂的各条生产线在每个时期的实际生产水平
- 每周的工作日数量
- 当前每条生产线相应的恶化水平


当生产线运行时，其实际生产水平等于潜在生产水平，在生产概述段中创建子声明段 ‘Production Overview’，然后嵌入以下声明：

```
PARAMETER:
  identifier   : ActualProduction
  index domain : (f,p,t)
  unit        : h|
  definition   : PotentialProduction(f,p,t) * ProductionLineInUse(f,p,t)
```

按以下步骤创建表格的第一部分：

- 确保生产概述页面处于 Edit 模式
- 点击工具栏 New Table 按钮
- 使用鼠标在页面中拖拽去一个矩形，使其满足所期望的表格尺寸
- 在 Identifier 向导栏中选择参数 ActualProduction(f, p, t)
- 点击 Next 按钮，然后
- 按 Finish 按钮

按以下步骤在新表格的第一列添加标识符 DeteriorationLevel(f, p)

- 选择当前表格对象
- 点击工具栏的 Properties 按钮
- 选择 Contents 选项
- 按 Add 按钮
- 使用 Identifier 向导选择标识符 DeteriorationLevel(f, p)
- 按 Next 按钮
- 取消选中 “Automatic split row/column” 复选框
- 选择弹出的 “split line” 条目的列表框（如图 11.4 所示）

- 按 Down 按钮
- 按 Finish 按钮
- 按下 Up 按钮显示标识符 DeteriorationLevel 作为第一列，然后
- 按 OK 按钮

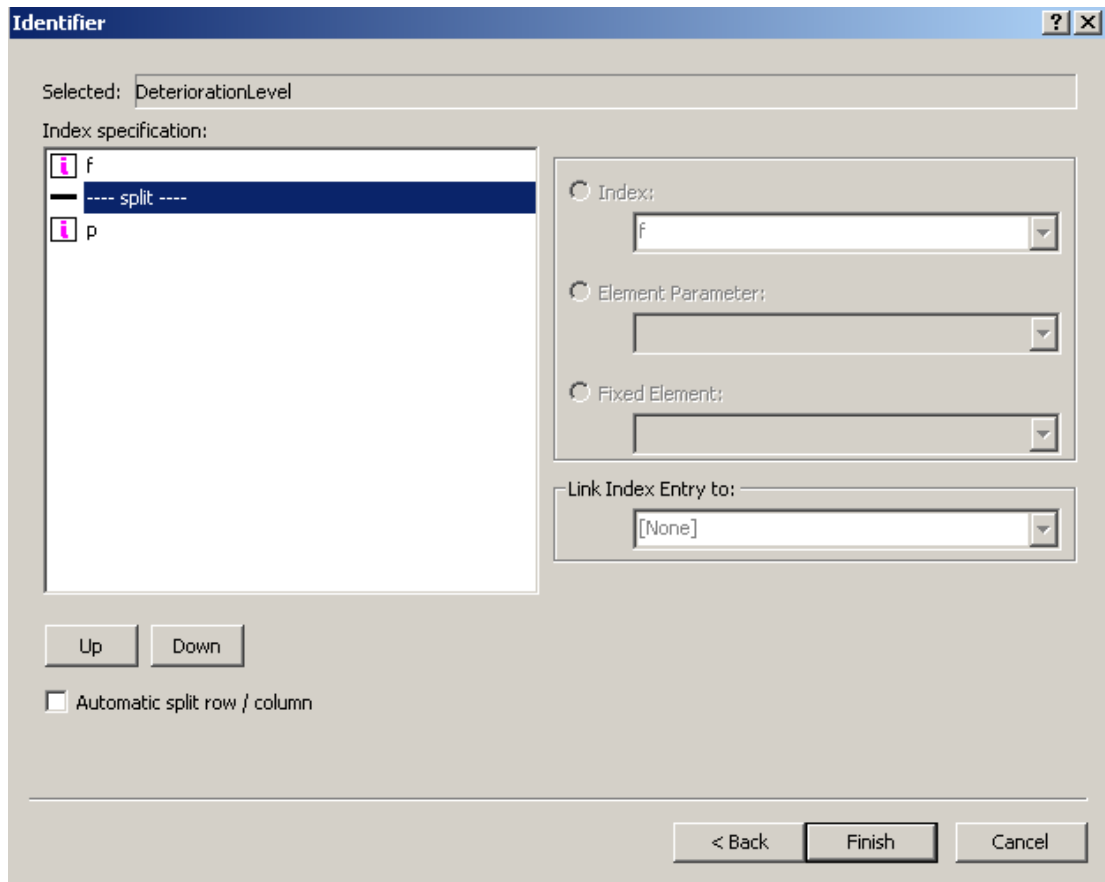


图 11.4: 指定行和列域

如果不移动分割线，AIMMS 将自动使用索引 f 为行，索引 p 为列。如果移动分隔符，则两个索引都可以作为行索引，如图 11.5 所示。

完成上面的程序后，现在在表格中添加标识符 ActualNumberOfDaysInPeriod(t) 作为新一行，此时屏幕中表格如图 11.5 所示：

	DeteriorationLevel	ActualProduction								
		part	period.01	period.02	period.03	period.04	period.05	period.06	period.07	period.08
Enschede										
Ine-01	2.3									
Ine-02	6.0									
Ine-03										
Ine-04										
Haarlem										
Ine-01	4.0									
Ine-02	12.3									
Ine-03	2.8									
Ine-04	2.0									
Zwolle										
Ine-01	6.5									
Ine-02	5.0									
Ine-03										
Ine-04										
ActualNumberOfDaysInPeriod										

图 11.5: 初始生产概述表

表中所引用的 ‘period’ 参数抽象且没有意义，在 AIMMS 中可以使用字符串参数改变时

间编号，请在段落 Production OverviewDeclarations 中创建下面字符串参数：

```
STRING PARAMETER:  
  identifier : PeriodDescription  
  index domain : t  
  definition : if ( t in Periods.past) then  
                "past"  
              elseif ( t in Periods.beyond ) then  
                "beyond"  
              else  
                FormatString("%e",WeekInPeriod(t))  
              endif
```

预定义函数 FormatString 允许创建一个字符串，该字符串是数字，字符串，几何元素的组合（具体参见语言参考第五章）

上文表格中字符串参数 PeriodDescription(t)可以用于元素文本，请按下列步骤执行：

- 打开表格中 Table Properties 对话框
- 选择 Element Text 选项（如图 1.6 所示）
- 选择索引 t
- 选择 Modify 按钮
- 选择标识符 PeriodDescription(t)
- 选择 Next 按钮
- 按 Finish 按钮，然后
- 按 OK 按钮

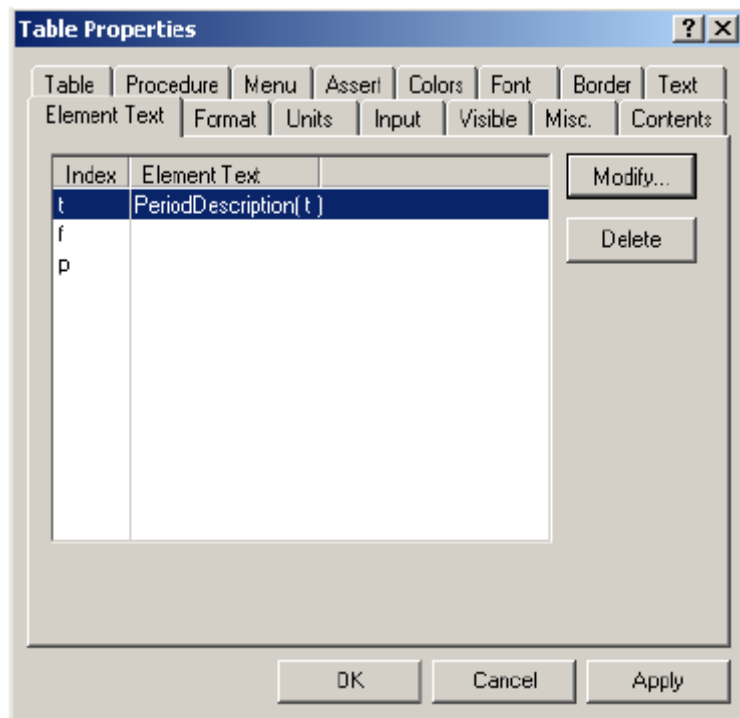


图 11.6: Table Properties 对话框中 Element Text

表格如果没有显示已创建的时间描述，并且收到一个初始化警告，此时您应该再次点击 Run Next 按钮，然后时间描述即出现。

显示表格中标识符时，AIMMS 会选择默认格式。然而如果想改变数字的位数/小数的位数。例如，参数 ActualNumberOfDaysInPeriod 是一整数，或者参数 ActualProduction 的数值相对于默认格式而言太大。

请按下列步骤将Actual-Production数字格式改为小数位数为2，字符宽度为8：

- 打开表格的Table Properties对话框
- 选择Format选项(如图11.7所示)
- 从下拉列表中选择元素ActualProduction(f, p, t)
- 在‘Width’中输入‘8’(没有引号)
- 在‘Decimals’中输入‘2’(没有引号)
- 按Apply按钮

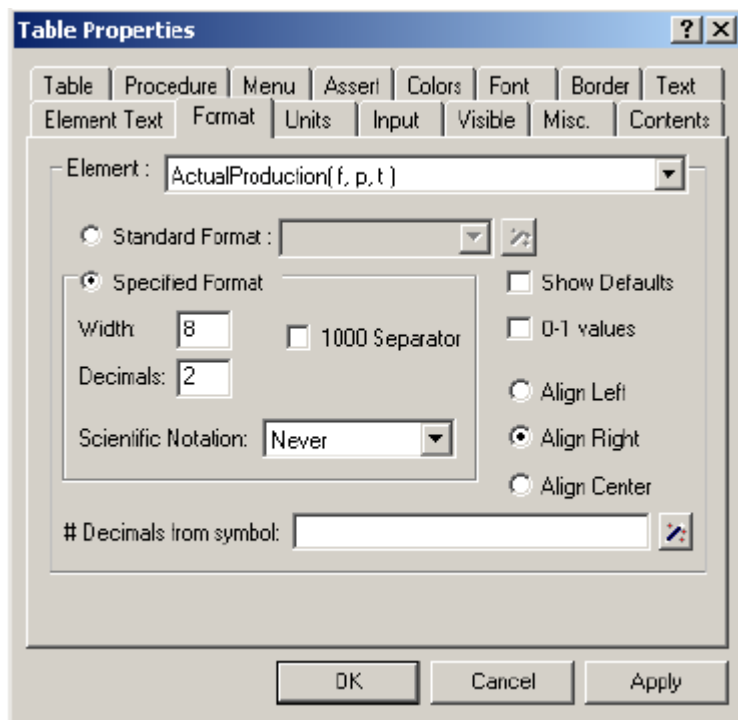


图 11.7：Table Properties 对话框中 Format 选项


接下来，参数 DeteriorationLevel 的格式改为小数位数为 2，宽度为 5。将参数 ActualNumberOfDaysInPeriod 的数字格式改为小数位数为 0，宽度为 5。完成之后表格如图 11.8 所示：

	Deterioration	Production					
		past	week 27, 2000	week 28, 2000	week 29, 2000	week 30, 2000	week 31, 2000
Endhoven							
line-01	3.25		2250.00	2250.00	2250.00	2250.00	2250.00
line-02	6.25						
line-03							
line-04							
Haarlem							
line-01	5.00		2250.00	2250.00	2250.00	2250.00	2250.00
line-02	13.25		2500.00	2500.00	2500.00	2500.00	
line-03	3.00						
line-04	2.25						
Zwolle							
line-01	6.75		2750.00	2750.00	2750.00	2750.00	2750.00
line-02	6.00						
line-03							
line-04							
Number of Working Days			5	5	5	5	5

图 11.8: 完成后的生产线表格

11.2.3 工厂生产条形图

生产线图标显示每条生产线的生产概述。下条形图提供了工厂生产水平的类似概述。请按下列步骤创建该条形图：

- 确认 Production Overview 页面在 Edit 模式中打开
- 点击工具栏中 New Bar Chart 按钮 
- 使用鼠标拖拽去与生产线表格相同宽度的矩形，然后
- 使用 Identifier 向导选择变量 Production (f, t)

如前面一样，使用字符串参数 PeriodDescription 将抽象的时间改为周，由此产生的条形图如 11.9 所示：

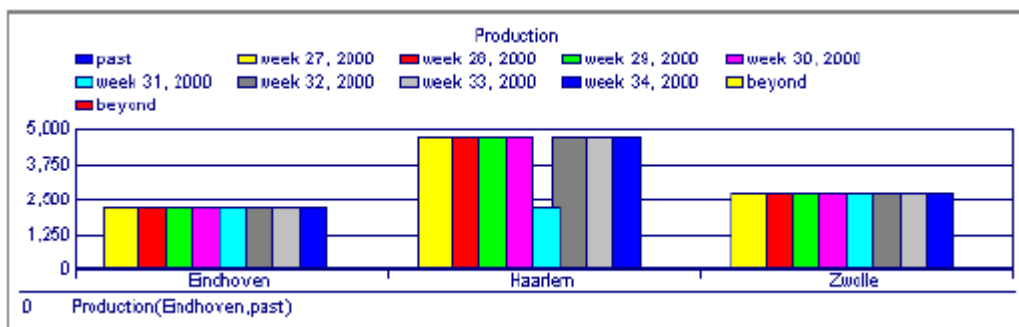



图 11.9 完成后的工厂生产条形图

1.2.4 假期图

创建的表格将显示休假期对应的全部周数，在这期间生产水平下降 40%。请按下列步骤顺序创建表格。

- 确保 Production Overview 页面处于 Edit 模式

- 点击工具栏中 New Table 按钮
- 使用鼠标拖拽去一个与工厂生产柱形图相同宽度的矩形
- 使用 Identifier 向导选择参数 IsVacationPeriod(f, t)，然后
- 将索引 t 的元素文本改为字符串参数 PeriodDescription(t)

标识符 IsVacationPeriod(f, t) 是一个二元参数，其中 0 值意味着没有假期。数值 1 表示一个假期时间。其中数值是否选择 1 是随意的，因此你也可以用交叉十字代替 1，可按如下方式完成修改：

- 打开表格 Table Properties 对话框
- 选择 Format 格式选项（如图 11.10 所示）
- 检查 ‘0 - 1 values’ 复选框，然后
- 点击 OK 按钮

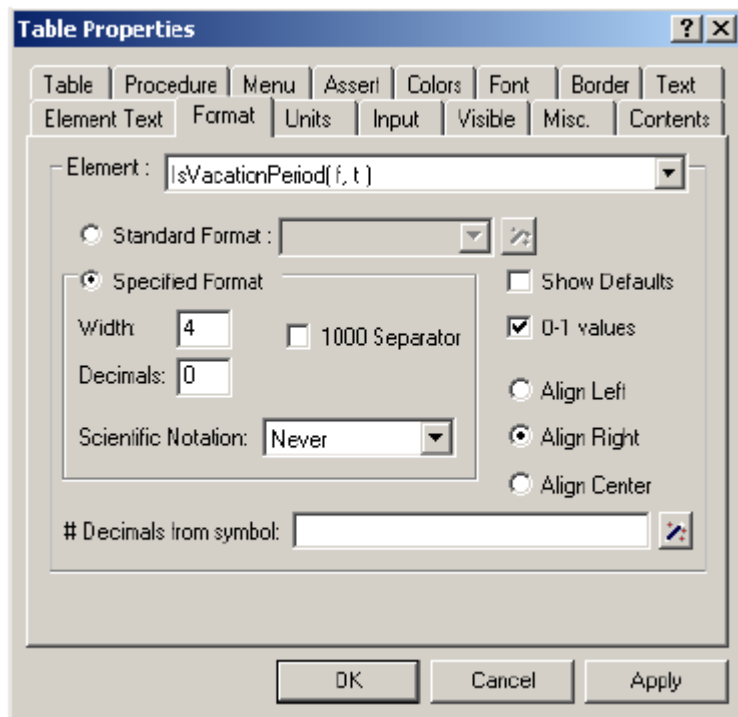



图 11.10: Table Properties 对话框中 Format 选项

注意, 此时由于假期 ‘周’ 尚未指定, 表格显示空白, 最后使用 Absentee Overview 页中 Gantt 对象指定假期周。

11.2.5 horizon-calendar 表

在本段中, 创建两个综合表建立抽象的层级时间与每周或每日的日历时间之间关系。AIMMS 中综合表 and 数据库表有类似的结构关系。你可以在图像界面的内部调整列的宽度, 请按以下步骤创建第一个综合表

- 点击工具栏中的 ‘New Composite Table’ 按钮
- 在页面中拖拽一个矩形
- 选择参数 WeekInPeriod(t)
- 点击 Next 按钮，然后
- 按 Finish 按钮

对于第二个综合表，请选择索引集 DaysInPeriod(t)。两个综合表格图像如图11.11所示：

t	WeekInPeriod	t	Days
past	week 26, 2000	period-01	03/07/2000
period-01	week 27, 2000	period-01	04/07/2000
period-02	week 28, 2000	period-01	05/07/2000
period-03	week 29, 2000	period-01	06/07/2000
period-04	week 30, 2000	period-01	07/07/2000
period-05	week 31, 2000	period-02	10/07/2000
period-06	week 32, 2000	period-02	11/07/2000
period-07	week 33, 2000	period-02	12/07/2000
period-08	week 34, 2000	period-02	13/07/2000
period-09	week 35, 2000	period-02	14/07/2000
period-10	week 36, 2000	period-03	17/07/2000
		period-03	18/07/2000
		period-03	19/07/2000

图11.11 层级和日历之间的映射

11.2.6 维修和模式切换表

与节假日类似，维修期间同样会造成生产效率降低。因此，维修概述同样对生产线表格和工厂生产柱形图的结果产生影响。从现在起，无需向你即可创建维修表，此表以标识符 LineInMaintenance(f, p, t) 作为其定义域，表格 f, p, t 作为其所要求的三列。完成该表之前你需要再次修改抽象的时间参考，

具体通过指定字符串参数 PeriodDescription(t) 作为索引 t 的元素文本（正如您之前所做）生产概述页面中最后一个综合表将显示当前规划周期的最优切换模式。他可以按照上面表格的同样方式进行指定，此时标识符 ProductionLineLevelChange(f, p, t) 被指定为表格的定义域。综合表如图 11.12 所示：

f	p	t	f	p	t
Haarlem	line-02	week 31, 2000	Eindhoven	line-02	week 27, 2000
			Haarlem	line-04	week 27, 2000

图 11.12：维修表（左）和模式切换表（右）

11.2.7 总成本条形图

本页中最后一个数据对象将显示聚合的四个成本分量，他们共同决定了总成本。此时，尚没有包含这四个分量值得标识符，因此第一步需要声明 4 个新参数，汇总一起去描述生产，运输，储存和模式切换成本，这些参数被放在 Production Overview 段的结尾，注意汇总中的运输和储存成本是预估成本。

PARAMETER:

identifier : TotalProductionCost

unit : \$

definition : $\text{sum}[(f, t), \text{UnitProductionCost}(f) * \text{Production}(f, t)]$

PARAMETER:

identifier : TotalTransportCost

unit : \$

definition : $\text{sum}[(f, c, t, s), \text{Probability}(s) * \text{UnitTransportCost}(f, c) * \text{Transport}(f, c, t, s)]$

PARAMETER:

identifier : TotalStockCost

unit : \$

definition : $\text{sum}[(l, t, s), \text{Probability}(s) * \text{UnitStockCost}(l) * \text{Stock}(l, t, s)]$

PARAMETER:

identifier : TotalModeSwitchCost

unit : \$

definition : $\text{sum}[(f, p, t), \text{FixedCostDueToLevelChange} * \text{ProductionLineLevelChange}(f, p, t)]$

完成上面标识符声明后，现在创建一个条形图作为第一个标识符 TotalProductionCost。然后打开 Bar Chart Properties 对话框，并使用 Contents 选项添加剩余的三个标识符（如图 11.3），最后忽略初始化警告。

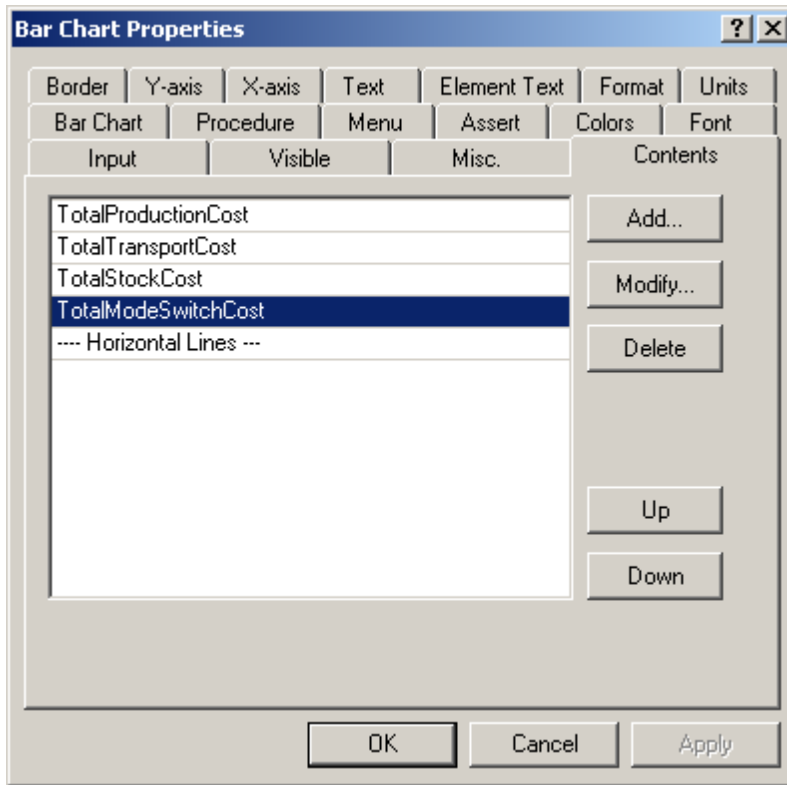


图 11.13: Bar Chart Properties 对话框中 Contents 选项

完成后的总成本条形图应如图 11.14 所示:

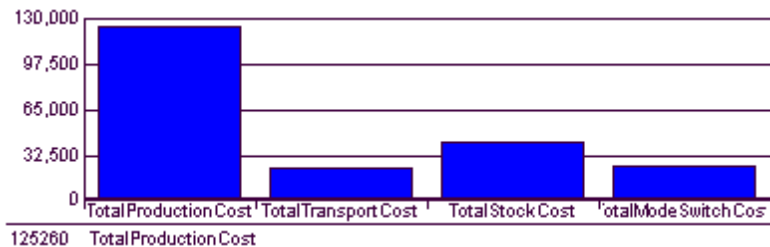



图 11.14: 完成后的总成本条形图

11.2.8 完成页面

如果想在页面中显示更多信息，一种方法是降低字体大小。你可以按以下步骤创建小的字体格式:

- 确保 Production Overview 概述页面处于编辑模式
- 选择表格，然后
- 按住 shift 键同时选择剩下的 7 个表格和条形图
- 点击工具栏中 ‘Production Overview’ 按钮 
- 选择 Font 选项，然后
- 选择 Add 选项
- 在 ‘Font Size’ 中输入 ‘7’ (如图 11.15 所示)

- 按 OK 键
- 输入 ‘Data Font’ 作为新字体名称，然后
- 两次点击 ‘OK 按钮

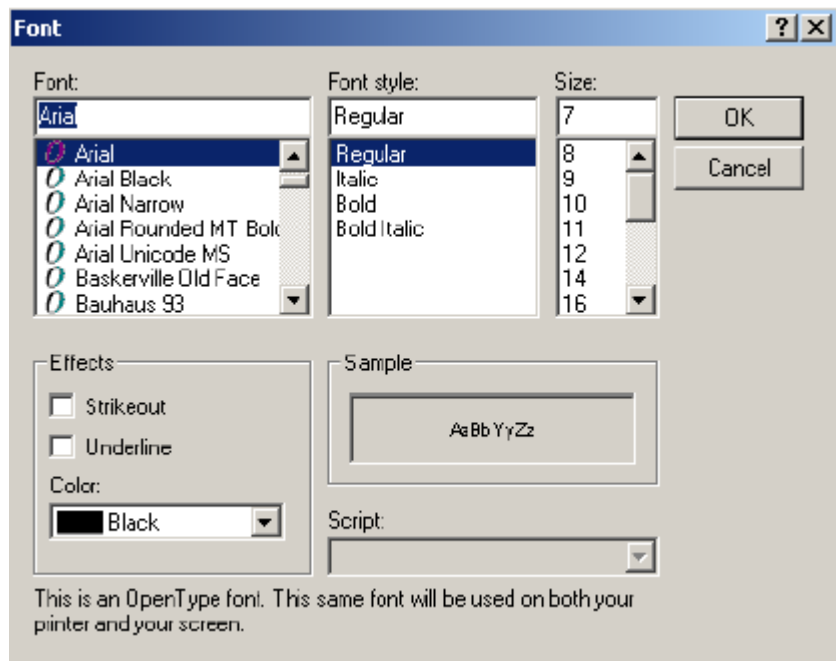


图 11.15:新字体编辑

这些表格，条形图和综合表都已放在 Production Overview 页面中，想完成页面，首先调整并使页面对齐，从而使创建的页面结构清晰且富有吸引力。为此，在 AIMMS ‘Edit’ 模式的 ‘Alignment’ 子菜单中提供一些对齐工具，并支持以下对齐选项：

- 调整对象位于左, 右, 顶部或底部
- 调整对象使其横向中心对齐或纵向中心对齐
- 调整对象使其横向展开或纵向展开
- 调整对象的长度和宽度

现按照前面所述，通过使用对齐工具调整全部页面对象，记住如果想选择多个页面，持续按住 shift 键。

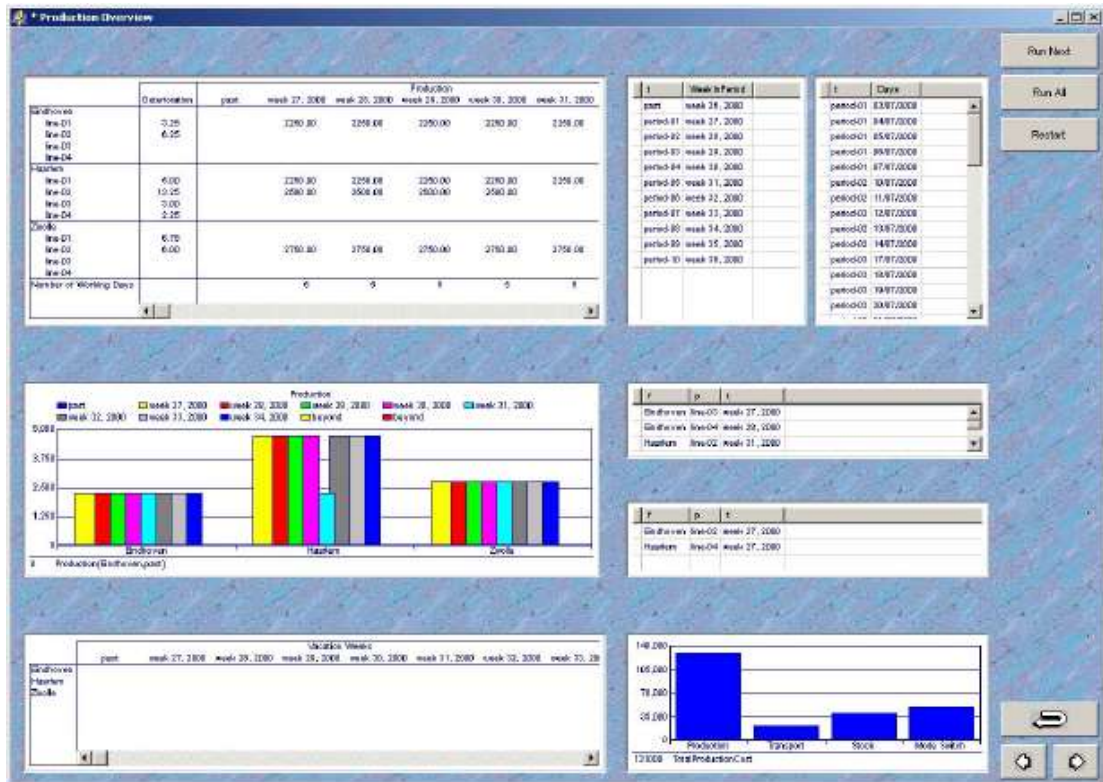


图 11.16: Production Overview 页面中对齐对象

如果想添加文本方便应用程序的终端用户,本段中你将创建一个文本对象,在下一段落,调整与文本相关的字体。首先考虑左上角的 production line 表格,请按以下步骤添加一行文本:

- 首先确保 Production Overview 处于编辑模式
- 从 Object 目录中选择 Text 命令
- 在生产线表格上方画一个矩形
- 在编辑区域输入 ‘Production Lines’ (不加引号) (如图 11.17 所示), 然后
- 按 OK 键

在本章节开始, 创建 6 个文如图 11.12 所示的文本对象。

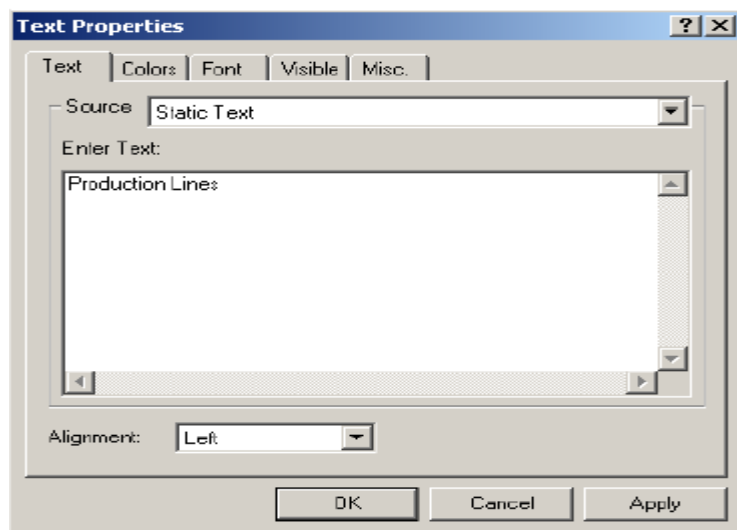


图 11.17: ‘Text Properties’ 对话框中 ‘Text’ 选项


如果想改变上文所提的文本对象的字体大小，首先按住 Shift 键选择全部对象，然后创建一个新的 18 号字体，并命名 ‘Title Font’，最后，文本对象仍如图 11.2 所示。

如果想更进一步提高页面的结构性，你可以在矩形中附上一个或多个页面对象，具体步骤如下：

- 首先确保 Production Overview 处于编辑模式
- 从 Object 目录中选择 Rectangle 命令
- 围绕页面中目标绘制矩形

然后再次修改结构，使这 6 个矩形如图 11.2 所示。

如果想扩大矩形边框的厚度，请执行以下步骤：

- 首先确保 Production Overview 处于编辑模式
- 按住 shift 键选择全部矩形
- 点击工具栏中 Properties 按钮
- 如图 11.18 所示，完成 ‘Rectangle Properties’ 对话框中 ‘Rectangle’ 选项，然后
- 按 OK 按钮结束

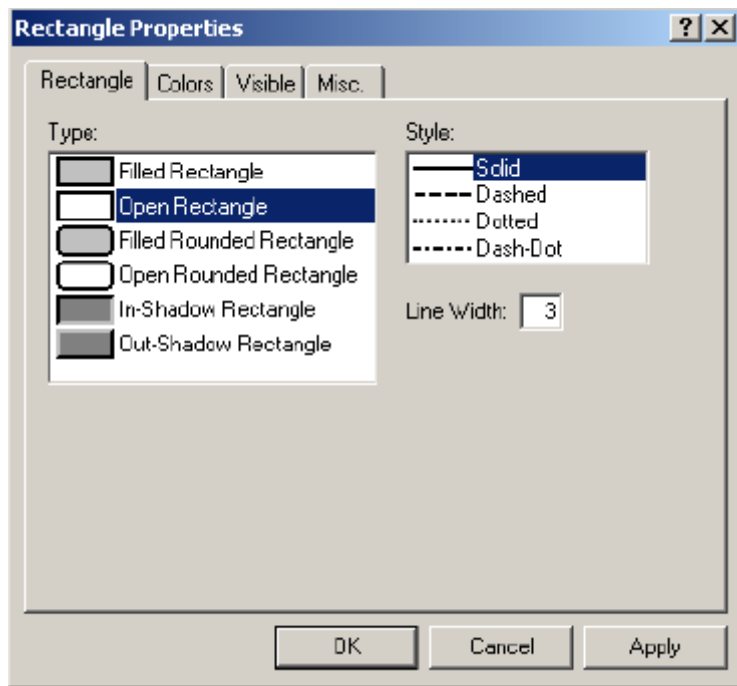



图 11.18: Rectangle Properties 对话框中 Rectangle 选项

如果想修改页面所有对象的默认颜色，由黑色变成海军蓝，请按以下步骤执行：

- 首先确保 Production Overview 处于编辑模式
- 按 Ctrl-A 组合键选择全部对象
- 使用 Shift 键取消选择三个执行按钮
- 点击工具栏 Properties 按钮
- 选择 Colors 选项
- 选择 ‘User’ 作为 ‘Foreground’ 颜色的使用者
- 从下拉列表中选择颜色 ‘Navy Blue’，然后
- 按 OK 按钮结束

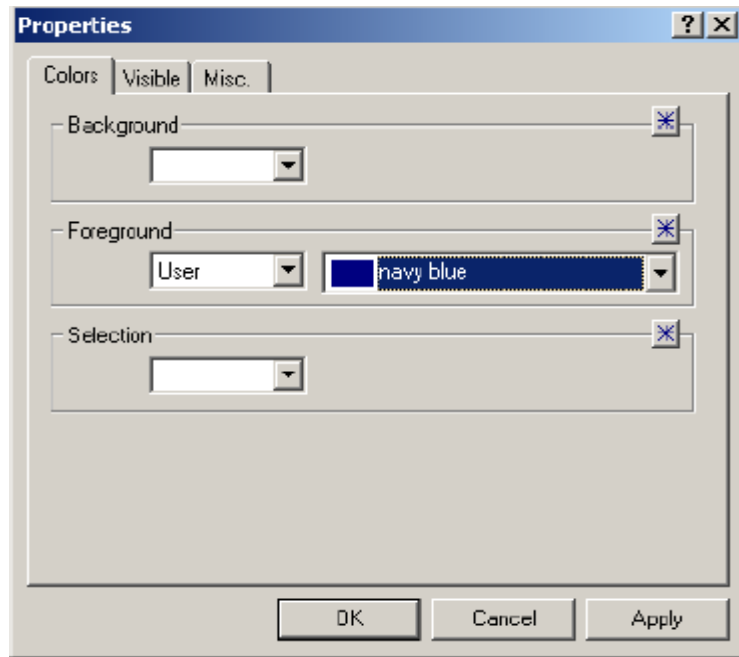


图 11.19 Properties 对话框中 Colors 选项

数据对象中会默认显示字符串名称，如果想修改面向终端用户的名称，你可以换成更合适的字符串，甚至换成字符串参数，以满足终端用户不同语言环境的需要。作为例证，请按下列步骤将默认的标识符‘ActualNumberOfDaysInPeriod’改为字符串‘Number of working days’：

- 选择生产线图表
- 打开 Table Properties 对话框
- 选择 Text 选项
- 选择标识符 ActualNumberOfDaysInPeriod(t)
- 在‘Title’的下拉清单中选择‘Other’
- 指“Number of working days”（加引号）作为新标题（如图 11.20 所示），然后
- 按 OK 按钮

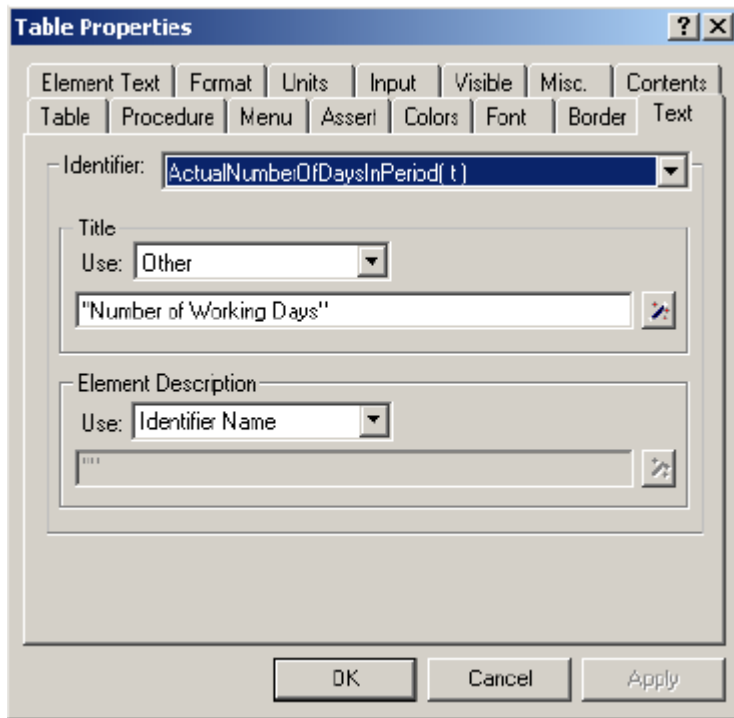


图 11.20: Table Properties 对话框中 Text 选项

AIMMS 甚至可以修改表格中每个数据分量的颜色，例如，你希望恶化率数值达到最大值时，其颜色从蓝色变成红色。为实现这一目标。首先创建一个颜色参数，该参数定义在 AIMMS 的集合 ‘AllColors’ 中的一个元素参数。可以通过 Tools 菜单栏中 User Colors 命令检查或修改该集合内容。

例如，在 Production Overview Declarations 段中声明下列颜色参数

ELEMENT PARAMETER:

identifier : DeteriorationColor

index domain : (f,p) | p in FactoryProductionLines(f)

range : AllColors


definition : if (DeteriorationLevel(f,p) > MaximumDeteriorationLevel(f,p)) then
' red'

else

' navy blue'

Endif

为了建立颜色参数与表中数据的连接，请执行以下步骤：

- 打开生产线表格的 Table Properties 对话框
- 选择 Colors 选项
- 在 ‘Identifier’ 段落中选择标识符 DeteriorationLevel(f,p)（最底部）
- 在 color determiner 中选择 ‘Model’
- 使用向导按钮 （如图 11.21）选择标识符 DeteriorationColor(f,p)，然后
- 按 OK 键结束

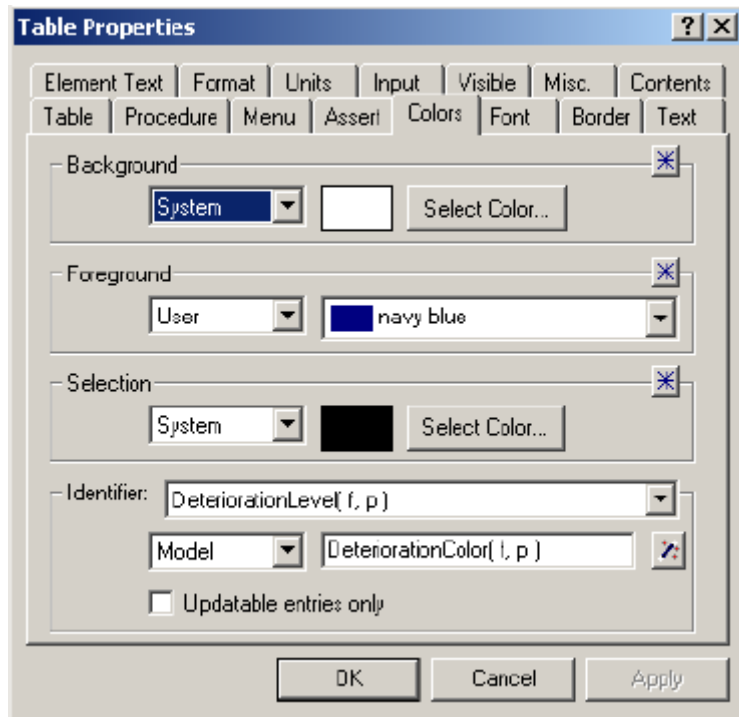


图 11.21: Table properties 对话框中 Colors 选项

完成后的 Production Overview 页面再次出现, 如图 11.22 所示, 你可以与你屏幕中创建的内容进行比较。

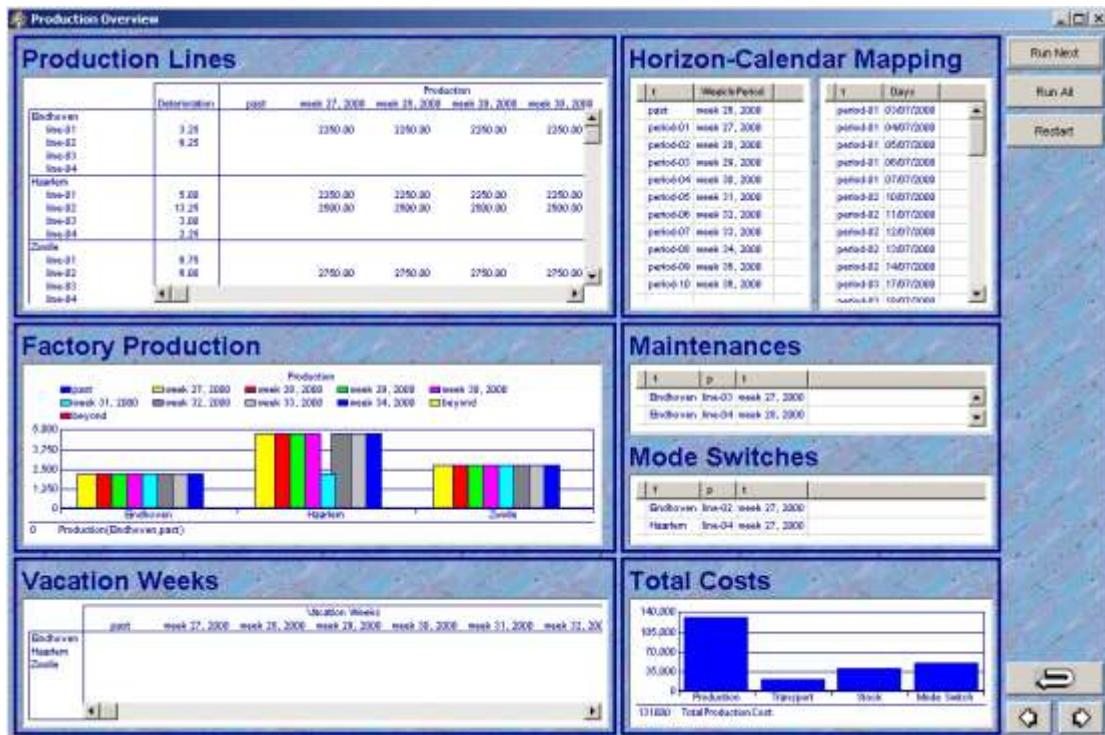


图 11.22 完成后的 Production Overview 页面

11.3 运输概述页面

本节中，你将创建的整个 Transport Overview 页面如图 11.23 所示，每个页面对象放在单独小节处理。

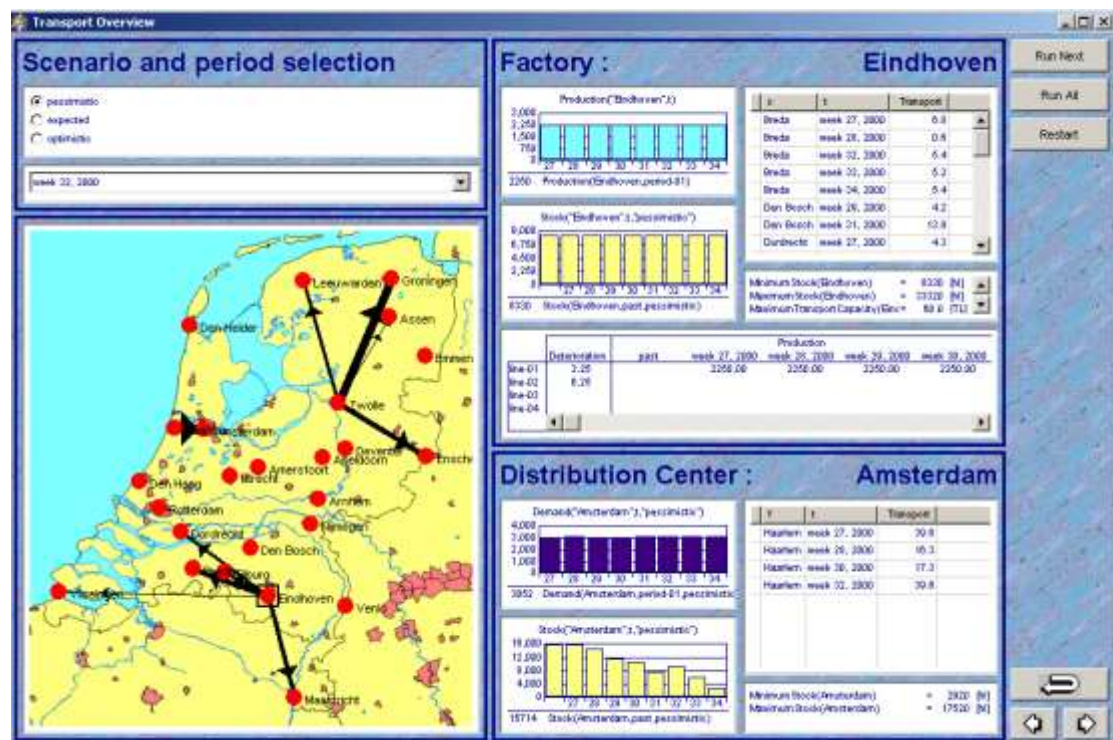


图 11.23: 完成后的 Transport Overview 页面

11.3.1 情景选择对象



针对不同情况，标识符 Transport and Stock 数值是不同的，在单一页面显示所有情况的数值会超出页面范围。因此只显示一种情况的信息，用户可以通过情况转化方式得到其他情况的信息。AIMMS 通过提供 ‘selection object’ 实现这一目标。首先在模型段 Transport Overview 中创建一个新的声明段 Transport Overview Declarations，该声明包含下面元素参数：

ELEMENT PARAMETER:

identifier : DisplayedScenario

range : Scenarios

按照下面步骤，连接其与选择对象从而决定元素参数的数值：

- 在编辑模式中打开 Transport Overview 页面
- 点击工具栏中 New Selection Object 按钮
- 使用鼠标在页面左上角拖拽出一个小矩形
- 从 ‘Single Item Selection’ 选项中选择 ‘Radio Buttons’
- 选择 ‘Element Parameter’ 作为 ‘Type of Data’
- 点击 ‘Element’ 域右边的向导按钮（如图 11.24 所示）
- 选择元素参数 DisplayedScenario
- 按 Finish 按钮，然后
- 按 OK 按钮

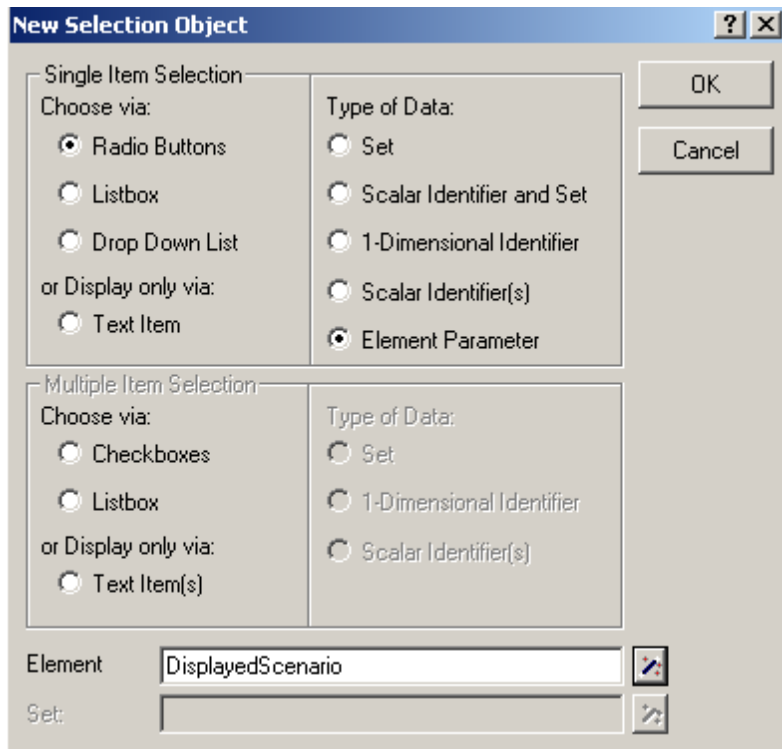


图 11.24 New Selection Object 对话框

你已创建的选择对象如图 11.25 所示，选择其中一个单选按钮将设置参数 DisplayedScenario 相应值。正如本节最后你将看到，其他页面对象将定义在该参数之上他们的数值也会随之相应调整。

- pessimistic
- expected
- optimistic

图 11.24: scenario 选择对象

11.3.2 期间选择对象

与元素参数DisplayedScenario类似，引入另一元素参数支持某一期间的选择。请在段落Transport Overview 结尾，声明下面元素参数：

ELEMENT PARAMETER:

identifier : DisplayedPeriod

range : Periods

创建该选择对象，将其元素参数设置为 Displayed-Period。注意选择 ‘Drop Down List’ 选项而非 ‘Radio Buttons’ 选项（如图 11.26 所示）

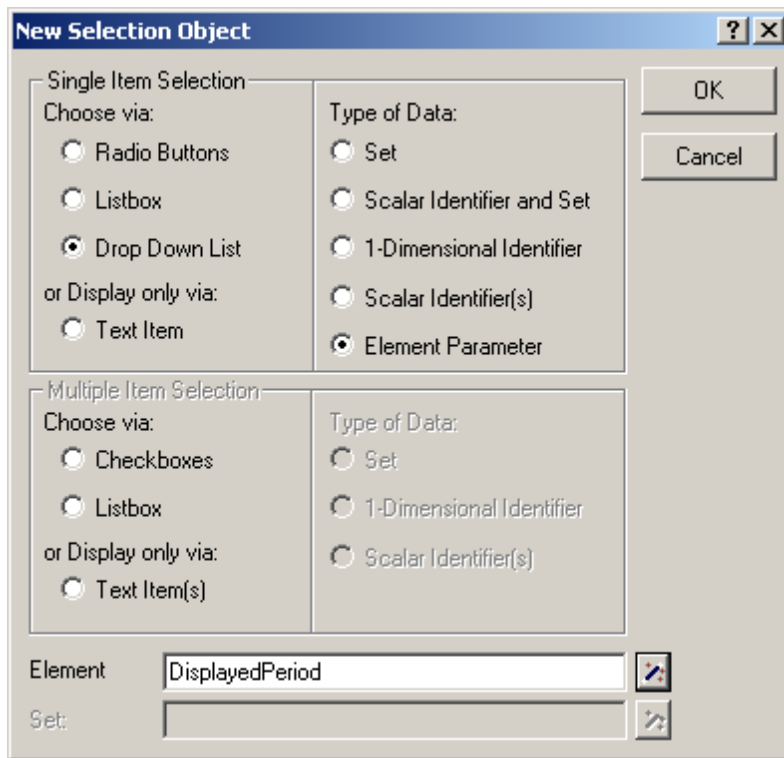


图 11.26: New Selection Object 对话框

创建下拉列表后，打开 Selection Object Properties 对话框（双击鼠标或者右击弹出菜单打开），将元素文本从抽象的时间改为指定的周，你可以通过选择 Element Text 选项完成这一修改，最后指定字符串参数 PeriodDescription(t) 作为索引 Periods 的元素文本。

为初始化 DisplayedScenario 和 DisplayedPeriod 两元素参数，暂时将页面调整为 User 模式，并通过它们选择 ‘optimistic’ 作为显示情况，选择 ‘week27, 2000’ 作为显示日期。

11.3.3 运输网络对象

运输页面中第三个需要创建的是网络对象，它将针对计划周期内的任一给定时间和场景，给出最优运输。我们在第五章已创建显示所有位置的网络对象，利用它创建一个新的网络对象。从 Locations 页面中复制该网络到 Transport Overview 页面，具体操作请按以下步骤执行：

- 在 Edit 模式中打开 Locations 和 Transport Overview 页面
- 选择 Locations 页面选项
- 在 Locations 页面中选择网络对象
- 点击工具栏中 Copy 按钮 
- 点击右上角的交叉十字  关闭该页面
- 选择 Transport Overview 页面选项
- 点击 Paste 按钮 
- 将该网络对象放在选择对象下方，然后
- 点击鼠标左键

第五章所见的网络对象仅显示全部位置，如果想在网络对象中添加弧线表示工厂和配送中心的最优运输，请按下面步骤执行：

- 在编辑模式中选择网络对象
- 打开 Network Object Properties 对话框
- 选项 Contents 选项
- 从列表中选择 ‘----- Arcs -----’ 元素
- 点击 Add 按钮
- 选择变量 Transport (f, c, t, s)，然后
- 点击 Next 按钮

接下来，分别指定索引 T 为元素参数 DisplayedPeriod 值，索引 S 为元素参数 DisplayedScenario 值

- 从列表中选择索引 t
- 选择 ‘Element Parameter’ 单选按钮
- 从下拉列表中选择元素参数 DisplayedPeriod
- 从列表中选择索引 s
- 选择 ‘Element Parameter’ 单选按钮
- 从下拉列表中选择元素参数 DisplayedScenario
- 点击 Finish 按钮（如图 11.27 所示），然后
- 按 OK 按钮

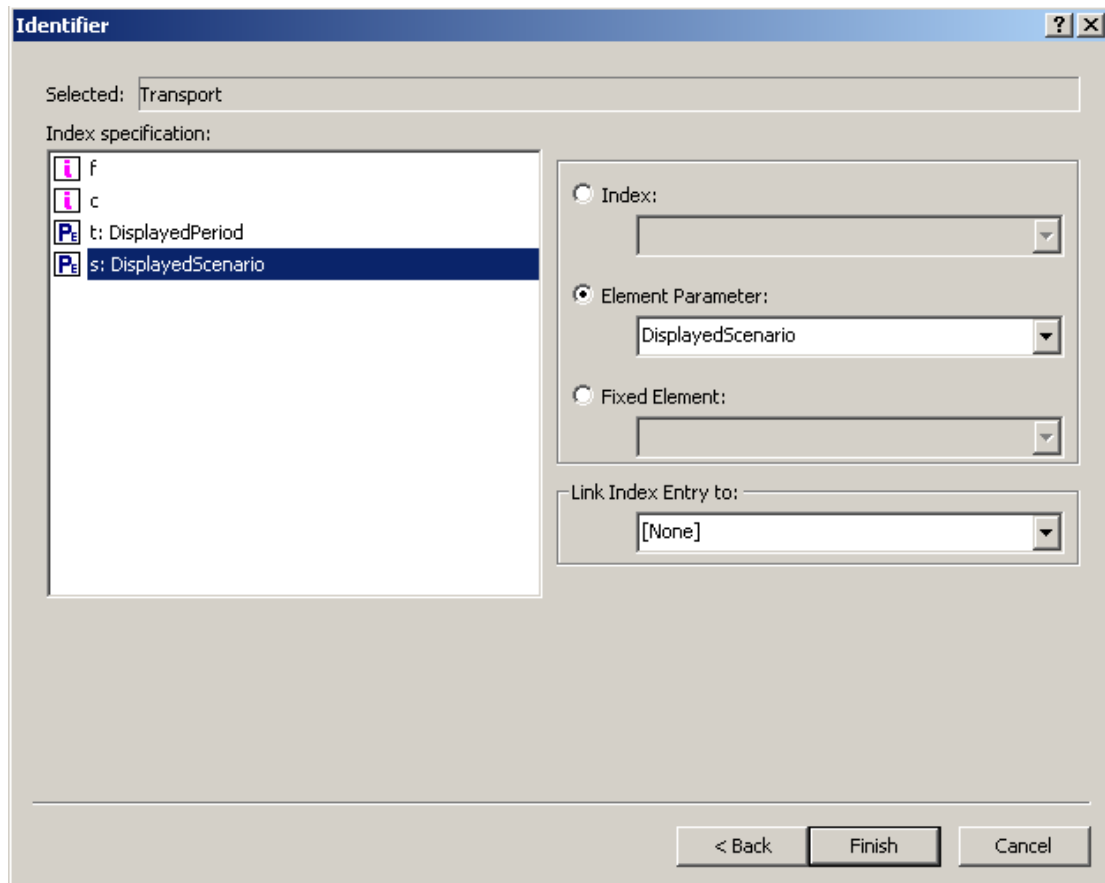



图 11.27：运输变量的修正索引

网络对象将显示全部运输量非零的弧线。为区分运输量的大小，将通过运输量的大小改变弧线的粗细，你可以执行下列步骤实现这一目标：

- 在编辑模式中选择网络对象
- 打开 Network Object Properties 对话框
- 选择 Arcs 选项
- 点击 ‘Size’ 域右边的向导按钮 
- 选择标识符 Transport (f, c, t, s)
- 按 Next 按钮
- 连接索引 t 和元素参数 DisplayedPeriod
- 连接索引 s 和元素参数 DisplayedScenario
- 按 Finish 按钮
- 按 OK 按钮

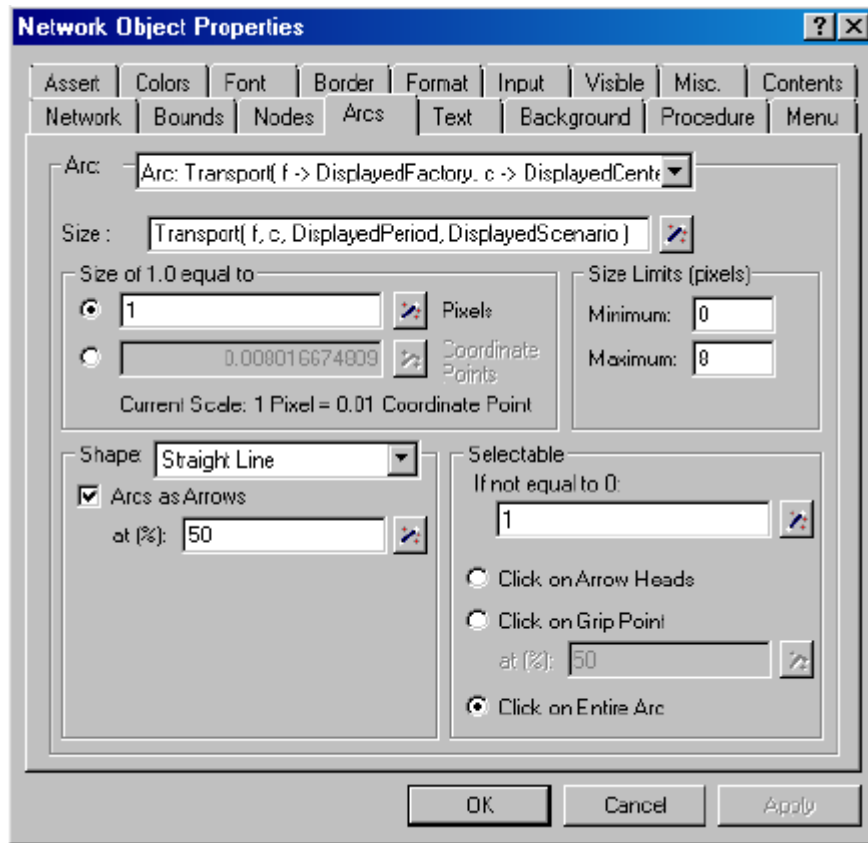


图 11.28 Network Properties 对话框中 Arcs 选项

假设你已解决该模型的第一步，此时网络对象中弧线应该有不同的宽度，如图 11.29 所示。



图 11.29：使用弧线厚度描述运输量的大小

终端用户选择网络对象的节点或者弧线时，AIMMS 可以显示节点或者弧线的相关信息。例如如图 11.23，页面右下角数据框将专门处理分配中心数据，而页面右上角的数据框显示某一工厂的相关信息。下面段落将告诉我们，选择弧线后如何更新这两个数据框，选择节点后如何更新其中一个数据框。

首先需要下面两个元素参数去保存当前工厂和配送中心的选择，请在 Transport

Overview Declarations 中添加以下声明:

```
ELEMENT PARAMETER:  
  identifier : DisplayedFactory  
  range : Factories  
ELEMENT PARAMETER:  
  identifier : DisplayedCenter  
  range : Centers
```

任一弧线的指定都依赖于以上两个参数, 当选择一条弧线后, 弧线两端节点的位置就是相应参数 DisplayedFactory 和 DisplayedCenter 的值, 一旦参数数值发生改变, 图 11.23 中数据框也会相应更新。为实现这一目标, 请按下列步骤操作:

- 在编辑模式中选择网络对象
- 打开 Network Object Properties 对话框
- 选择 Contents 选项
- 选 择 弧 线
Transport (f, c, DisplayedPeriod, DisplayedScenario)
- 顶级 Modify 按钮
- 选择 Next 按钮
- 从 'Index specification' 列表中选择 index f
- 从 'Link Index Entry' 下面的下拉列表中选择元素参数 DisplayedFactory
- 重复前面操作连接 index c 和 DisplayedCenter
- 按 Finish 按钮, 然后
- 按 OK 按钮

通过索引和元素参数的简单连接 (如图 11.30 所示), 我们就建立了弧线选择和数据框之间连接。该功能同样适用 AIMMS 中其他数据对象。

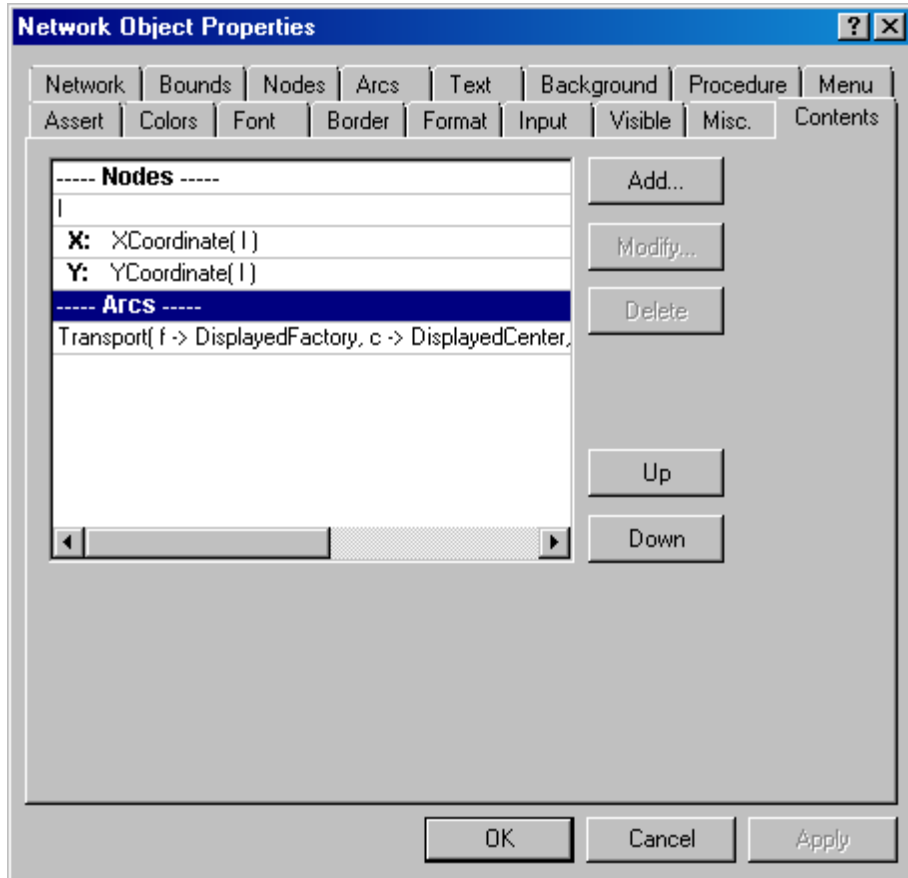


图 11.30: Network Object Properties 对话框中 Contents 选项

指定节点比弧线要困难，因为一个节点即可以使工厂的位置也可以使配送中心的位置。这造成了难以建立节点和数据框之间的对应关系。然而存在一个简单过程可以解决该问题。一旦指定该过程后，即可直接连接到网络对象。

创建过程 ‘SelectLocationInNetwork(SelectedLocation)’ 其中声明 ‘Selected-location’ 作为元素参数，该参数包含 Range 属性 Location 和 Property 属性 Input，具体如图 11.31 所示。以下条件语句作为该过程的 Body 属性：

```

if ( SelectedLocation in Factories )
then DisplayedFactory := SelectedLocation ;
else DisplayedCenter := SelectedLocation ;
endif;

```

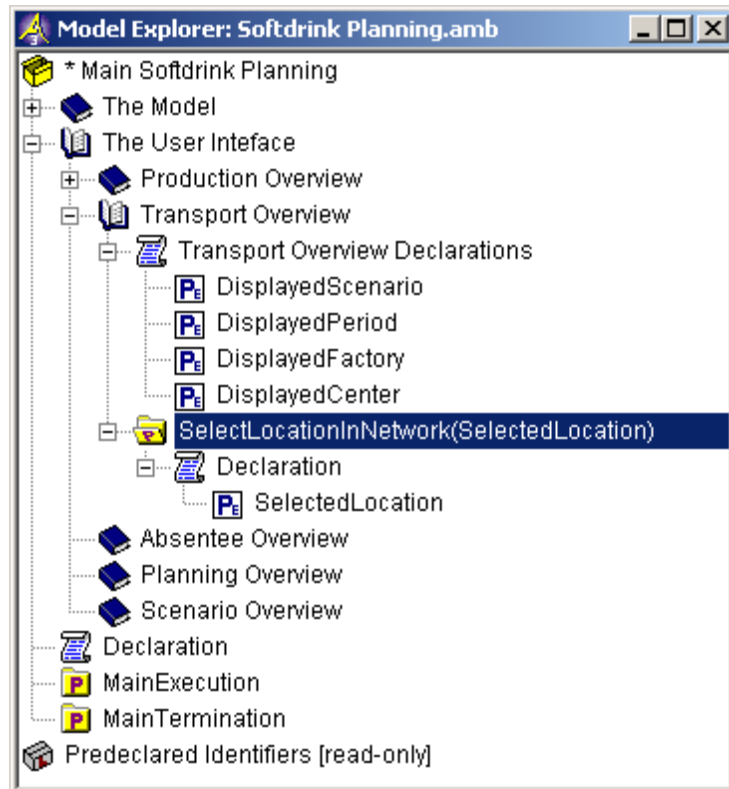


图 11.31: Transport Overview 段目录

通过下面步骤，将以上程序和网络对象进行连接作为在选程序：

- 在编辑模式中选择网络对象
- 打开 Network Properties 对话框
- 选择 Procedure 选项
- 验证是否选择 ‘Node: 1’ 作为 ‘Identifier’
- 选择程序 SelectLocationInNetwork 作为 ‘Upon Selection’ 程序
- 按 Next 按钮
- 选择 Index 单选按钮
- 从 ‘Index’ 下拉列表中选择 index 1 (如图 11.32)
- 按 Finish 键，然后
- 按 OK 按钮

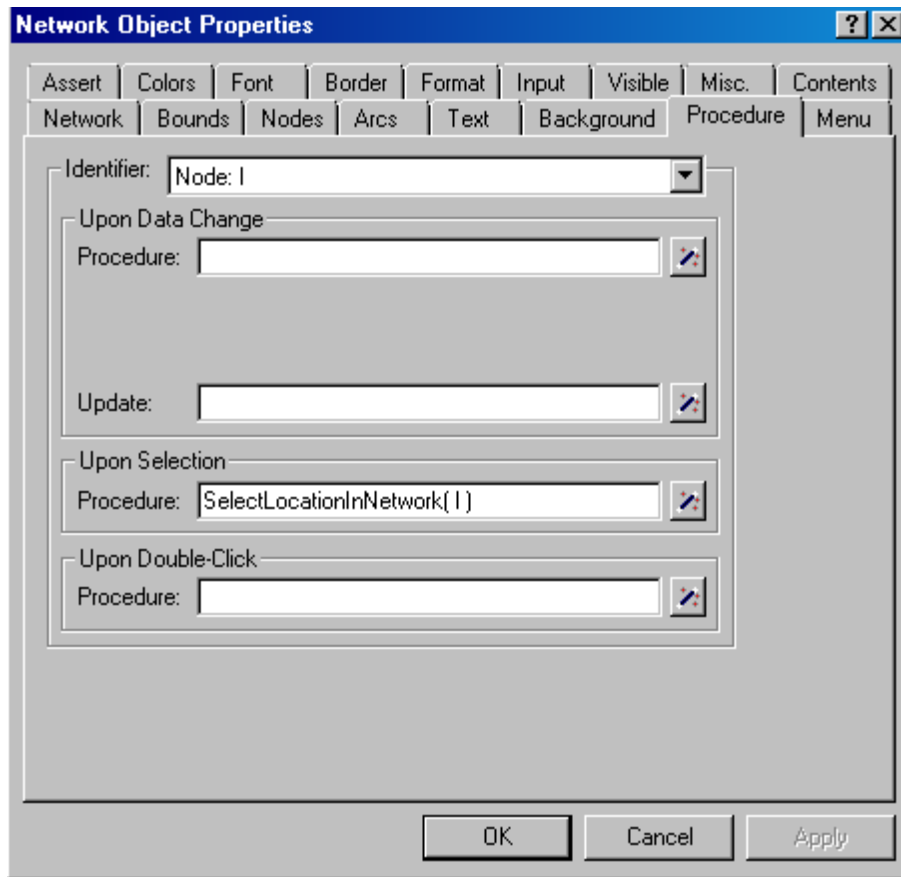


图 11.32: Network Object Properties 对话框中 Procedure 选项

如果想更清楚看到网络中节点。可以通过修改网络对象中 Node 选项增加节点大小
具体如图 11.33 所示，如果需要的话，也可以使用 Colors 选项修改节点颜色

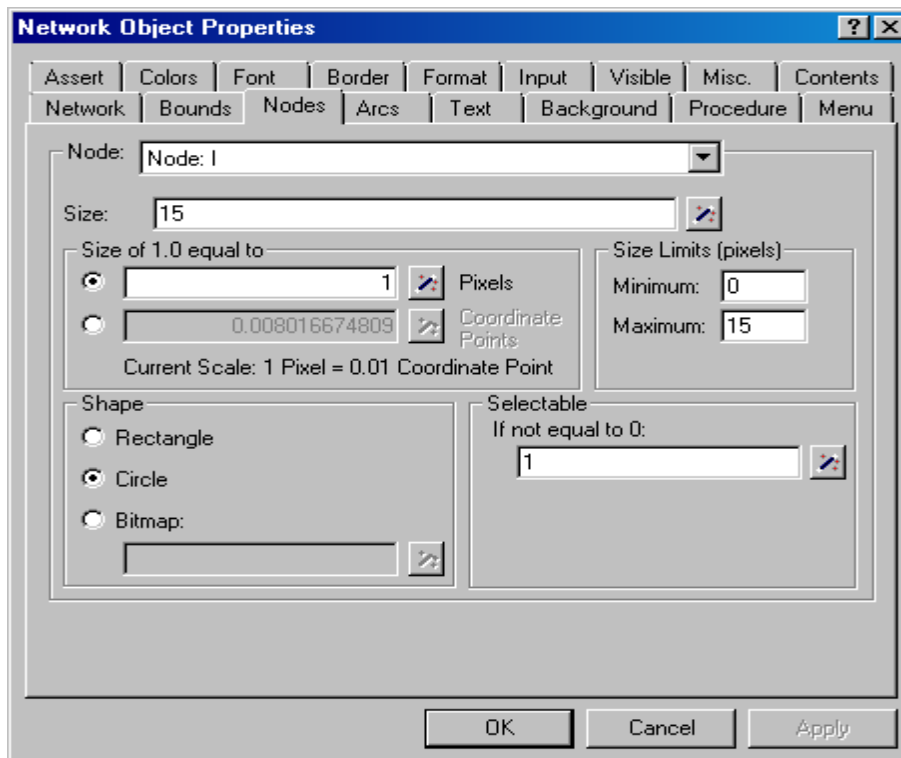


图 11.33: Network Object Properties 对话框中 Nodes 选项

增加节点大小后，网络对象图像如图 11.34 所示，如果你想用单独的节点集合区分工厂和配送中心，可以在网络对象中使用不同的图标区分他们。



图 11.34：增加节点大小后的网络对象

11.3.4: 工厂文本对象

Transport Overview 页面右上角的数据框包含某一工厂的附属数据，通过文本对象使工厂名称显示在数据框的顶部，需要在对象中填写下面串参数：

STRING PARAMETER:

identifier : FactoryDescription

definition : FormatString("%e", DisplayedFactory)

请在 'ransport Overview Declarations 段结尾增加以上声明，然后创建文本对象，显示你已声明的串参数的内容。请试着独立完成创建，在显示字符串参数之前，你还需要完成 Text Properties 对话框的 Text 选项，如图 11.35 所示，你可以改变其颜色和字体大小。

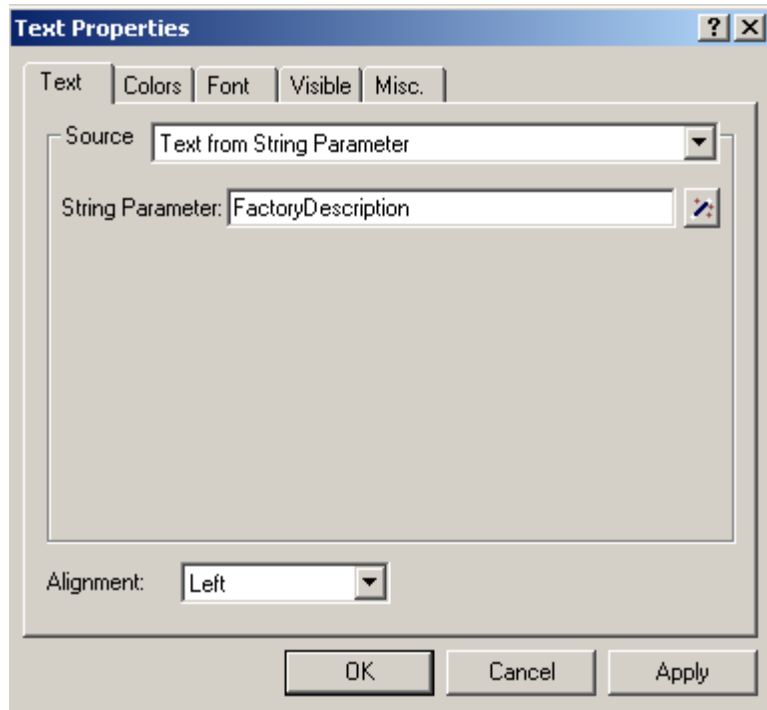



图 11.35: Text Properties 对话框中 Text 选项

11.3.5 工厂生产柱形图

现在开始创建一个条形图，该图包含当前所选工厂相应的生产数据。该过程的名称等于元素参数 `DisplayedFactory` 的数值，请按下列步骤执行：

- 首先确保运输概述页面在编辑模式下打开
- 点击工具栏中 New Bar Chart 按钮 
- 在工厂描述文本对象下方拖拽出一个矩形
- 在 Identifier 向导中选择变量 `Production(f, t)`
- 按 Next 按钮
- 连接 Index f 和元素参数 `DisplayedFactory`，然后
- 按 Finish 按钮

考虑到沿着 X-轴的时间可能太长而不满足，而参数 `Period-Description` 甚至包含更长的字符串，因此在创建短的参照之前需要创建下列字符串参数：

STRING PARAMETER:

identifier : ShortPeriodDescription

index domain : t

definition :

if (WeekInPeriod(t))

then FormatString("%n", TimeslotCharacteristic(WeekInPeriod(t), ' week'))

else ""

endif

然后通过 Bar Chart Properties 对话框中 Element text 选项将索引 t 的元素描述改为字符串参数 ShortPeriodDescription。此时，您屏幕中页面应类似于图 11.36 中部分完成的 Transport Overview 页面

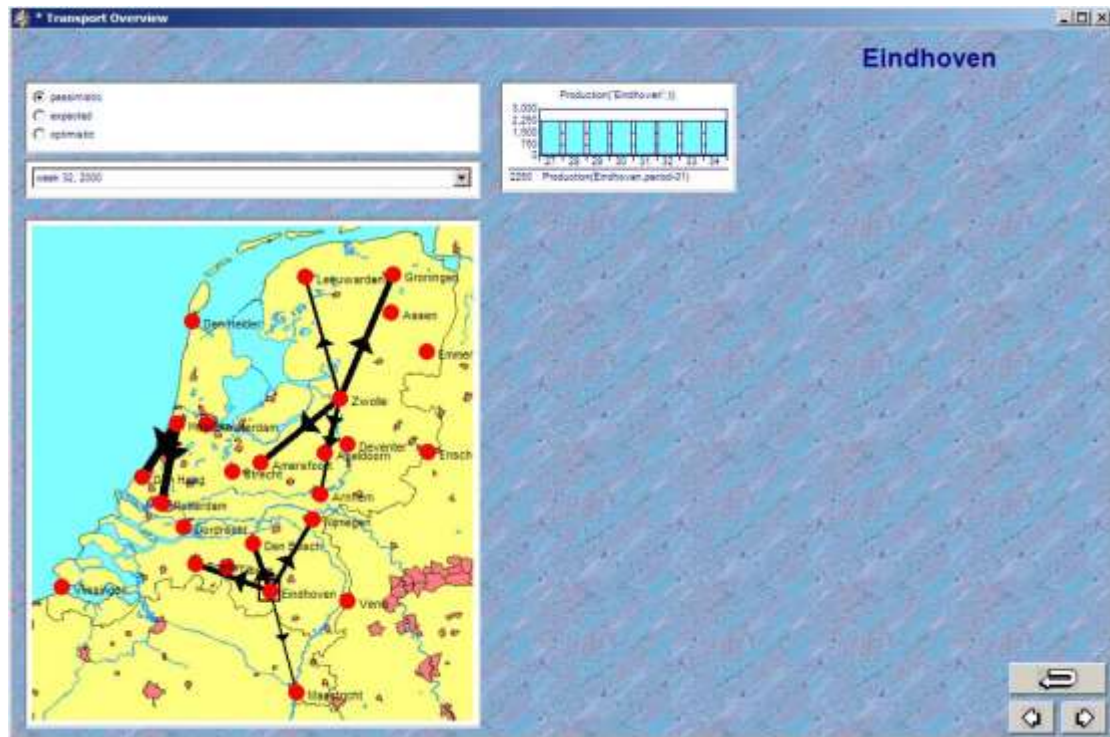





图 11.36:当前 Transport Overview 页面

11.3.6 工厂库存柱形图

为创建柱形图包含当前所选工厂的库存值，您可以使用下面的复制, 粘贴和调整操作：

- 选择刚刚创建的生产条形图
- 按工具栏中复制按钮
- 按工具栏中黏贴按钮
- 移动新柱形图，使其位于生产条形图下方
- 按工具栏中 Properties 按钮
- 选择 Contents 选项
- 从列表中选择标识符 Production(DisplayedFactory, t)
- 按 Modify 按钮
- 选择标识符 Stock(l, t, s)
- 按 Next 按钮
- 连接索引 t 和元素参数 DisplayedFactory
- 连接索引 s 和元素参数 DisplayedScenario
- 按 Finish 按钮，然后
- 按 OK 按钮

11.3.7 工厂运输综合表

当前网络对象仅显示所选期间的运输值。要想查看计划期间全部时期的运输值。你需要按照下列步骤创建一张综合表：

- 点击工具栏中 New Composite Table 按钮 
- 在要么拖拽出一个矩形
- 选择参数 Transport (f, c, t, s)
- 按 Next 按钮
- 连接索引 f 和元素参数 DisplayedFactory
- 连接索引 s 和元素参数 DisplayedScenario
- 按 Finish 按钮

如果想提高表格的整体外观，请执行下列步骤：

- 指定字符串参数 PeriodDescription(t) 作为索引 t 的元素文本，然后
- 修改子段落 11.2.8 已指定的 ‘Data Font’ 字体

完成后的表格应该如图 11.37 所示：

c	t	Transport	
Breda	week 27, 2000	6.8	▲
Breda	week 28, 2000	0.6	
Breda	week 32, 2000	5.4	
Breda	week 33, 2000	5.2	
Breda	week 34, 2000	5.4	
Den Bosch	week 29, 2000	4.2	
Den Bosch	week 31, 2000	13.9	
Dordrecht	week 27, 2000	4.3	
Dordrecht	week 32, 2000	3.6	▼

图 11.37：工厂运输综合表

11.3.8 工厂属性标量对象

为了能查看所选工厂最大和最小库存水平以及最大运载能力，首先需要创建一个标量对象并包含第一个标识符

- 创建标量对象
- 选择标识符 MinimumStock(1)，然后
- 连接索引 l 和元素参数 DisplayedFactory

接下来，按照下列步骤，在标量对象中添加剩余两个标识符：

- 打开 Properties 对话框
- 选择 Contents 选项
- 按 Add 按钮
- 选择标识符 MaximumStock (1)
- 按 Next 按钮
- 连接索引 l 和元素参数 DisplayedFactory
- 按 Finish 按钮
- 按 Add 按钮

- 选择标识符 MaximumTransportCapacity (f)
- 按 Next 按钮
- 连接索引 f 和元素参数 DisplayedFactory
- 按 Finish 按钮，然后
- 按 OK 按钮



完成后表格应如图 11.38 所示，并包含适当数据

MinimumStock(Eindhoven)	=	8330 [hl]
MaximumStock(Eindhoven)	=	33320 [hl]
MaximumTransportCapacity(Eindhoven)	=	60.0 [TL]

图 11.38: 包含边界的工厂标量对象

11.3.9 工厂生产线表格

工厂生产线表格和 Production Overview page 页面的生产线表格基本类似，除了索引 t 需要被元素参数 DisplayedFactory 代替。下面步骤关于如何从一个页面复制表格到另一个页面：

- 同时在编辑模式中打开 Production Overview 和 Transport Overview 页面
- 选择 Production Overview 页面
- 选择生产线表格
- 点击工具栏复制按钮 
- 改变该页面
- 选择 Transport Overview 页面
- 点击黏贴按钮 
- 移动目标到每个工厂信息对象下面，然后
- 点击鼠标左键

如果只想显示当前所选工厂信息，则需要执行下面步骤：

- 打开新表格的 Properties 对话框
- 选择 Contents 选项
- 在列表中选择 DeteriorationLevel (f, p)
- 按 Modify 按钮
- 按 Next 按钮
- 连接索引 f 和元素参数 DisplayedFactory 后关闭向导界面
- 在列表中选择 ActualProduction (f, p, t)
- 按 Modify 按钮
- 按 Next 按钮
- 连接索引 f 和元素参数 DisplayedFactory 后关闭向导界面
- 从列表中选择 ActualNumberOfDaysInPeriod (t)
- 按 Delete 按钮，然后
- 按 Apply 按钮

在 Colors 选项中仍然存在关于索引 f 的引用，所以会出现一个错误对话框，点击对话框窗口的 OK 按钮，AIMMS 会剔除索引引用（即，移除 DeteriorationColor(f, p)）。因此你需要再次为 DeteriorationLevel(f, p) 指定颜色，并改变其索引应用，可通过执行下列步骤实现：

- 选择 Colors 选项
- 在 Identifier 段中选择 Model 作为颜色决定者
- 再次按 Wizard 按钮 ，选择标识符 Deterioration-Color(f, p)
- 连接索引 f 和元素参数 DisplayedFactory
- 按 Finish 按钮，然后
- 按 OK 按钮

最终表格如图 11.39 所示：

	Deterioration	Production				
		past	week 27, 2000	week 28, 2000	week 29, 2000	week 30, 2000
line-01	3.25		2250.00	2250.00	2250.00	2250.00
line-02	6.25					
line-03						
line-04						

图 11.39：工厂生产线表格

此时，使用 11.2.8 章节介绍的校准和调整功能重新排列组合对象（如图 11.40 所示）。当工厂数据框整齐排列后，然后复制全部，从而为配送中心创建一个类似数据框。

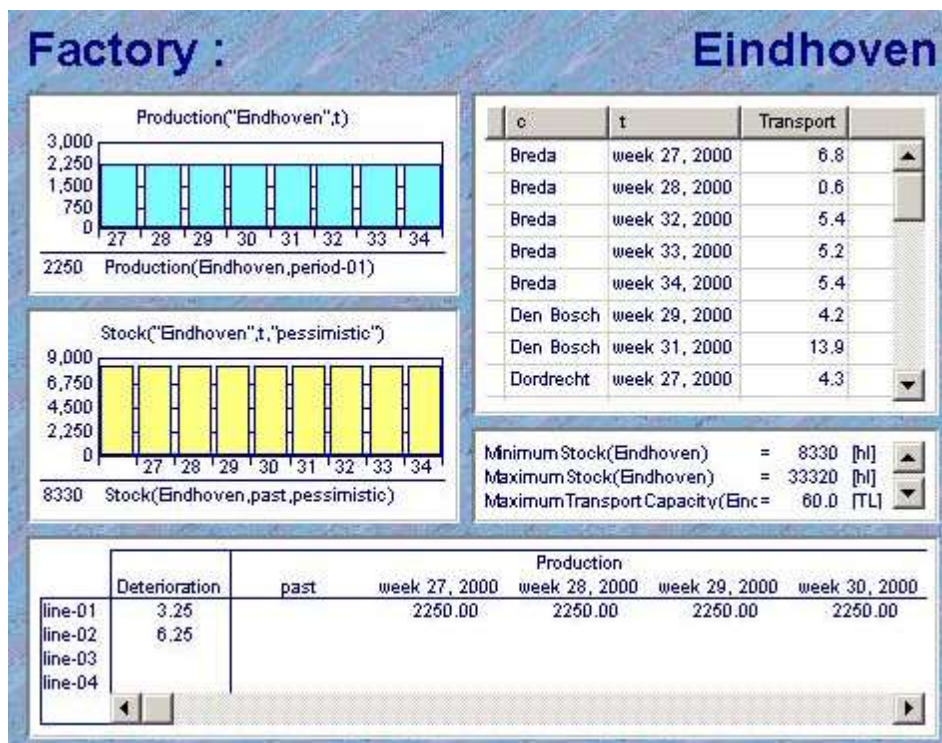


图 11.40：工厂数据框

11.3.10 配送中心数据框

请执行下列步骤，创建 4 个关于配送中心页面对象



- 按住 Shift 键，在工厂数据框中选择除顶端的生产线表格外的全部页面对象
- 点击工具栏中复制按钮
- 点击工具栏中黏贴按钮
- 移动对象，使其位于工厂信息域下方（如图 11.41 所示），然后
- 按鼠标左键



图 11.41: 复制和黏贴过程说明

现在，你应该有足够经验完成一系列修改，将工厂数据框转换成配送中心数据框。首先在 Transport Overview Declarations 段结尾增加下列声明：

STRING PARAMETER:

identifier : CenterDescription

definition : FormatString("%e", DisplayedCenter)

现在需要执行下列步骤，并使用了到目前为止最详细的知识：

- 在文本对象的复制内容中，使用 Text Properties 对话框中 Text 选项，将字符串参数 'FactoryDescription' 改为字符串参数 CenterDescription
- 从标量对象的 Contents 选项中移除 MaximumTransportCapacity-(DisplayedFactory)
- 在生产柱形图的 Contents 选项中找到 Production(Displayed-Factory, t)
- 将其改为 Demand(DisplayedCenter, t, DisplayedScenario)
- 在工厂运输综合表的 Contents 选项找出 Transport(Displayed-Factory, c, DisplayedPeriod, DisplayedScenar

- io)
- 将 其 改 为 Transport (f, DisplayedCenter, DisplayedPeriod, DisplayedScenario)
- 依次打开表格, 标量对象, 两个柱形图的 Properties 对话框中 Contents 选项
- 替换全部参考到元素参数 DisplayedFactory, 有一个被替换到元素参数 DisplayedCenter

11.3.11 完成页面

此时, 从 Production Overview 页面复制三个执行按钮 (Run Next, Run All and Restart), 在 Transport Overview 页面的相同位置黏贴, 为此你需要介绍新的模板页面。

最后, 你可以通过 11.2.8 节所说的方式: 增加矩形, 改变文本颜色和尺寸, 提高页面质量。当你完成 Transport Overview 页面后, 可与图 11.42 进行比较参考。

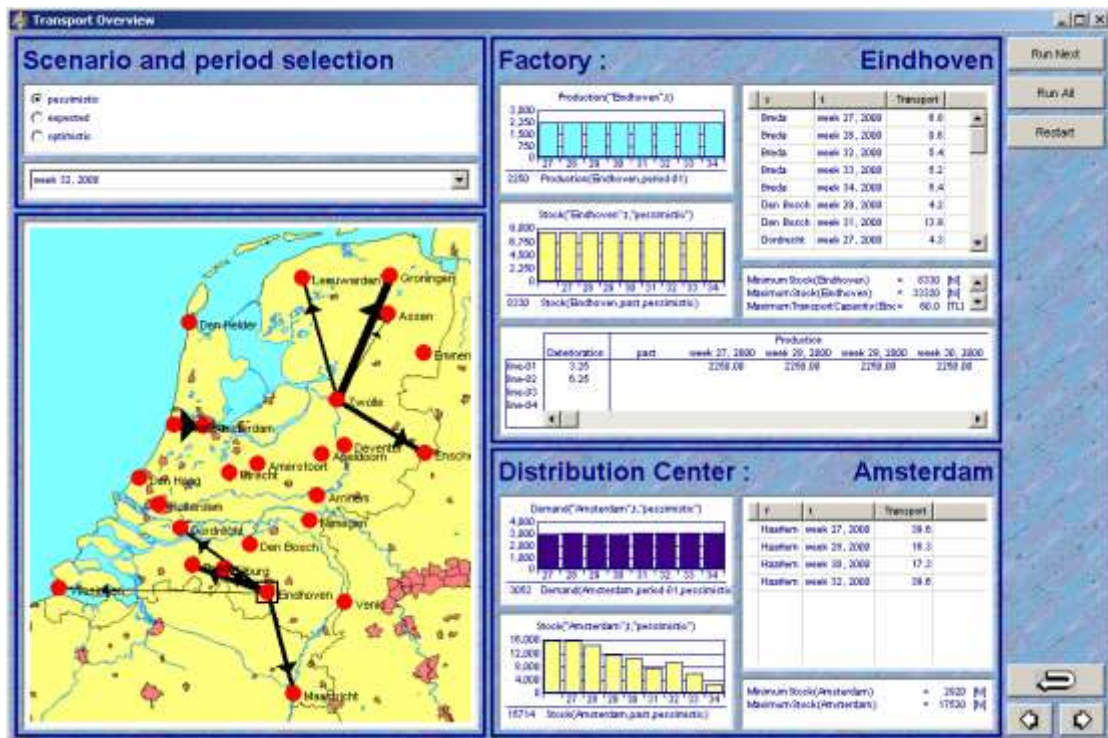


图 11.42: 完成后的 Transport Overview 页面

第十二章

Absentee 与规划概要

在本章，你将会构建包括 Gantt 表及用来展示模型数据的复合表在内的两个目的用户页面。Gantt 是一种高级的页面对象，它对于展示图和规划随着时间变化的数据特别有效。

12.1 Gantt 表

Gantt 表通常包括以时间刻度相对应的一系列相关联的任务、过程或者工作。这种表格包括一行或者多行显示水平条形的行。每一个水平条形代表着一个任务，而这个条形的长度则对那个具体工作的完成时间和日期给出了一个直观的印象。行通常指单个任务所需要的资源。你的表格有可能需要几种类型的任务(如维护任务及命令行用法任务)。在这种情况下，Gantt 表的设置可以使用条形里面的颜色和/或文本来指明每种资源所对应的任务。

你可以使用几种 AIMMS 标识符来控制 Gantt 表的外形。这些繁多的控制不可能在一个段落里解释清楚。但是，你可以随着时间控制 x 轴（见图 12.2）的时间刻度和每个条形的位置和颜色。

在本章，你将会构建三个 Gantt 表。第一个 Gantt 表用于规划每个工厂每周的假期时间。第二个 Gantt 表用于安排每天的官方假日。利用这两个表，你的目的用户通过点击表格内的条形，就可以以图的形式来规划假日和假期。第三个 Gantt 表不适用于数据输入，但是会用于展示模型的总体维护和命令行用法输出。

12.2 Absentee Overview 页面

在这一节，你将构建如图 12.1 所示的整个页面。这两个 Gantt 表和复合表将在各自的小节中得到阐述。

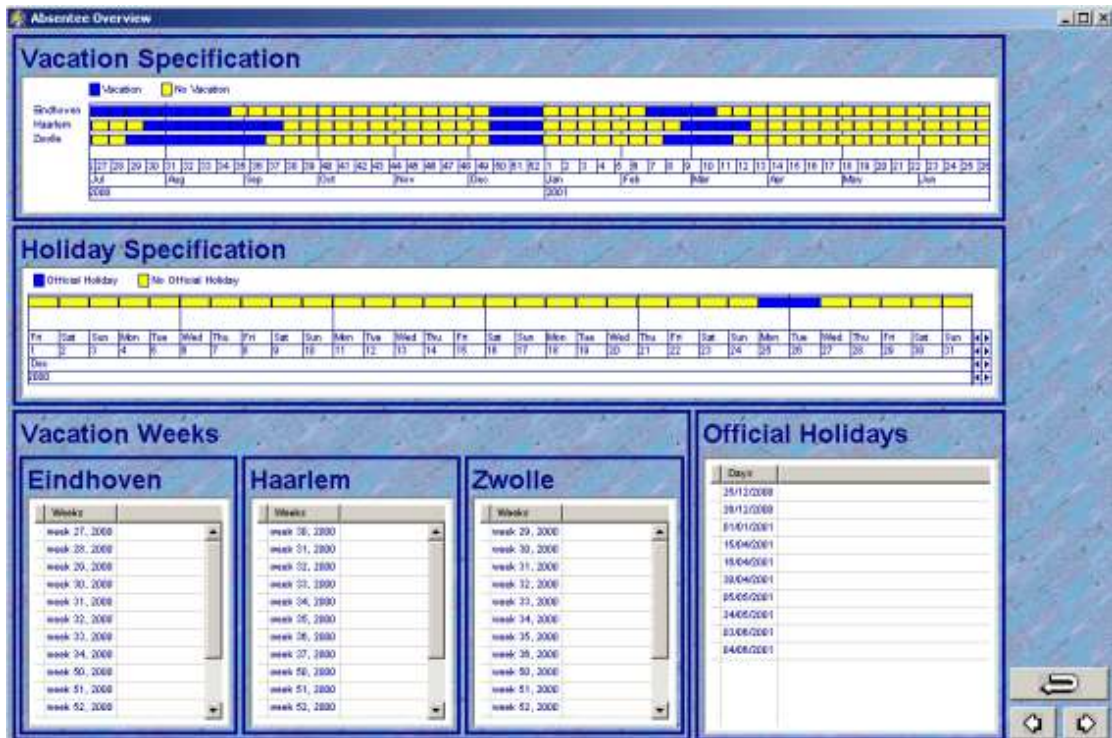


图 12.1 完整的缺值概要页面

12.2.1 假期 Gantt 表

假期 Gantt 表对每一个工厂都有一行。工厂可以被认为拥有工人的资源。当工人放假时，一部分的资源就被使用。在这个 Gantt 表中，每一行有两种类型的有色条形。一个条形是表示某周已被计划为“假期”，而另一行则表示一种情况。你将制作的 Gantt 表的一部分如图 12.2 所示。

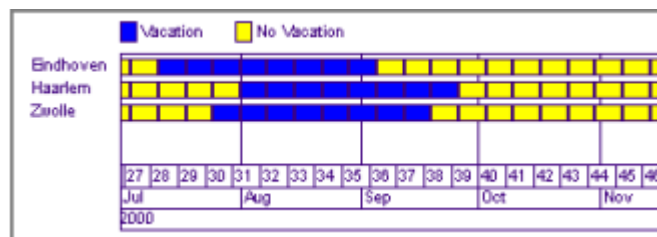


图 12.2 假期计划 Gantt 表

Gantt 表展示 x 轴上所有可能的星期。这个表上的每一个条形都有一个星号作为标记，表明星号所指的特定的星期，以及用于表示条形长度的持续时间。假期 Gantt 表使得直接用户可以通过点击鼠标来设置假期。为了构建这种工具，你需要声明几个标识符和一个可以转换条形在“假期”与“无假期”状态之间的简单程序。在你的模型的 Absentee Overview 部分，插入新的声明节 Vacation Gantt Declarations，并增加以下的声明。

SET:

identifier : VacationGanttChartBarTypes

index : v

definition : data { ' Vacation' , ' No Vacation' }

ELEMENT PARAMETER:

identifier : VacationGanttChartStartingWeek

index domain : w

range : Weeks

definition : w

PARAMETER:

identifier : VacationGanttChartDuration

index domain : (f, w, v)

每当直接用户选择 Gantt 表中的一条形时，你可以让 AIMMS 执行某个程序。在这个例子中，你希望这个程序切换“假期“与”非假期“的状态。以下的语句实现了这个任务：

```
VacationGanttChartDuration(f, w, v) := 1 - VacationGanttChartDuration(f, w, v);
```

当相应的程序执行时，持续时间参数的值便在 0 和 1 之间切换。

创作一个如图 12.3 所示的名为 ToggleVacationGanttChart(f, w) 的新的程序。使用 **Argument** 向导以声明 f 为 set Factories 里面的一个有输入特点的元素参数。类似的，声明 w 为 weeks 日历里面的一个有输入特点的元素参数。接下来，在 **Body** 属性里输入上一段落的语句。

持续参数的使用有三种方式。第一，如之前所说，它可以被用来表示条形的长度。值 1 正好对应 x 轴上的时间间隔的长度，即一周。而且，这个参数可以作为 Gantt 表的一个值域参数，表示将要画的条形。最后，持续时间参数可以用来建立 Gantt 表和数学程序里使用的 set VacationWeeks(t)。

使 Gantt 表初始化的程序和切换持续时间参数的程序是一样短的。**Body** 属性只需用到下列的语句：

```
VacationGanttChartDuration(f, w, ' No Vacation' ) :=
```

```
1 - VacationGanttChartDuration(f, w, ' Vacation' )
```

所有的值都在默认为零的初始值，这个语句使所有星期初始化为“无假期”周。请增加如图 12.3 所示的步骤 InitializeVacationGanttChart，并在 **Body** 属性里插入以上语句。

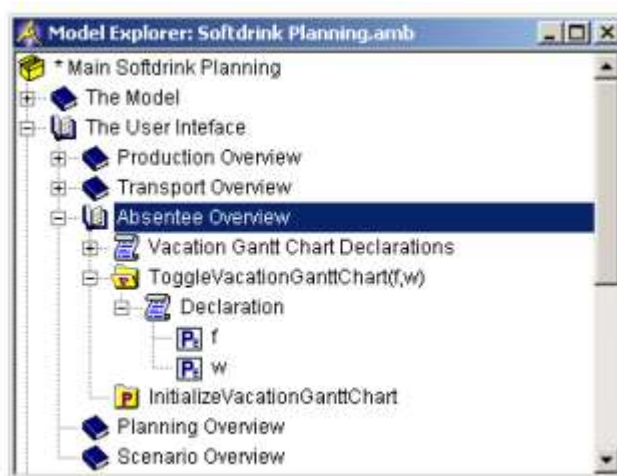



图 12.3: Absentee Overview 节的内容

现在你应该回到 MainInitialization 程序，并在 **Body** 属性的后面添加

InitializeVacationGanttChart。你可以通过按 Ctrl-F 的组合键或工具栏里的 **Find** 按钮  (见图 12.4) 很快地在你的模型树里找到这个程序。

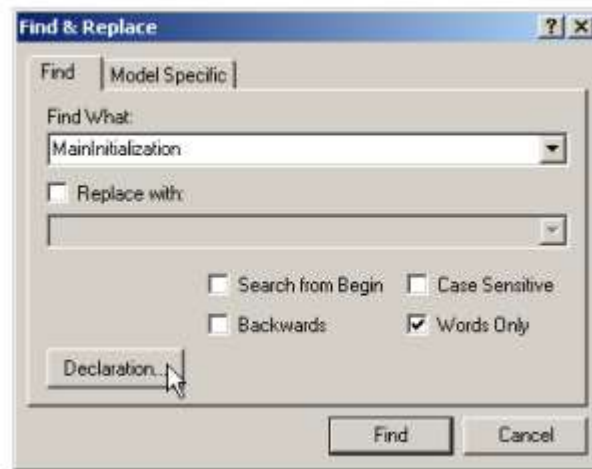


图 12.4: Find & Replace 对话框

为防止配置 Gantt 图表时的任何初始化错误，你现在可以通过在模型树里选择 InitializeVacationGanttChart 并从鼠标右键的弹出菜单里发出 **Run Procedure** 的命令来执行这个程序。

你现在可以遵循以下的步骤在页面上创造假期规格 Gantt 图表：

- 在 **Edit** 模式里打开 Absentee Overview 页面
- 在工具栏里按 **New Gantt Chart** 按钮 
- 拖放符合你页面的 Gantt 图表尺寸的长方形
- 使用 **Wizard** 按钮  来完成图 12.5 所示的 **Gantt Chart** 对话框

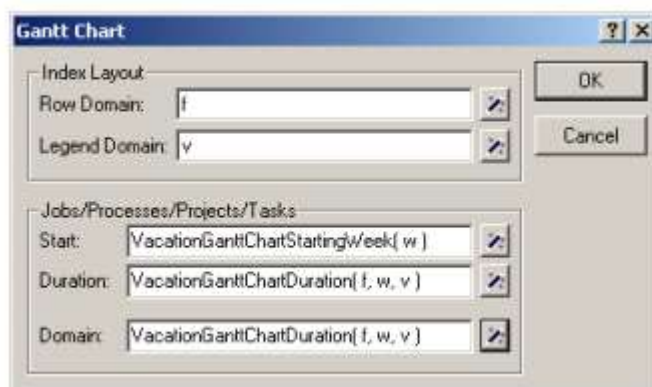





图 12.5 假日规划的 Gantt Chart 对话框

Gantt 图表的 x 轴最初展示 weeks 日历里元素的描述。通过把日历元素的描述映射到相应的时间上的瞬间，AIMMS 可以改变 x 轴上的标签。在本教程中，元素的描述包含对周，月和年

的引用。要改变 Gantt 图表上 x 轴的时间引用，你可以执行以下步骤。

- 选择 Gantt 图表
- 打开它的 **Properties** 对话框
- 选择 **X-axis** 标签
- 选择 ‘Real-time Calendar’ 作为 the ‘Type of X-axis’
- 如图 12.6 所示，选择 ‘Weeks’，‘Months’ 和 ‘Years’
- 选择 ‘weeks’ 作为 ‘Unit of Measurement’
- 使用 **Wizard** 按钮  选择 ‘String Parameter’ BeginDateOfCalendar 作为 ‘Left Bound’
- 使用 **Wizard** 按钮  选择 the ‘String Parameter’ EndDateOfCalendar 作为 ‘Right Bound’，然后
- 按 Apply 按钮

为了实现“假期”与“非假期”条形类型的自动切换，你应该完成如图 12.7 所示的步骤。取决于你的 Gantt 表的大小，Gantt 表的默认字体可能太大。建议您创建一个大小为 7 的新的 Gantt 表字体，而不是 10.3 所示的大小默认为 8 的方式。

现在 Gantt 表看起来应该如图 12.8 所示。为了检测这个表，你应该通过按页面工具栏的 **Page User Mode** 按钮 ，来把这个页面放在用户模式。当在任何条形上点击鼠标时，它的颜色就会改变，而且 Gantt 表底部的状态栏也会相应的调整。

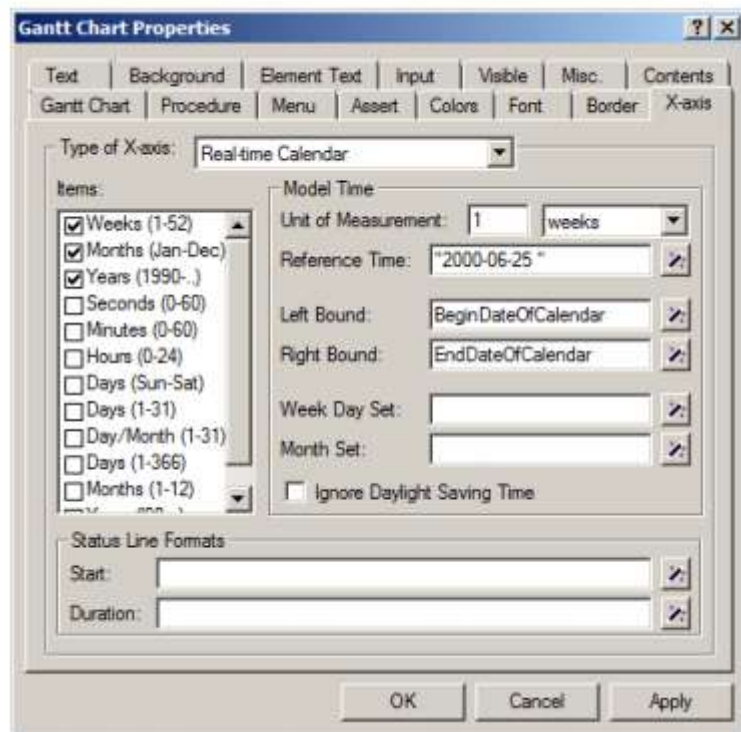


图 12.6 Gantt Chart Properties 对话框的 X-axis

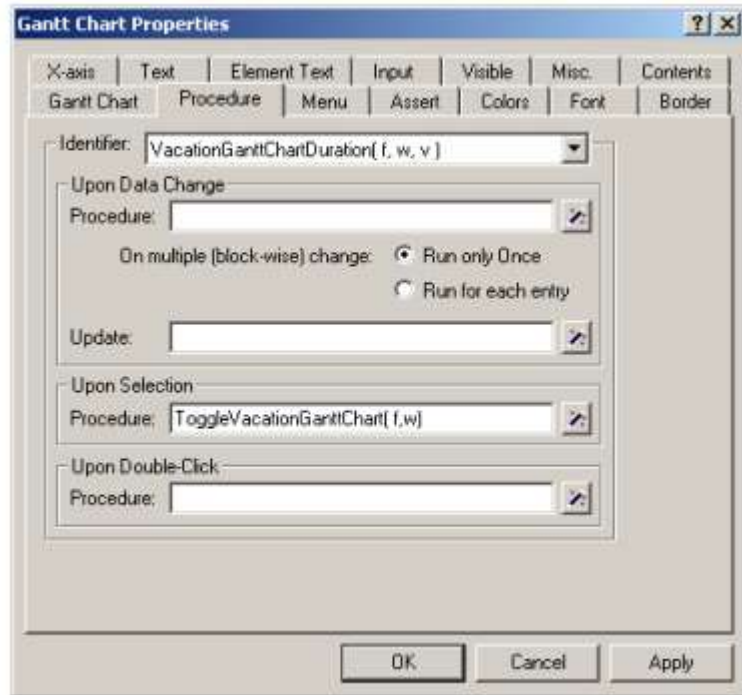


图 12.7: Gantt Chart Properties 对话框的 Procedure 标签

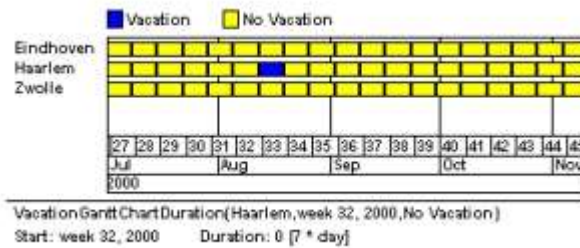


图 12.8: 完整的假期 Gantt 表

通过点击 Gantt 表的条形，终端用户可以修改 `VacationGanttChartDuration(f, w, v)` 的参数值。输入数据的变更必须发送到数学程序里使用的 `VacationWeeks` 集。你可以提供以下语句作为该集的 **Definition** 属性，这样就可以很容易地完成这个数据链接：

```
{ w | VacationGanttChartDuration(f, w, ' Vacation' ) }
```

12.2.2 假日 Gantt 表

假日 Gantt 表类似于假期 Gantt。主要的区别在于假日 Gantt 表以日而不是周作为指定，而且它包含一行而不是三行。

假日 Gantt 表包含两种类型的条形。一种条形类型表明某天是官方假日，而另一种表明相反的情况。如图 12.1 所示，这两种条形类型还作为图例。

你现在应该在 `Absentee Overview` 节里插入新的声明。在新的声明节里，需要输入以下三个标识符。

SET:

identifier : `HolidayGanttChartBarTypes`

index : h

```

definition : data { ' Official Holiday' , ' No Official Holiday' }
ELEMENT PARAMETER:
identifier : HolidayGanttChartStartingDay
index domain : d
range : Days
definition : d
PARAMETER:
identifier : HolidayGanttChartDuration
index domain : (d,h)

```

然后，和之前小节的程序 `ToggleVacationGanttChart(f,w)` 一样，引入一个程序 `ToggleHolidayGanttChart(d)`。它的参数应该被声明为 `Days` 集里的元素参数，而且 **Property** 属性为 `Input`，**Body** 属性应该包含以下语句：

```
HolidayGanttChartDuration(d,h) := 1 - HolidayGanttChartDuration(d,h);
```

由于总的规划期的天数很多，要在单一个 Gantt 表里浏览所有的单独的天是不可能的。滚动条是必须的。AIMMS 让你可以设置字符串参数作为 Gantt 表的左右界限。当字符串参数是可更新的模型标识符时，随着你移动时间，这些参数的值便会调整。注意，之前小节的边界参数是有定义的字符串参数，因此是不能更新的。它们的值是不能改变的，所以 AIMMS 不会显示任何滚动条。

请补充以下两个声明到 `Holiday Gantt Chart Declarations` 节。

```

STRING PARAMETER:
identifier : HolidayGanttChartLeftBound
STRING PARAMETER:
identifier : HolidayGanttChartRightBound

```

边界参数和持续时间参数都需要在一个新的程序 `InitializeHolidayGanttChart` 里初始化。你可以把这个程序直接放在程序 `ToggleHolidayGanttChart` 下面。**Body** 属性应该以下规定：


```

HolidayGanttChartLeftBound := BeginDateOfCalendar;
HolidayGanttChartRightBound := "2000-08-01";

HolidayGanttChartDuration(d,' No Official Holiday' ) :=
1 - HolidayGanttChartDuration(d,' Official Holiday' );

```

注意，持续时间参数初始化和假期 Gantt 里的初始化是一样的。现在你应该回到 `MainInitialization` 程序，并在 **Body** 属性后面补充 `InitializeHolidayGanttChart` 语句。

如之前所示，你可以按 `Ctrl-F` 组合键或工具栏里的 **Find** 按钮  在你的模型树里快速找到这个程序。

为了防止规定 Gantt 表时的任何初始化错误，你应该选择模型树里的 `InitializeHolidayGanttChart` 来执行它并从鼠标右键弹出菜单里发出 **Run Procedure** 的指令。

图 12.9 展示含有和假日 Gantt 表相关声明的模型树。

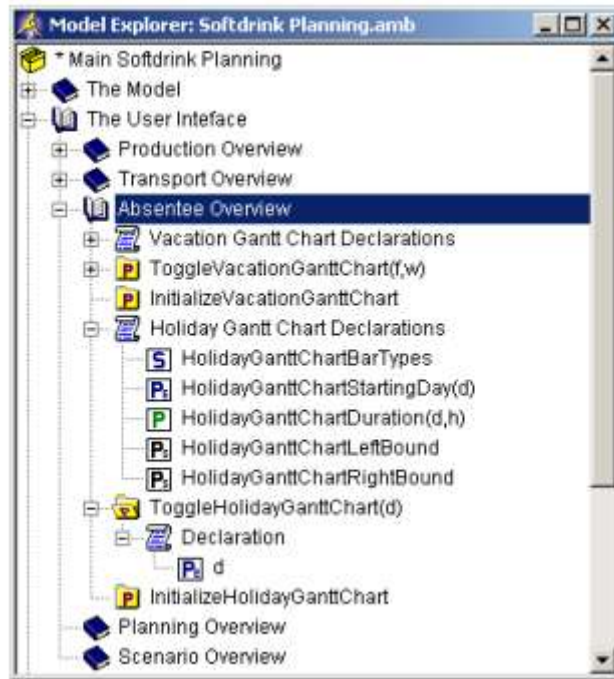


图 12.9 Absentee Overview 节的内容

你现在可以遵循以下的步骤，在假期规格 Gantt 表下面上创造假日规格 Gantt 图表：

- 在 **Edit** 模式里打开 Absentee Overview 页面
- 在工具栏里按 **New Gantt Chart** 按钮 
- 拖放符合你页面的 Gantt 图表尺寸的长方形
- 使用 **Wizard** 按钮  来完成图 12.10 所示的 **Gantt Chart** 对话框

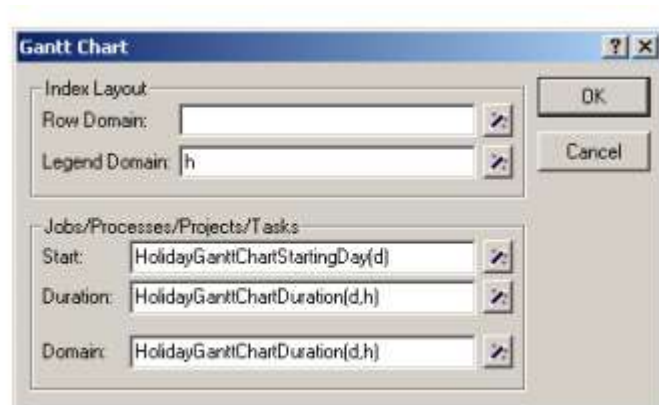





图 12.10 假日规划的 **Gantt Chart** 对话框

Gantt 图表的 x 轴最初展示 weeks 日历里元素的描述。通过把日历元素的描述映射到相应的时间上的瞬间，要改变 Gantt 图表上 x 轴的时间引用，你可以执行以下步骤。

- 选择 Gantt 图表
- 打开它的 **Properties** 对话框
- 选择 **X-axis** 标签
- 选择 ‘Real-time Calendar’ 作为 the ‘Type of X-axis’

- 如图 12.11 所示，选择 ‘Days (Sun-Sat)’，‘Days (1-31)’，‘Months’ 和 ‘Years’
- 选择 ‘days’ 作为 ‘Unit of Measurement’
- 使用 **Wizard** 按钮  选择 ‘String Parameter’ BeginDateOfCalendar 作为 ‘Reference Time’
- 使用 **Wizard** 按钮  选择 ‘String Parameter’ HolidayGanttChartLeftBound 作为 ‘Left Bound’
- 使用 **Wizard** 按钮  选择 ‘String Parameter’ HolidayGanttChartRightBound 作为 ‘Right Bound’，然后
- 按 Apply 按钮

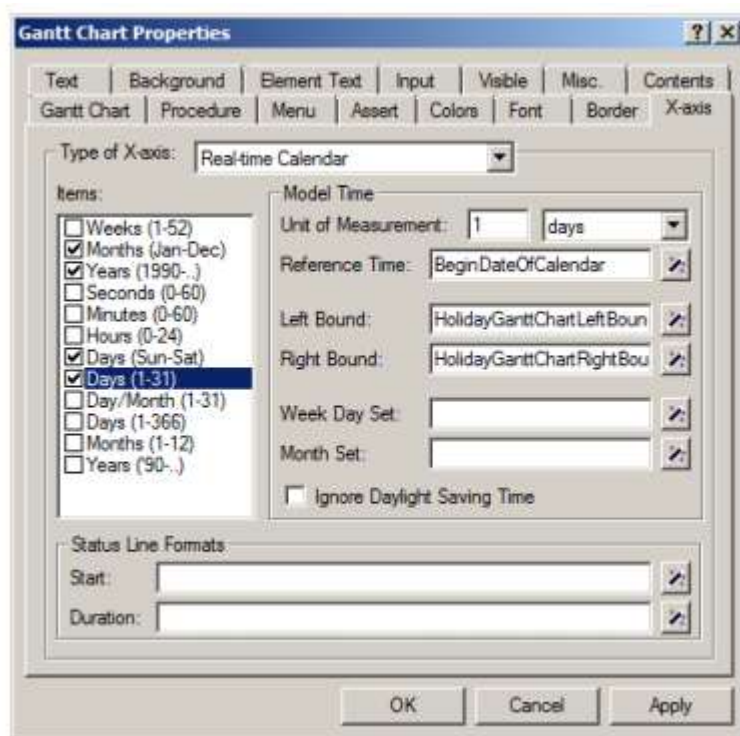


图 12.11 Gantt Chart Properties 对话框

一旦你遵循上两个段落的指示，你的屏幕就会如图 12.2 所示。

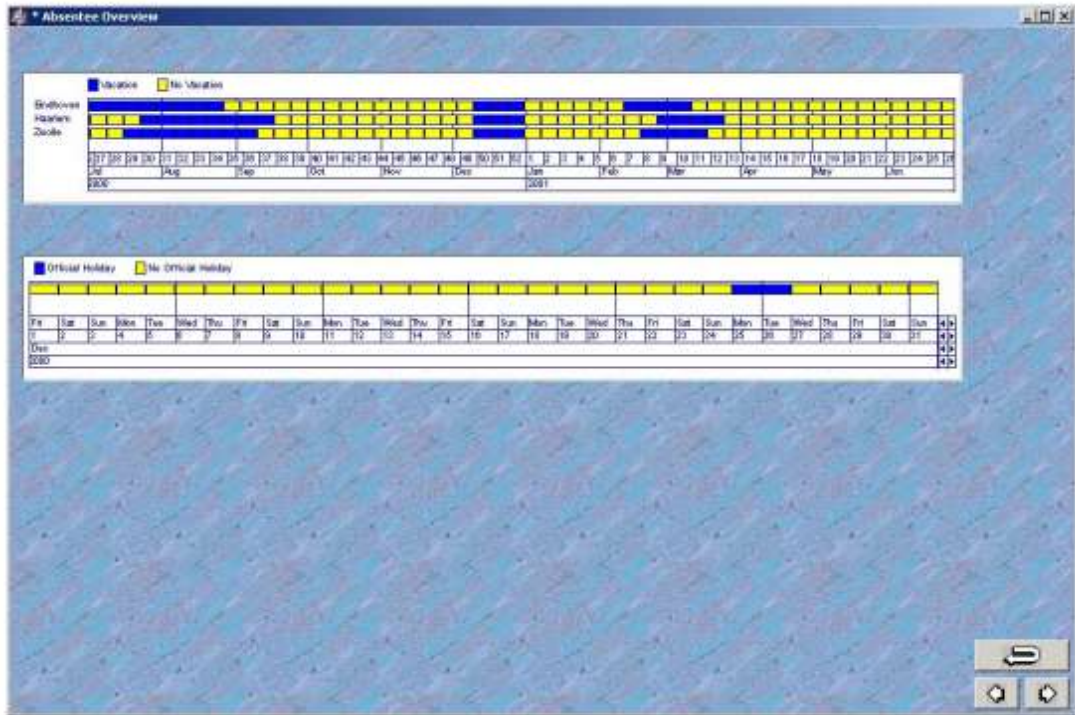


图 12.12 假日和假期规格页面

为了实现“官方假期”与“非官方假期”条形类型的自动切换，你应该完成如图 12.13 所示的 **Procedure** 标签。

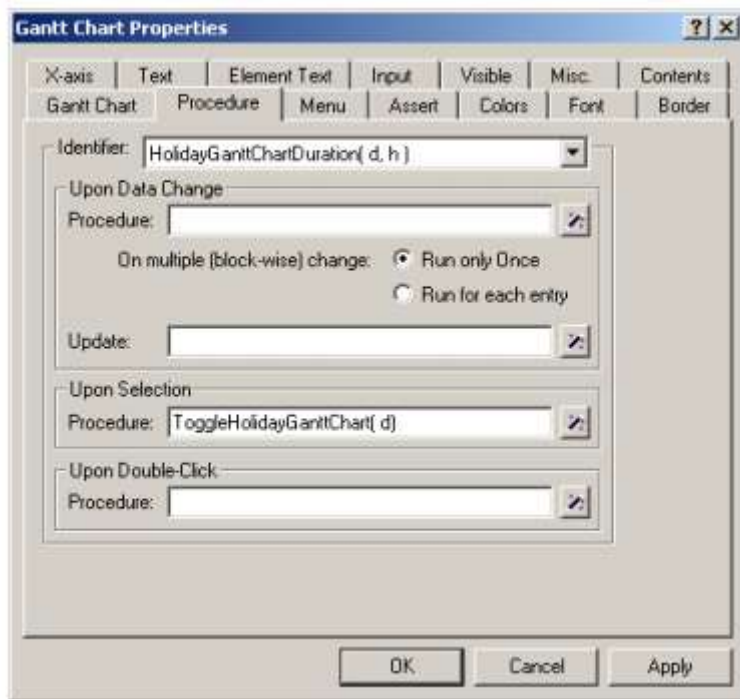


图 12.13 Gantt Chart Properties 对话框的 Procedure 标签

通过点击 Gantt 表的条形,终端用户可以修改 `HolidayGanttChartDuration(d, h)` 的参数值。


输入数据的变更必须发送到数学程序里使用的 OfficialHolidays 集。你可以使用以下语句作为 OfficialHolidays 集的 **Definition** 属性，这样就可以很容易地完成这个数据链接：

```
{ d | HolidayGanttChartDuration(d, ' Official Holiday' ) }
```

12.2.3 完成页面

在你的当前页面能像图 12.1 所示之前，你仍然需要增加四个表格。这些表格提供了关于两个 Gantt 里面的假日和假期信息的一个清楚的总结。

假如几个标识符有共同的索引范围，那么 AIMMS 里的复合表就可以包含几个这样的标识符。但是，你要创建的第一个这样的表格只包含一个标识符，也即是展示 ‘Eindhoven’ 工厂的所有假期周。要创建这种表格，你需要执行以下的操作：

- 确定页面处于 **Edit** 模式
- 按 **New Composite Table** 键 
- 在页面绘制一个矩形
- 在 **Identifier** 向导盒的第一个标签选择 VacationWeeks 集
- 如图 12.14 所示，选择 ‘Eindhoven’ 作为索引 f 的 ‘Fixed Element’

创造了你的第一个复合表之后，你就可以马上验证它对假期 Gantt 表的正确回应。只要点击假期 Gantt 表的 ‘Eindhoven’ 行的某个地方，然后表格的内容就会马上调整。

要创建在 Haarlem 和 Zwolle 的两个工厂的类似的复合表，你可以遵循同样的步骤，或者利用复制黏贴的方法。后一种选择需要以下步骤：

- 复制黏贴 ‘Eindhoven’ 的复合表
- 打开复制的复合表的 **Properties** 的话框
- 移至 **Contents** 标签
- 选择范围标识符 VacationWeeks(‘ Eindhoven’ , Weeks)
- 按 **Modify** 按钮
- 在 **Identifier** 向导按 **Next** 按钮
- 把 ‘Fixed Element’ 值从 ‘Eindhoven’ 改至 ‘Haarlem’ (or ‘Zwolle’)
- 按 **Finish** 按钮，然后
- 按 **OK** 按钮

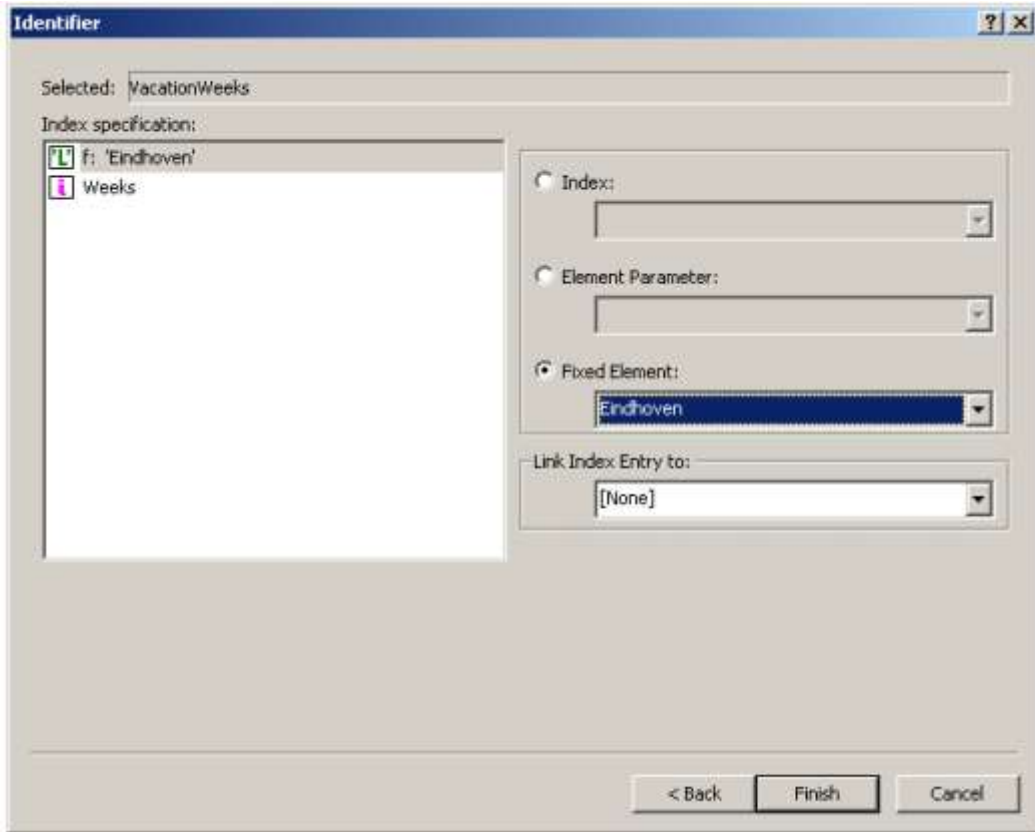


图 12.14: 标识符向导盒的内容

你可以以你创建第一个复合表的方式创建第四个复合表。这个新表应该包含 Official Holidays 集。

你的屏幕的页面还不能如图 12.15 所示。如果你喜欢的话，你可以增强你的页面，比如对齐数据对象，增加文本对象和矩形，和改变字体大小和颜色。

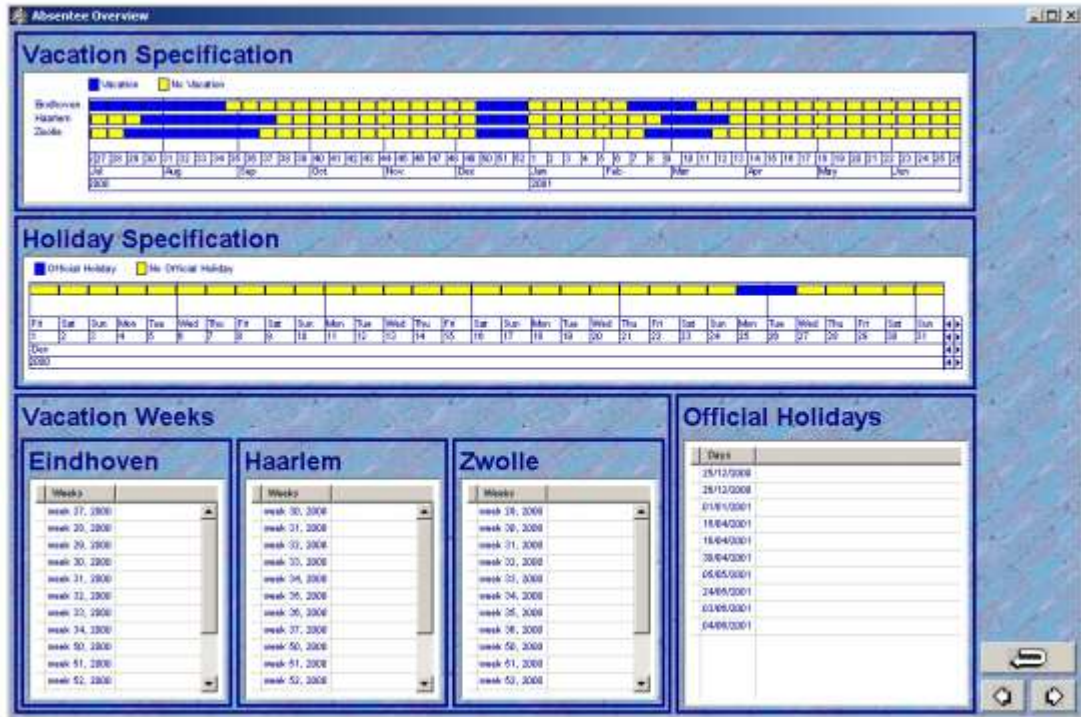


图 12.15 完整的 Absentee Overview 页面

12.3 Planning Overview 页面

在这一节，我们将创建图 12.16 所示的整个页面。我们将在不同的小节对 Gantt 图和表格分别阐述。

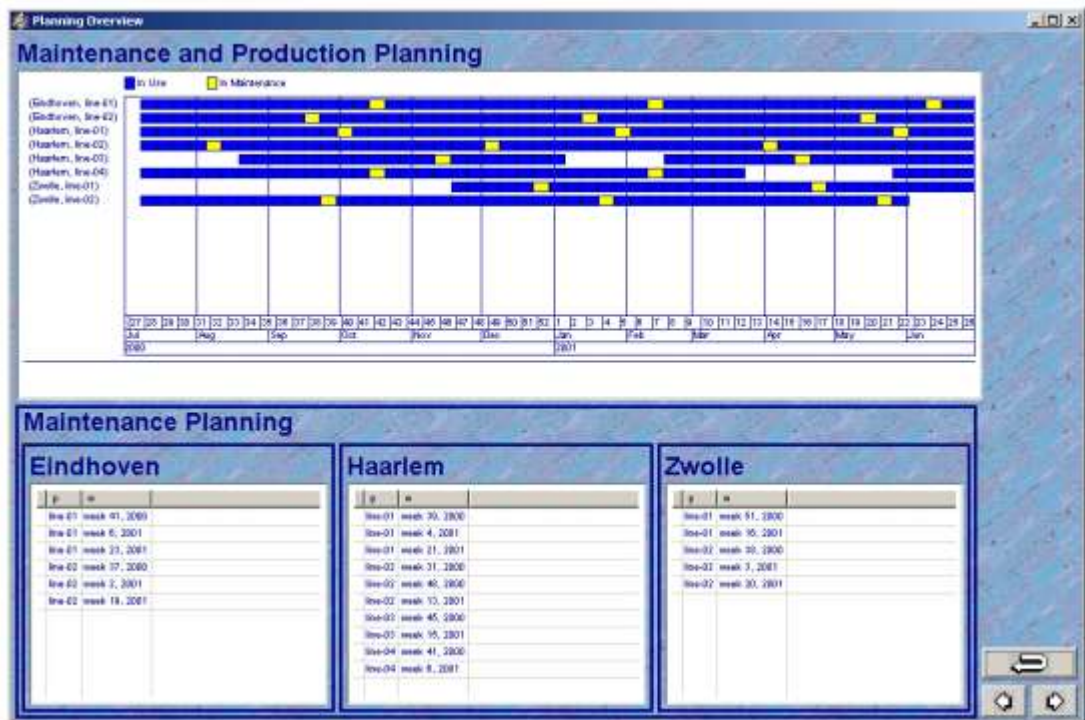


图 12.16 完整的 Planning Overview 页面

规划概要页面应该展示一个 Gantt 表，而这个表总结了每个工厂和生产线组合的规划和维护

计划表。因此，每个这样的组合是 Gantt 表的一行。在每一行都有两种类型的条形。一种条形表示相应的生产线在使用中，另一种条形表示生产线在维护中。这两种条形类型组成 Gantt 表的图例。

与之前讨论的 Gantt 表相比，规划 Gantt 表包含一种新的特点。在每一行的描述中，有对两个元素而不是一个元素的引用，即一个工厂和一个生产线。因此，要规定每个行的描述需要一个复合集而不是一个简单集。请在 Planning Overview 节里插入一个新的声明节 Planning Gantt Chart Declarations，并输入以下的声明：

SET:

```
identifier : PlanningGanttChartRows
subset of : (Factories, ProductionLines)
index : r
definition : { (f,p) | p in FactoryProductionLines(f) }
```

SET:

```
identifier : PlanningGanttChartBarTypes
index : b
definition : data { ' In Use' , ' In Maintenance' }
```

ELEMENT PARAMETER:

```
identifier : PlanningGanttChartStartingWeek
index domain : w
range : Weeks
definition : w
```

PARAMETER:

```
identifier : PlanningGanttChartDuration
index domain : (r,w,b)
```

在周期滚动程序之后，0-1 参数 OverallLineUsagePlanning(f, p, w) 和 OverallMaintenancePlanning(f, p, w) 更新至包含规划期的首周规划信息，正如数学程序所产生的。这便是更新相应的持续时间参数所需要的第一周信息，而持续时间参数是用来重新绘制 Gantt 表。一旦持续时间参数得以更新，AIMMS 便会自动更新 Planning Overview 页面的 Gantt 表。

现在你需要在模型（如图 12.17 所示）的 Planning Overview 节里插入新的程序 UpdatePlanningGanttChart(iw)。它的变量 iw 应被声明为集 weeks 里面的元素参数，**Property** 属性为 Input。它的 **Body** 属性应该包含下列的语句：

```
PlanningGanttChartDuration(f, p, iw, ' In Use' ) := 1 onlyif
(OverallLineUsagePlanning(f, p, iw) and not OverallMaintenancePlanning(f, p, iw));
PlanningGanttChartDuration(f, p, iw, ' In Maintenance' ) := 1 onlyif
OverallMaintenancePlanning(f, p, iw);
```

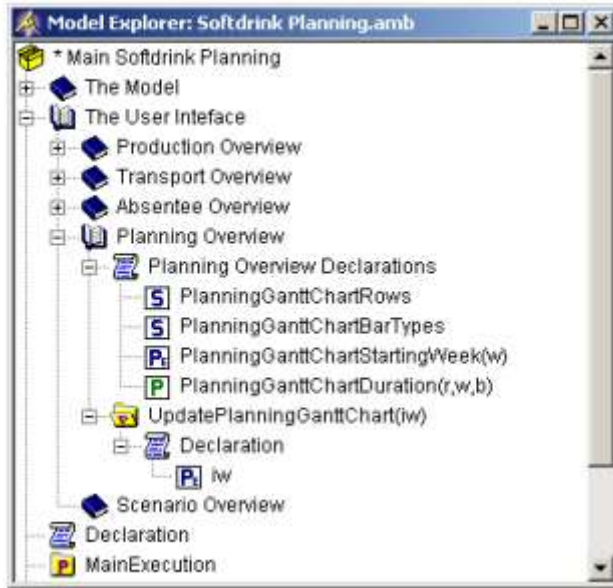


图 12.17 模型树里的 Planning Overview 节

上述的 UpdatePlanningGanttChart (iw) 程序需要在周期滚动过程的每一步后执行。由于它和参数 OverallLineUsagePlanning (f, p, w) 及 OverallMaintenancePlanning (f, p, w) 的连接, 插入这个程序调用作为程序 RegisterInOverallPlanning (iw, ip) 里面的最后语句是合理的, 正如图 12.18 所示。

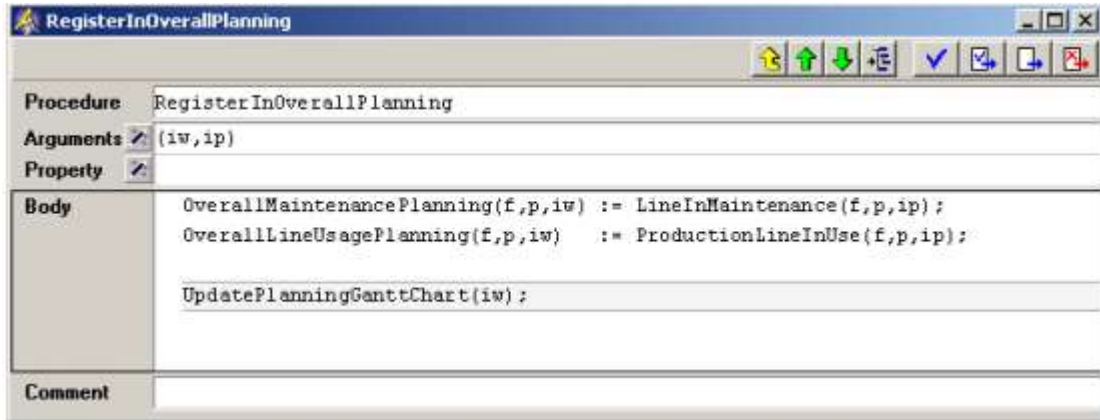


图 12.18 程序 RegisterInOverallPlanning 的 Body 属性

通过遵循以下列出的步骤, 你现在可以创建 Planning Overview 的维护规划 Gantt 表。

- 在 Edit 模式下打开 Planning Overview 页面
- 在工具栏里按 New Gantt Chart 按钮 
- 拖放一个符合你页面的 Gantt 表大小的矩形, 然后
- 使用 Wizard 按钮  完成如图 12.19 所示的 Gantt Chart 对话框

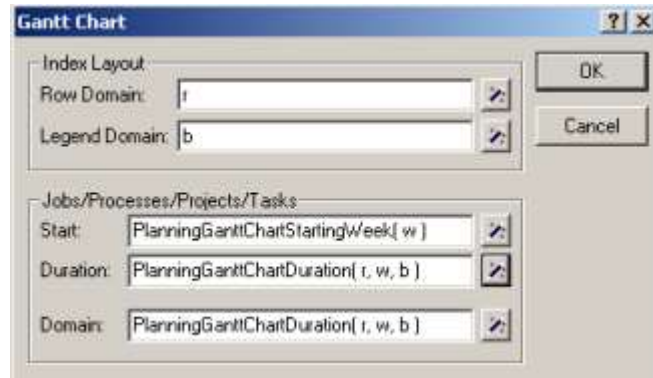


图 12.19 维护规划 Gantt 表的 Gantt Chart 对话框

规划 Gantt 表的 x 轴和之前讨论的假期规格 Gantt 表关于周、月和年，应该是一样的。要改变 Gantt 表的 x 周的目前时间引用，你应该执行以下步骤：

- 选择 Gantt 表
- 打开它的 **Properties** 对话框
- 选择 **X-axis** 标签
- 选择 Real-Time Calendar 作为 Type of X-axis
- 如图 12.20，勾选 ‘Weeks’，‘Months’ and ‘Years’
- 输入” 2000-06-26”（包含引号）作为 ‘Reference Time’
- 选择 EndDateOfCalendar 作为 ‘Right Bound’，然后
- 按 **Apply** 按钮

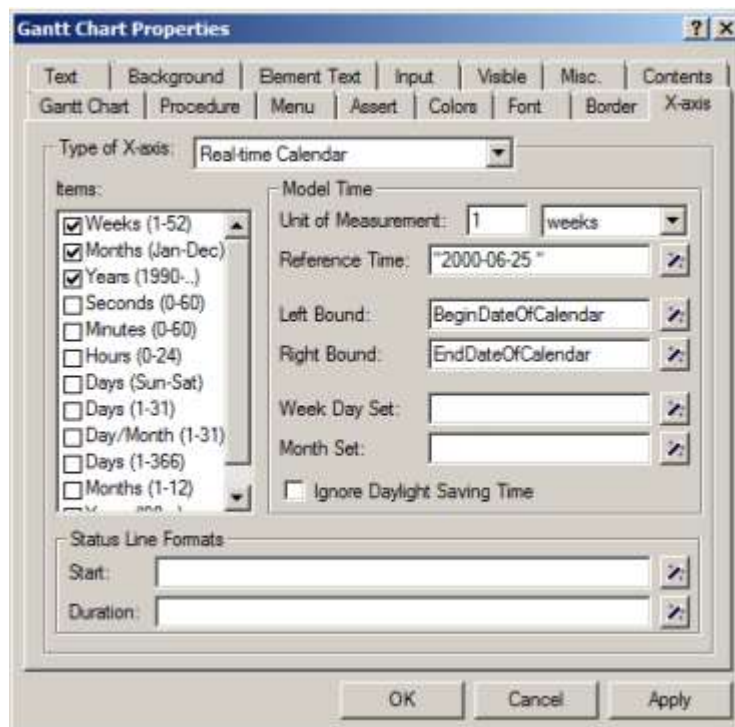


图 12.20 Gantt Chart Properties 对话框的 X-axis 标签

12.3.2 完成页面

一旦你完成了规划概要 Gantt 表，剩下要做的就是增加如图 12.21 所示的三个复合表。以在 Absentee Overview 页面增加这种表格的同样的方式，增加三个显示标识符的表格

- OverallMaintenancePlanning(' Eindhoven' , p, w),
- OverallMaintenancePlanning(' Haarlem' , p, w), and
- OverallMaintenancePlanning(' Zwolle' , p, w)

你的屏幕上的页面还不能如图 12.21 所示那样。如果你喜欢的话，你可以增强你的页面，比如对齐数据对象，增加文本对象和矩形，和改变字体大小和颜色。

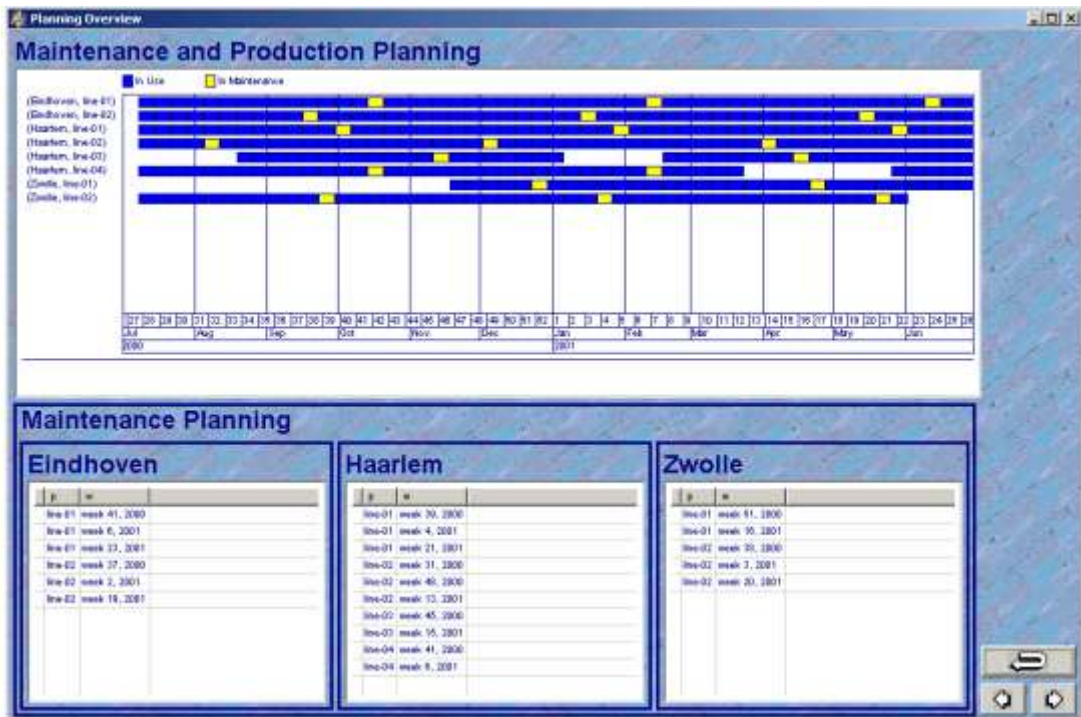


图 12.21: 完整的 Planning Overview 页面

12.4 Scenario Overview 页面

与其创建这个页面，你将完全导入它，如图 12.22 所示。

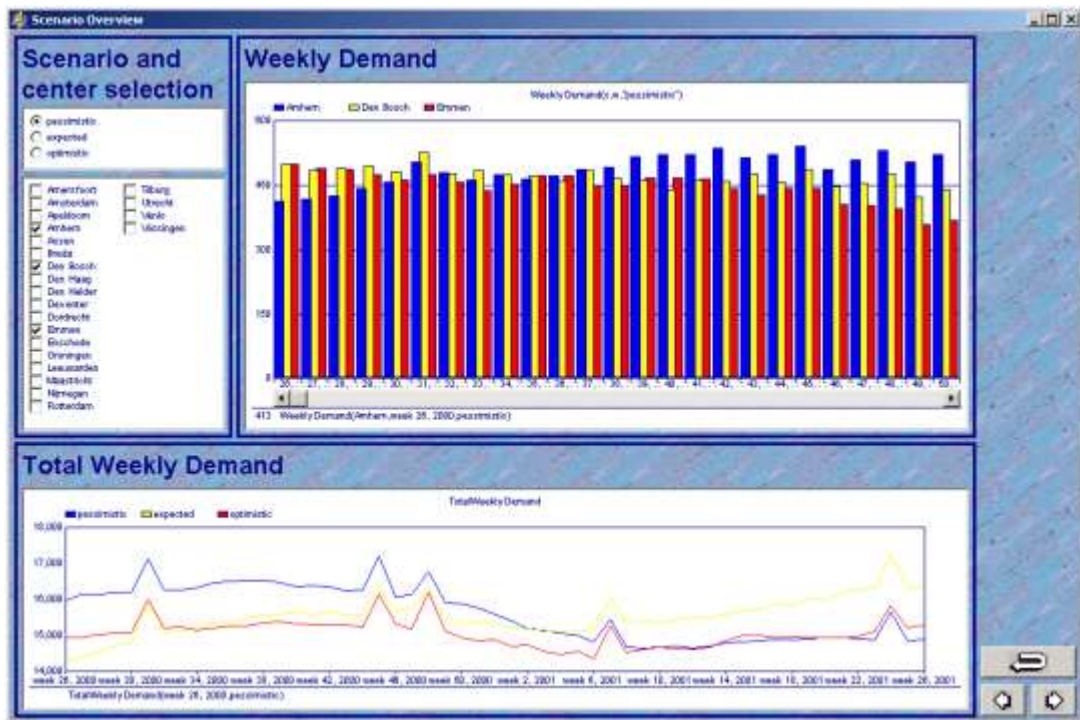


图 12.2.22 完整的 Scenario Overview 页面

在导入 Scenario Overview 页面之前，你需要导入这个页面用到的以下模型声明。为了导入这些模型声明，你需要执行以下步骤：

- 在模型树里选择 Scenario Overview 模型节
- 在 **Edit** 菜单里，选择 **Import** 命令
- 在 **File Selection** 的对话框的 ‘Sections’ 子目录里选择 ‘Scenario Overview.amb’ 文件
- 按 **Open** 按钮，然后
- 按 **OK** 按钮以确定导入

要导入完整的 Scenario Overview 页面，你应该执行以下步骤：

- 按 **F9** 键以打开 **Page Manager**
- 选择 Page Tree
- 从 **Edit** 菜单里选择 **Import** 命令
- 从 **File Open** 对话框选择文件 ‘Pages\Scenario Overview.pag’
- 按 **Open** 按钮
- 在 **Import Page File** 对话框（见图 12.23）按 **Import** 按钮

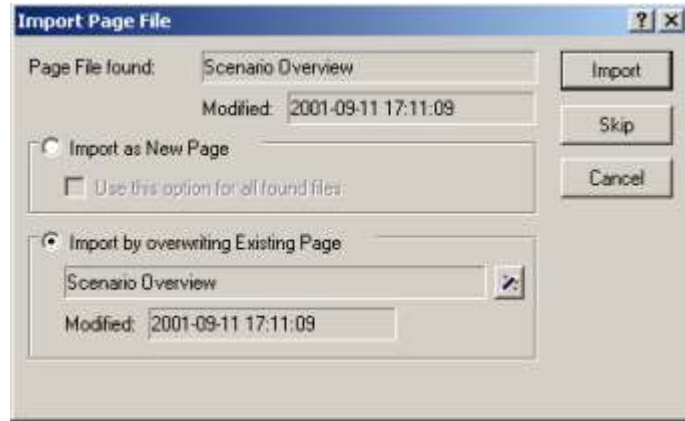


图 12.23: Import Page File 对话框

第十三章 创建用户菜单

在这一章，你将会通过在你的应用中增加菜单栏，以增强终端用户界面。

13.1 菜单管理

菜单栏显示为页面顶部的水平条形，且包含激活命令的弹出菜单。使用点击行为就可以打开菜单。

工具栏是在菜单栏下面的可选的水平条形，且包含一行位图按钮。这些按钮使用户可以很容易用到最经常使用的命令。

弹出菜单包含一系列菜单项目和其他的弹出菜单。它们包含描述命令的文本和激活键盘命令的可选的快捷方式。

分隔符用来构成弹出菜单里的菜单项目。分隔符显示为弹出菜单里的水平分割线或工具栏的按钮直接的空白（见图 13.1）。

User 模式的 AIMMS 页面默认地包含菜单栏和工具栏，如图 13.1 所示。



图 13.1: 默认页面菜单栏和工具栏

13.2 Softdrink 规划菜单栏

一般地，你为了让你的终端用户在 User 模式里使用而设计菜单栏和工具栏。针对开发者的命令，比如提供模型树的使用，不应该在终端用户的页面出现。

当你构建你的菜单栏时，你应该尽可能尝试遵循被广泛接受的做法。而且，如果菜单栏和交叉页面一致时，你的终端用户用起来就会容易。一个典型的例子就是把 **Exit** 命令放在菜单栏的首个菜单的最后一个菜单选项。

你在本教程将会用到的菜单栏结构包含以下 7 个菜单：

- 用于备份、打印和退出的 **File** 菜单
- 用于执行一般编辑操作的 **Edit** 菜单
- 用于储存和获取数据的 **Data** 菜单
- 用于控制周期滚动过程的 **Run** 菜单
- 用于方便进入其他页面的 **Overview** 菜单
- 用于跟踪打开的窗口的 **Window** 菜单
- 用于提供应用方面的帮助的 **Help** 菜单

使用 AIMMS **Menu Builder** 可以创建和规定用户菜单。这个工具层次性地显示包含所有菜单栏和工具栏的树。这个菜单树的外观和感觉类似于其他基于树的 AIMMS 工具。


要创建想要的菜单栏结构，你首先需要通过按 AIMMS 工具栏里的 **Menu Builder** 按钮或按 Ctrl-F9 键以打开 **Menu Builder**，然后才菜单树里打开 Default Page Menubar。这个出事的菜单树如图 13.2 所示。



图 13.2 Menu Builder 及初始的菜单树

初始的菜单树的 Default Page Menubar 和 Default Page Toolbar 是只读的。这个特点由菜单树的禁用图标指示。虽然如此，这些条形仍然可以作为基础构建，你从中可以开始构建你自己的菜单栏和工具栏。在这个教程中，你在创建自己的 Softdrink Planning Menubar 时，将被要求复制和黏贴 Default Page Menu 的几个部分。

要创建你首个菜单栏，你应该执行下列的操作：

- 选择 User Menu Tree
- 按工具栏里的 **New Menubar** 按钮 
- 规定 ‘Softdrink Planning Menubar’ 作为它的名字，然后
- 按 Enter 键以注册这个名字



13.2.1 File 菜单

图 13.3 显示所推荐的包含一个子菜单的 **File** 菜单和五个菜单项目。**Backup** 子菜单和数据的备份有关，而 **Print** 菜单项目打印活跃的窗口的内容。其他菜单项目是很明显的。



图 13.3 所推荐的 **File** 菜单



要创建这个 **File** 菜单，你需要执行以下的操作：

- 选择树的 Softdrink Planning Menubar
- 双击菜单栏图标  以打开这个节点
- 按 **New Menu** 按钮 

- 规定 ‘&File’ 作为新菜单的名字，然后
- 按 Enter 键以注册这个名字

字符串 ‘&File’ 的符号&将自动创建一个由 Alt-F 组合键触发的快捷方式。在&符号后面的字母将在实际菜单（见图 13.3）里加下划线。&符号能放在字符串的任何字母的前面。

要创建 File 菜单的 Backup 子菜单，你应该遵循以下步骤：

- 选择菜单树的 **File** 菜单
- 双击菜单图 , 打开这个节点
- 按 **New Menu** 按钮 
- 规定 ‘Backup’ 作为这个新菜单的名字，然后
- 按 Enter 键以注册这个名字

你的屏幕上的 **Menu Builder** 应该如图 13.4 所示。

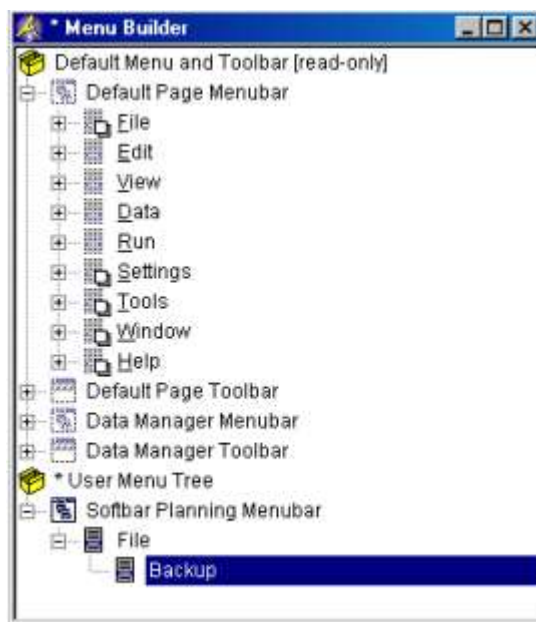








图 13.4: 目前的 **File** 菜单

复制现存的菜单和菜单项目提供两个主要的优势。首先，复制提供了一种创建菜单的快捷方式：你不需要重复输入相应的菜单操作。而且，由于一个菜单项目的更新会自动扩散给其他项目，因此复制菜单项目较容易维护。

File 菜单的所有菜单项目都是现有的菜单项目的复制。请执行与各种菜单项目相关的各组步骤。

- 移至 Default Page Menubar 的 **File-Backups-Data** 菜单
- 同时选择两个菜单项目 ‘Create’ 和 ‘Restore’
- 按工具栏里的 **Copy** 按钮 
- 选择之前创建的 **Backup** 按钮
- 打开它并点击 ‘empty’, 然后
- 从 **Edit** 菜单中选择 **Paste as Duplicate** 命令
- 在 **Backup** 按钮前面按负号标志按钮 , 然后
- 在工具栏里按 **Separator** 按钮 
- 移至 Default Page Menubar 的 **File** 菜单

- 同时选择菜单选项 **Print Setup** 和 **Print**
 - 按工具栏里的 **Copy** 按钮 
 - 选择你刚才创建的分隔符
 - 从 **Edit** 按钮选择 **Paste as Duplicate** 命令
 - 在工具栏里选择 **New Separator** 按钮 
 - 移至 Default Page Menubar 的 **File** 菜单
 - 选择菜单选项 **Exit**
 - 按工具栏里的 **Copy** 按钮 
 - 选择你刚才创建的分隔符
 - 从 **Edit** 按钮选择 **Paste as Duplicate** 命令
- 完整的 **File** 菜单应该如图 13.5 所示

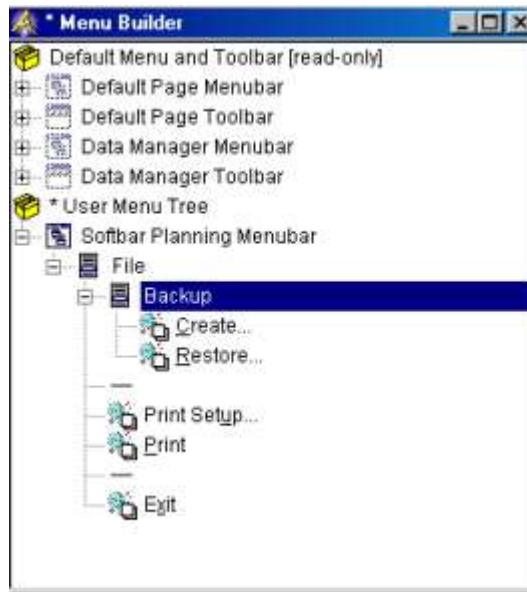



图 13.5: 完整的 **File** 菜单

13.2.2 Edit 和 Data 菜单

要创建的 **Edit** 和 **Data** 菜单应该和 Default Page Menubar 里已经存在的菜单相同。要创建 **Edit** 和 **Data** 菜单，你应该遵循以下步骤：

- 移至 Default Page Menubar
- 同时选择两个菜单项目 **Edit** 和 **Data**
- 按工具栏里的 **Copy** 按钮 
- 从 Softdrink Planning Menubar 选择 **File** 菜单
- 确定它是关闭的，然后
- 从 **Edit** 菜单选择 **Paste as duplicate** 命令控制

包含新的 **Edit** 及 **Data** 菜单的 Softdrink Planning Menubar 如图 13.6 所示

13.2.3 Run 菜单

Run 菜单包含控制周期滚动过程的命令。没有标准的操作，而且你需要创建这些菜单选项以及它们的操作。首先，你应该使用以下步骤，创建三个菜单选项和一个分隔符，如图 13.7 所示：

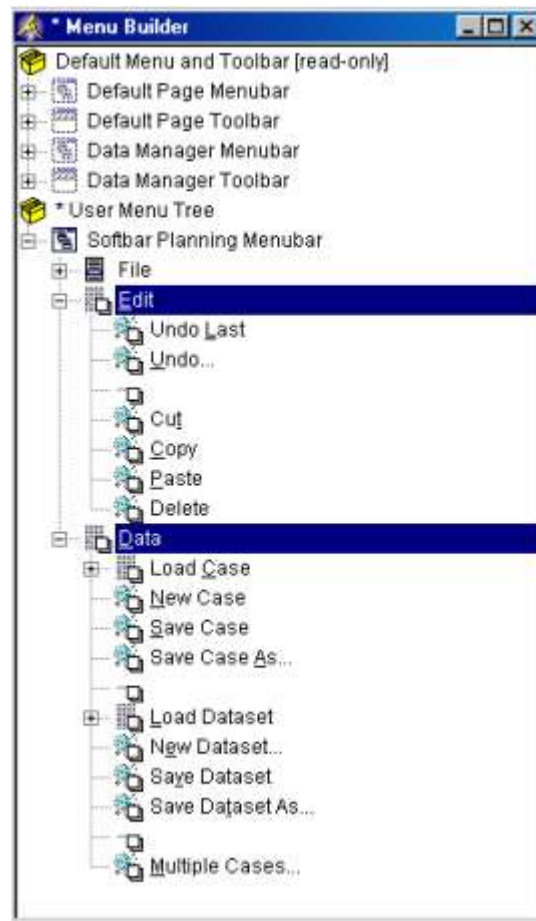


图 13.6: 新的 File 和 Data 菜单

- 从菜单树选择 **Data** 菜单
- 如果菜单是打开的话，关闭它
- 按 **New Menu** 按钮 
- 规定 ‘&Run’ 作为这个新菜单的名字
- 按 Enter 键以注册这个名字
- 双击它的图标以打开它
- 按工具栏里的 **New Item** 按钮 
- 再次按工具栏里的 **New Item** 按钮 
- 输入 “Run All”（不加引号）作为它的文本
- 按工具栏的 **New Separator** 按钮 
- 再次按工具栏的 **New Item** 按钮 
- 输入 “Run Next”（不加引号）作为它的文本

创建了这三个菜单项目之后，你现在需要设置这些菜单项目被选择时要执行的命令。以下的步骤设置和 Run Next 菜单选项相关的命令：

- 选择 **Run Next** 菜单选项
- 按工具栏的 **Properties** 按钮 
- 按 **Actions** 标签
- 选择“Run”操作
- 按 **Add** 按钮
- 选择“Procedure”选项（不是“Page Procedure”选项）
- 使用 **Wizard** 按钮  以选择程序 RollHorizonOnce
- 按 **Finish** 按钮，然后
- 按 **OK** 按钮

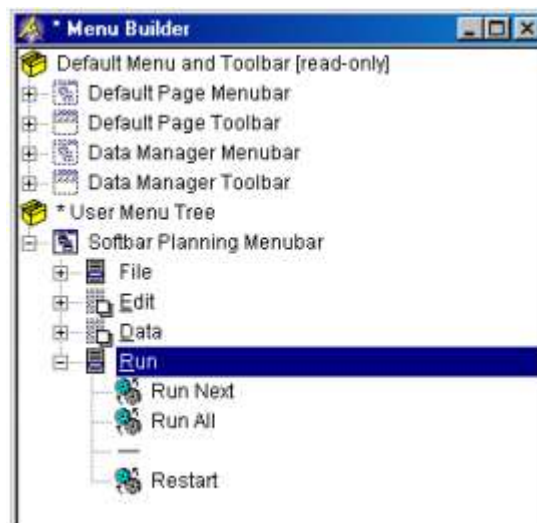


图 13.7: Run 菜单

完整的 **Menu Properties** 对话框的 **Action** 标签应该如图 13.8 所示。

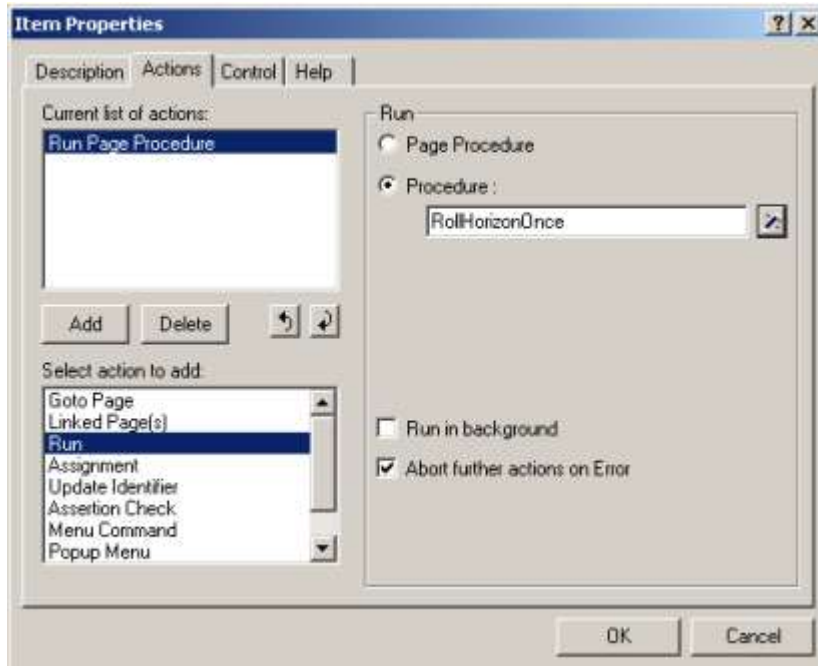





图 13.8: Menu Properties 对话框的 Action 标签

重复以上步骤，把程序 RollHorizonToEnd 链接到 **Run All** 菜单选项。然后再次重复这些步骤，把程序 MovePlanningIntervalToStartOfCalendar 链接到 **Restart** 菜单选项。

13.2.4 Overview 菜单

Overview 菜单提供不同的菜单选项以进入五个概要页面。你不需要分别制定这些菜单选项，你可以利用 Page Manager 的页面结构。

New Navigator 按钮  使你可以增加导航菜单到你的应用上。这些导航菜单有菜单选项，以及可能存在的子菜单，都是指向页面。这些菜单和 **Page Manager** 里相应的页面有着同样的层次结构。因此，当页面树的页面结构改变时，AIMMS 的导航菜单便会自动更新。要创建完整的 Overview 菜单作为导航菜单，你需要执行以下步骤：

- 从菜单树里选择 **Run** 菜单
- 如果菜单是打开的话，关闭它
- 按 **New Menu** 按钮 
- 规定 '&Overview' 作为这个新菜单的名字
- 按 Enter 键以注册这个名字
- 打开新的 **Overview** 按钮
- 按工具栏的 **New Navigator** 按钮 
- 规定“Overview Pages”作为菜单的名字，然后
- 按 Enter 键以注册这个名字

你的屏幕上的菜单树应该如图 13.9 所示。

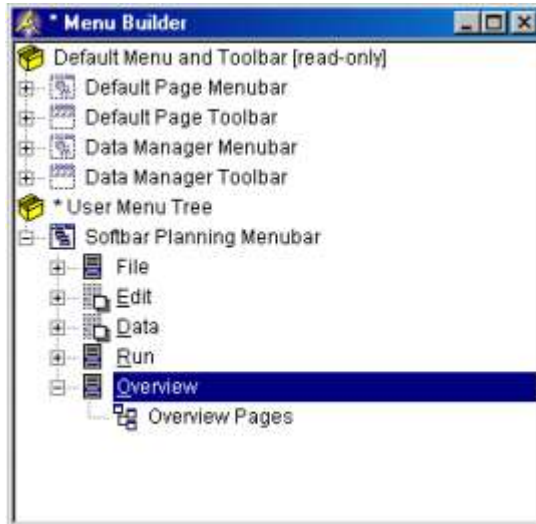


图 13.9: 目前的菜单树

要设置显示在 **Overview** 菜单的页面，你需要执行以下步骤：

- 从菜单树里选择 “Overview Pages” 导航项目
- 按工具栏里的 **Properties** 按钮 
- 选择 **Navigation** 标签
- 在 “Reference Page”（见图 13.10）里的选项中选择 “Other Page”
- 在 “Other Page” 编辑域的右边选择 **Wizard** 按钮
- 选择 Contents 页面
- 按 **OK** 按钮两次

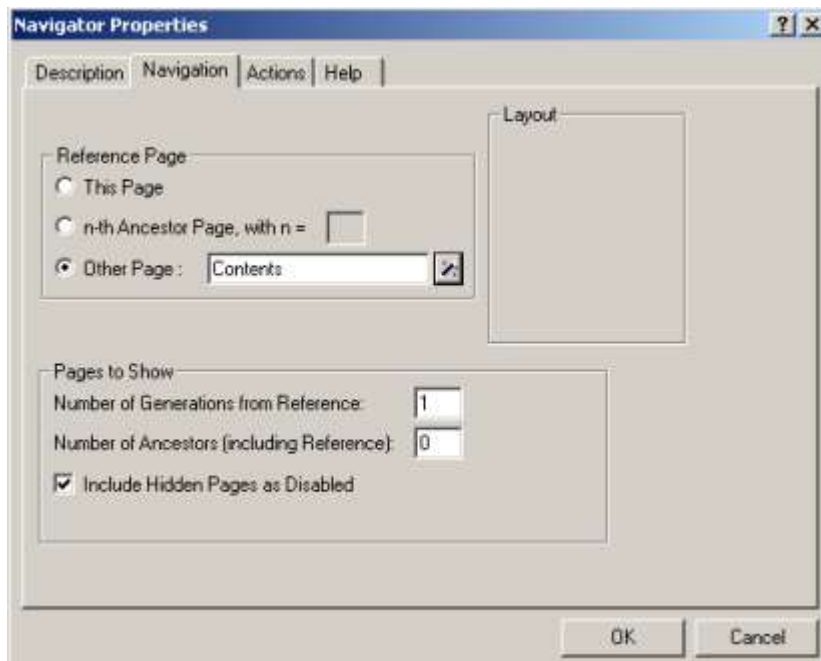


图 13.10: Menu Properties 对话框的完整的 Navigation 标签

最终的 **Overview** 菜单如 13.11 所示。




图 13.11 Overview 菜单

13.2.5 Window 菜单

Softdrink Planning Menubar 的 **Window** 菜单和 Default Page Menubar 的 **Window** 菜单是一样的。

要从 Default Page Menubar 复制 **Window** 菜单，你需要执行以下操作：

- 从 Default Page Menubar 选择 “Menu” 菜单
- 按工具栏的 Copy 菜单 
- 从 Softdrink Planning Menubar 选择 **Overview** 菜单
- 确认它是关闭的，然后
- 从 Edit 菜单选择 **Paste as Duplicate** 命令

13.2.6 Help 菜单

Help 菜单的内容如图 13.12 所示。第一个菜单选项将打开 AIMMS Help 文件。第二个菜单选项在 PDF 视图里显示模型的总结。第三个菜单选项打开一个包含一些应用的信息的 “About” 对话框。

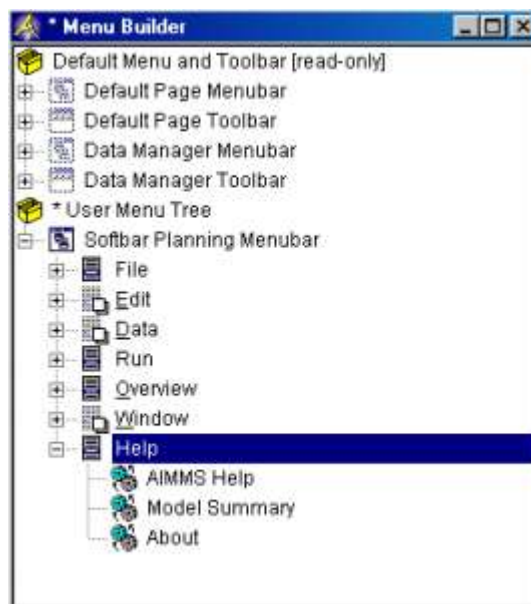




图 13.12: Softdrink Planning Menubar 的 Help 菜单

现在你应该可以自己创建 **Help** 菜单和它的三个菜单选项。注意，这三个菜单选项应该利用工具栏的 **New Item** 按钮  从零开始创建。

与其复制第一个菜单选项，你需要通过执行以下操作来直接设置菜单命令。

- 选择“AIMMS Help”菜单选项
- 按工具栏的 **Properties** 按钮 
- 按 **Actions** 标签
- 选择“Menu Command”选项
- 按 **Add** 按钮
- 选择“Help-Contents and Index”输入（见图 13.13），然后
- 按 **OK** 按钮

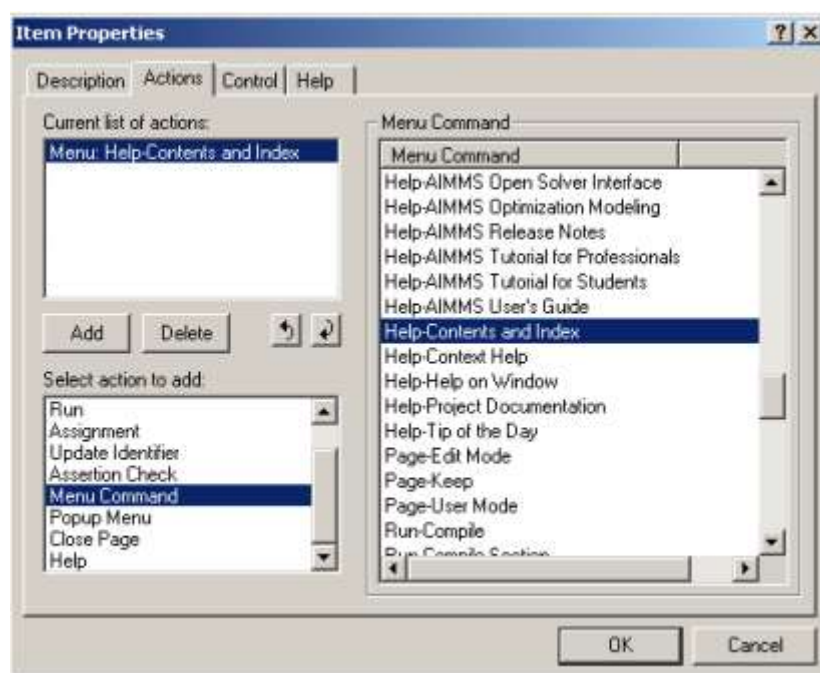


图 13.13: Menu Properties 对话框的 Action 标签

要指定 **Model Summary** 菜单命令，你需要声明一个辅助的 AIMMS 程序。为了保持你的模型树的完好组织，首先你应该创建一个名为 `Softdrink Planning Menubarunderneath the Scenario Overview` 的新模型部分，然后在这一部分创建一个程序 `ShowModelSummary`，如图 13.14。这个程序需要有以下 **Body** 属性：

```
ShowHelpTopic("section. 3. 4", "Tutorial/AIMMS_tutorial_for_professionals.pdf");
```

注意，你可能需要改变作为函数 `ShowHelpTopic` 的第二个变量的教程文件的路径。



图 13.14: 模型树的 Softdrink Planning Menubar 部分

你现在可以执行以下步骤，把你刚才创建的程序链接到 **Model Summary** 菜单命令：

- 选择” Menu Summary” 菜单选项
- 按工具栏的 **Properties** 按钮 
- 按 **Actions** 标签
- 选择” Run” 操作
- 按 **Add** 按钮
- 选择” Procedure” 选项
- 使用 **Wizard** 按钮 ，选择程序 ShowModelSummary
- 按 **Finish** 按钮，然后
- 按 **OK** 按钮

Help 菜单的最后一个选项打开一个” About” 对话框，提供关于应用方面的信息，如版本号或版权信息。你可以在 AIMMS 中使用以下操作创建一个对话页面：

- 打开 **Page Manager**
- 创建名为” About Softdrink Planning”（见图 13.15）的新页面，
- 在 **Edit** 模式里打开页面
- 打开 **Page Properties** 对话框
- 在” Style” 下面勾选” Behaves as Dialog”
- 按 **OK** 按钮，然后
- 重置它的大小，给出一个适合大小的对话框



图 13.15: 新的 About Softdrink Planning 对话页面的页面树

你可以在 About Softdrink Planning 对话页面中插入任何你想要的内容。图 13.16 便是一个例子，而且它包含一个 **Close** 按钮，一个图形，还有显示应用信息的文本。这个页面也可以从“Pages”子目录里导入。页面的导入过程在上章的最后一节已经描述了。



图 13.16: About Softdrink Planning 对话框

请执行以下步骤来指定第三个菜单命令：

- 选择 **About** 菜单选项
- 按工具栏的 **Properties** 按钮 
- 按 **Actions** 标签
- 选择“Linked Page(s)”作为要添加的操作
- 按 **Add** 按钮
- 按 **New Page Link** 按钮 
- 选择 About Softdrink Planning 页面（见页面 13.17），然后
- 连续按 **OK** 按钮两次

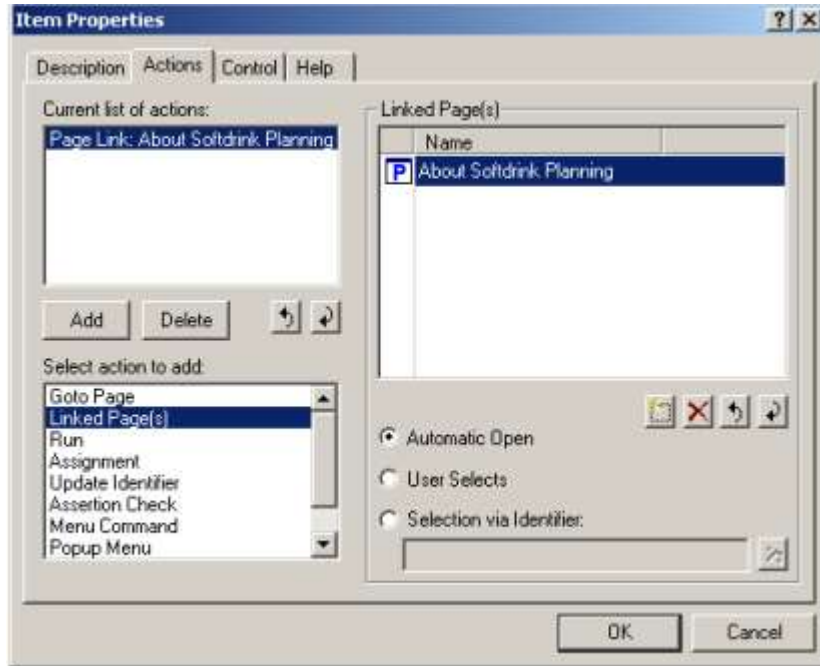


图 13.17: 菜单 **Item Properties** 对话框的 **Action** 标签

13.2.7 把菜单栏链接到页面

你现在一节完成了 Softdrink Planning Menubar 的指定。与其把菜单栏链接到每个单独的页面，把它链接到 Background Color 模板会更方便。这个模板是所有页面共有的，而且页面的菜单栏默认继承模板。

要把菜单栏链接到 Background Color 模板，需要执行以下操作：

- 在 **Edit** 模型里打开 Background Bitmap
- 打开它的 **Page Properties** 对话框
- 选择 **Menu** 标签
- 选择” Other” 作为 Menu Bar 选项（见图 13.18）
- 在” Other” 编辑域的右边按 **Wizard** 按钮 
- 选择 Softdrink Planning Menubar，然后
- 按 **OK** 按钮

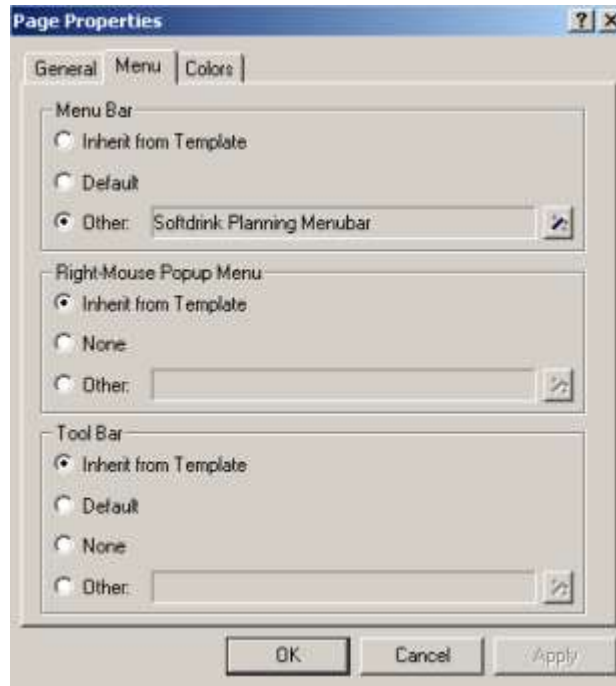



图 13.18: Page Properties 对话框的 Menu 标签

你现在可以使用新创建的菜单栏。通过按工具栏的 **Page User Mode** 按钮  来改变页面模式。这一章里创建的 TheSoftdrink Planning Menubar 应该可以在你所有的页面显示，如图 13.19 所示。

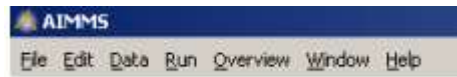


图 13.19: 完整的 'Softdrink Planning Menubar'

第 14 章 数据管理

在这一章，你将学会如何使用容器和\或数据集来管理的你模型数据。这种管理通常是以使用菜单命令为基础。你还将写程序来自动产生 AIMMS 部分的容器。然后这些容器便在多重容器综述里得以观察和比较。

14.1 储存容器的解决方案

容器是在某个时刻的一集数据，并保护所有模型标识符的一个子集的值。这样的—个子集称为容器类型。默认的容器类型是所有标识符的集。容器让你储存中间数据值以便以后查看。你还可以在以后的 AIMMS 会话中使用容器来继续你的工作。

按 **Run Next** 按钮启动周期滚动过程的迭代之后，你可以通过执行以下步骤来储存你的输入和解值在一个新的容器。

- 从 **Data** 菜单选择 **Save Case as...** 命令，
- 在名字编辑域里指定” Solution After First Roll”（不加引号”），然后
- 按 **Save** 按钮（见图 14.1）。

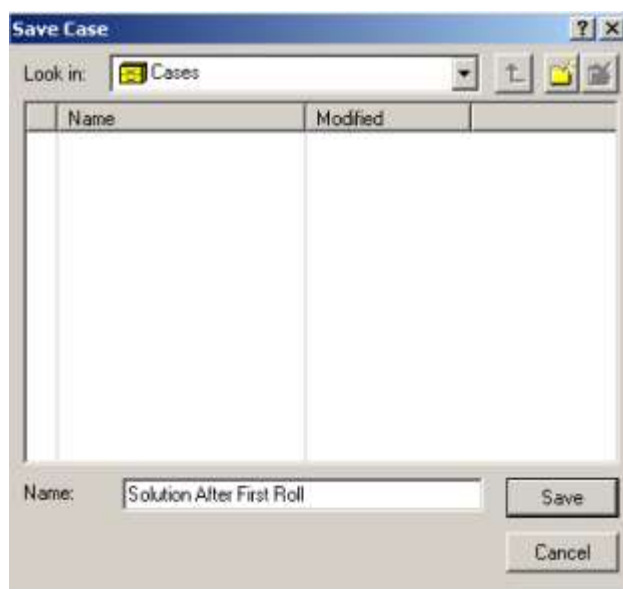


图 14.1: 创建你的首个容器

以下指令关闭并重新打开你的 AIMMS 项目。然后，通过加载你刚刚保存的容器，你将合并你所有数据值。请遵循以下指示：

- 设置当前页面至 **Edit** 模式，改变到默认页面菜单栏
- 从 **File** 菜单选择 **Close Project** 命令
- 再次打开项目
- 从 **Data** 菜单中选择 **Load Case** 子菜单
- 选择 **as Active...** 指令
- 从列表框选择 ‘Solution After First Roll’ 输入，然后
- 按 **Load** 按钮（见图 14.2）



图 14.2: 加载你的首个容器

在 AIMMS 中，你目前处理的所有数据都称为是活动的容器。目前活动的容器的名字显示在 AIMMS 窗口的状态栏的底部，如图 14.3 所示。



图 14.3: AIMMS 状态栏的一部分

14.2 在数据集里储存假日和假期


一个数据集包含某一时间所有模型标识符的子集的数据值。这样的数据集被称为是数据类，而且通常包含和你的应用的某个功能部分相关的模型标识符。

注意，数据集和数据类类似于容器和容器类。主要的差别在于，容器类可以包含一种或多种数据类。因此，容器类可以由多重数据集构建。本教程没有阐述容器类和数据类的这种高级用法，有兴趣的读者可以参考 The User's Guide 的第 17 章。

数据类是通过 **Data Management Setup** 工具来进行管理。要创建描述假期周和官方假日的数据类，你应该执行以下操作：

- 按工具栏的 **Data Management Setup** 工具按钮 
- 在数据管理创建树里选择” Data categories” 节
- 按工具栏的 **New Data Categories** 按钮 
- 指定” Absentee Data” 作为数据类的名字，然后
- 按 Enter 键，注册这个名字

要指定储存在新的数据集的模型标识符，你需要执行以下操作：

- 有需要的话，选择” Absentee Data” 节
- 按工具栏的 **Properties** 按钮 

- 在 **Data Category Properties** 对话框的 Contents 标签按 **Add** 按钮
- 利用 Ctrl 键选择 **HolidayGanttChartDuration** 和 **VacationGanttChartDuration**(两个都在 **Absentee Overview of The User Interface** 里)
- 按 **OK** 键 (你可以比较你的屏幕和图 14.5 的对话框), 然后
- 再次按 **OK** 按钮

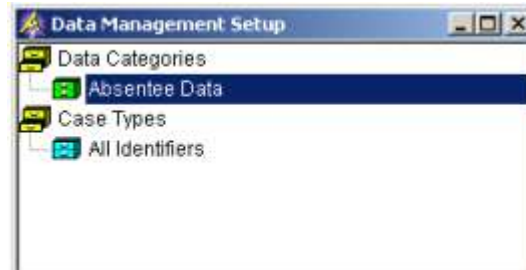


图 14.4 数据管理创建树

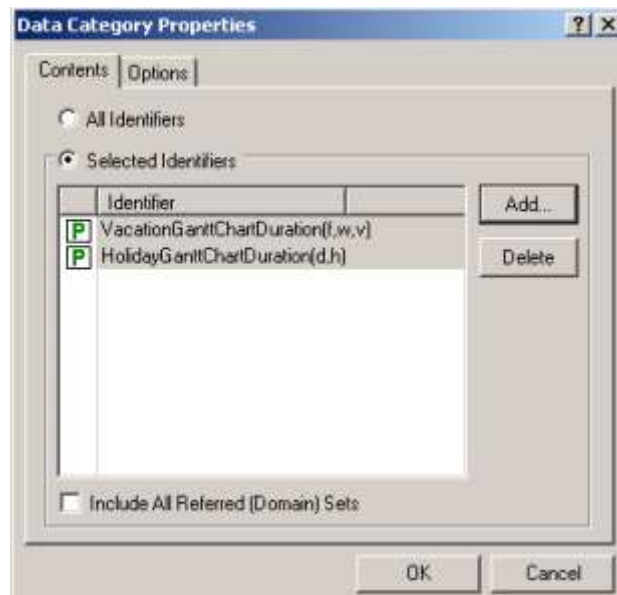


图 14.5: **Dataset Properties** 对话框的 Contents 标签

接下来, 你应该在 **User** 模式下打开 **Absentee Overview** 页面, 并通过点击两个 Gantt 表设置表 14.1 所列的假期周和官方假日。

Vacation Weeks			Official Holidays
Eindhoven	Haarlem	Zwolle	
week 27, 2000	week 30, 2000	week 29, 2000	Dec 25, 2000
week 28, 2000	week 31, 2000	week 30, 2000	Dec 26, 2000
week 29, 2000	week 32, 2000	week 31, 2000	Jan 1, 2001
week 30, 2000	week 33, 2000	week 32, 2000	Apr 15, 2001
week 31, 2000	week 34, 2000	week 33, 2000	Apr 16, 2001
week 32, 2000	week 35, 2000	week 34, 2000	Apr 30, 2001
week 33, 2000	week 36, 2000	week 35, 2000	May 5, 2001
week 34, 2000	week 37, 2000	week 36, 2000	May 24, 2001
week 50, 2000	week 50, 2000	week 50, 2000	Jun 3, 2001
week 51, 2000	week 51, 2000	week 51, 2000	Jun 4, 2001
week 52, 2000	week 52, 2000	week 52, 2000	
week 7, 2001	week 9, 2001	week 8, 2001	
week 8, 2001	week 10, 2001	week 9, 2001	
week 9, 2001	week 11, 2001	week 10, 2001	
week 10, 2001	week 12, 2001	week 11, 2001	

表 14. 1: 假期周和官方假日

要储存你刚才设置的假日和假期数据，执行以下步骤：

- 按工具栏的 **Data Manager** 按钮 
- 从数据数选择” Datasets” 输入
- 选择” Absentee Data” 数据类
- 通过双击图标打开这个数据类
- 按工具栏的 **New Dataset** 按钮 
- 设置” Holiday and Vacation Data” 作为新数据集的名字
- 按 Enter 键，然后
- 从你的页面的 **Data** 菜单 **Save into Selected Node**



图 2: 数据管理树

或者，你也可以利用 **Data** 菜单的 **Save Dataset** 或 **Save Dataset as...** 菜单命令来储存数据集。

要在项目开始时自动加载以上数据集，你需要设置一个加载数据集的程序。请直接在 Softdrink Planning Menubar 模块下面创建一个命名为 Data Management 的模型模块。在这一模块下，声明一个命名为 LoadHolidayAndVacationDataset 的程序。在这个程序下，你应该创建一个元素参数 DatasetReference，它的 **Range** 属性为 AllDatasets(见图 14. 7)。然后，你应该指定以下执行语句为它的 **Body** 属性。

```
if ( not DatasetFind(' Absentee Data' , "Holiday and Vacation Data",
DatasetReference) )
```

```
then DialogMessage( "Error loading holiday and vacation data." );  
return;  
endif;  
DatasetLoadCurrent( ' Absentee Data' , DatasetReference, 0 );
```

要获取上述预先定义的 AIMMS 函数，你首先可以把文本光标放在某个函数名称，然后从鼠标右键的弹出菜单中选择 **Help On** 指令，这样就可以从 AIMMS 中打开 The Function Reference。

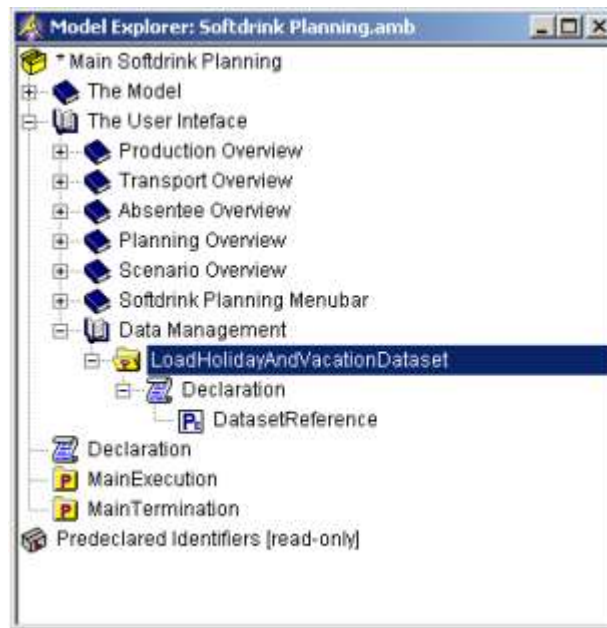


图 14.7: 模型树的 LoadHolidayAndVacationDataset 程序

要使程序 LoadHolidayAndVacationDataset 作为启动程序，当你设置第 10 章最后的启动页面时，你应该遵循以下步骤。相应的 Options 对话框如图 14.8 所示。

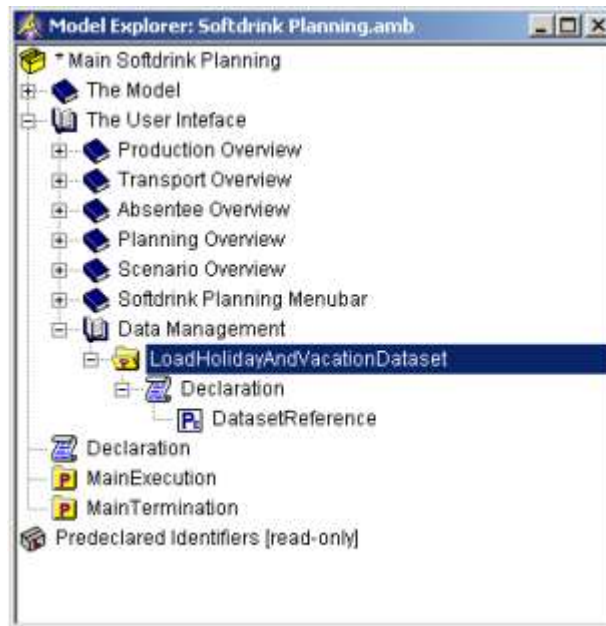


图 14.8: AIMMS Options 对话框

14.3 自动生成容器

在这一节，你将会创建自动生成容器的程序。在这之后，你将开发一种实验来研究规划范围的长度对公司运营的总成本的影响。最后，你将创建一个多重容器对象来浏览和比较这个研究的结果。

在典型的“如果…会怎样”的实验中，你想研究当数据输入改变对你的模型输出的影响。你可以通过一个互动性的会话来做这个实验。如果这个实验涉及面广，而且/或者需要很多 CPU 时间，那么另一种做法就是写一个程序来执行整个实验。把结果储存在由于实验进行而生成的容器是重要的。接下来的段落将给你展示如何使用自动容器储存程序来构建一个多方面的实验。

实验的输出是公司运营的总成本。在实验中，规划范围的长度从 4 周改至 10 周。请在模型树（图 14.9）的 Data Management 部分下面创建一个 Data Management Declarations 声明部分，并在这个声明部分声明以下的标识符。

ELEMENT PARAMETER:

identifier : CurrentPeriod

range : Periods

PARAMETER:

identifier : TotalCostInCurrentPeriod

unit : \$

definition :

sum[s, Probability(s) * (

sum[(f,p), FixedCostDueToLevelChange *

$ProductionLineLevelChange(f, p, CurrentPeriod)] +$
 $sum[f, UnitProductionCost(f) * Production(f, CurrentPeriod)] +$
 $sum[l, UnitStockCost(l) * Stock(l, CurrentPeriod, s)] +$
 $sum[(f, c), UnitTransportCost(f, c) * Transport(f, c, CurrentPeriod, s)])]$

PARAMETER:

identifier : AccumulatedTotalCostInCurrentPeriod

unit : \$

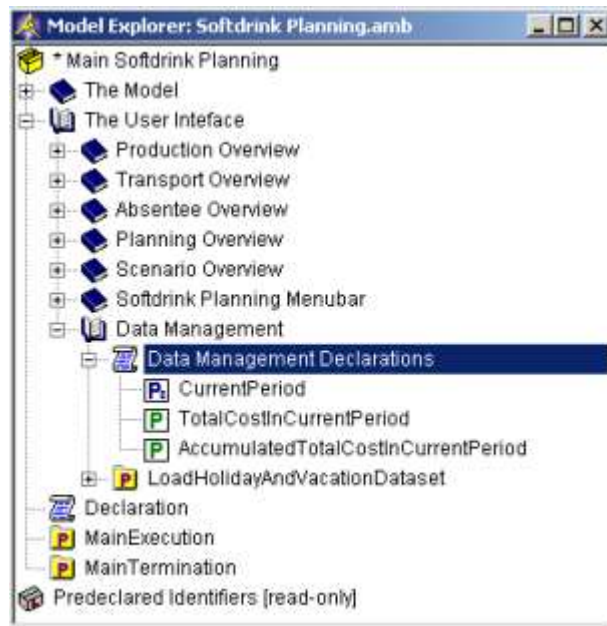


图 14.9: Data Management Declaraions 部分

要创建只包含一个标识符的容器，即 AccumulatedTotalCost，你需要执行以下步骤：




- 按工具栏的 **Data Management Setup** 工具按钮 
- 选择” Cae Types” 节点
- 选择” All Identifiers” 容器类型（见图 14.10）
- 按工具栏的 **New Case Type** 按钮 
- 指定” Accumulated Total Cost 作为容器类型的名字，然后
- 按 Enter 键，注册这个名字



图 14.10: Data Management Setup 树

要指定只有一个标识符储存在” Accumulated Total Cost” 容器类型，请执行以下步骤：

- 有需要的时候，选择” Accumulated Total Cost ” 容器类型
- 按 **Properties** 按钮 ，打开 **Case Type Properties** 对话框
- 选择 **Contents** 标签
- 按 **Add** 按钮
- 选择参数 `AccumulatedTotalCostInCurrentPeriod`（见图 14.11），然后
- 按 **OK** 按钮两次

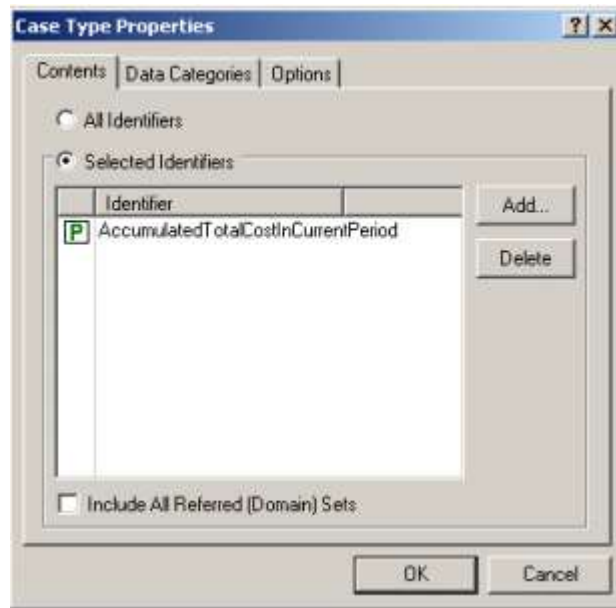


图 14.11: **Case Type Properties** 对话框的 **Contents** 标签

接下来你需要创建如图 14.12 所示的名为 `SaveCase(CaseName)` 的程序。使用 **Argument** 向导，声明 `Casename` 为字符串参数，属性为 'Input'。而且，声明一个局部元素参数 `CaseReference`，**Range** 属性为 `Allcases`。新程序的 **Body** 属性应该输入如下：

```

if ( not CaseFind( CaseName, CaseReference ) ) then
if ( not CaseCreate( CaseName, CaseReference ) ) then
DialogMessage( "Error creating case." );
return;
endif;
endif;
CaseSetCurrent( CaseReference );
CaseSave( 0 );

```

正如之前所说，你可以在 *The Function Reference* 中找到事先定义的 AIMMS 函数的解释。

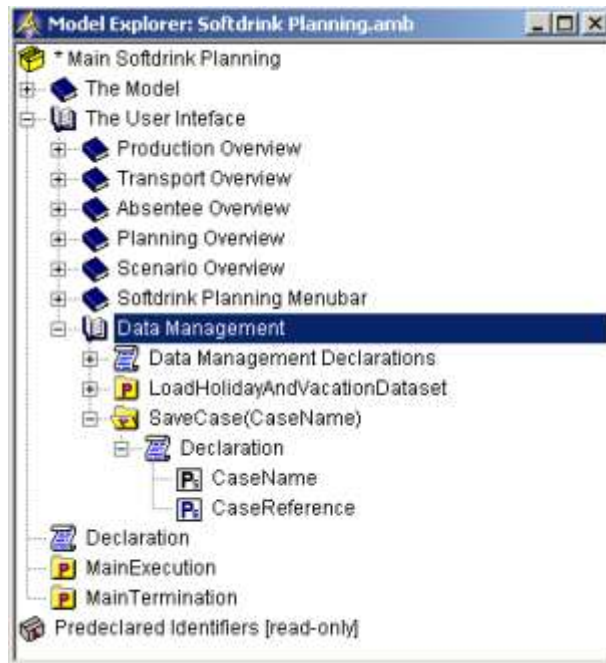


图 14.12: 模型树的 SaveCase 程序

终于，你现在可以设置 Data Management 部分的 RunExperiment 程序，如图 14.13 所示。这个程序的内容很广泛，但大部分是显而易见的。注意之前在以下 **Body** 属性设置的 SaveCase 程序：

```

CurrentDefaultCaseType := ' Accumulated Total Cost' ;
NumberOfPeriodsInPlanningInterval := 4;
repeat "outer-loop"
MovePlanningIntervalToStartOfCalendar;
AccumulatedTotalCostInCurrentPeriod := 0;
CurrentPeriod := FirstPeriodInPlanningInterval;
while ( LastWeekInPlanningInterval < LastWeekInCalendar ) do "inner-loop"
RollHorizonOnce;
AccumulatedTotalCostInCurrentPeriod += TotalCostInCurrentPeriod;
PageRefreshAll;
break "inner-loop" when ( LeastCostPlan.ProgramStatus <> ' Optimal' );
endwhile;
for ( t | t > FirstPeriodInPlanningInterval ) do
CurrentPeriod := t;
AccumulatedTotalCostInCurrentPeriod += TotalCostInCurrentPeriod;
endfor;
SaveCase( FormatString( "Length %n", NumberOfPeriodsInPlanningInterval ) );
break "outer-loop" when ( NumberOfPeriodsInPlanningInterval = 10 );
NumberOfPeriodsInPlanningInterval += 1;
endrepeat;
CurrentDefaultCaseType := ' ' ;

```


模型树的完整的 Data Management 部分应该如图 14.13 所示。

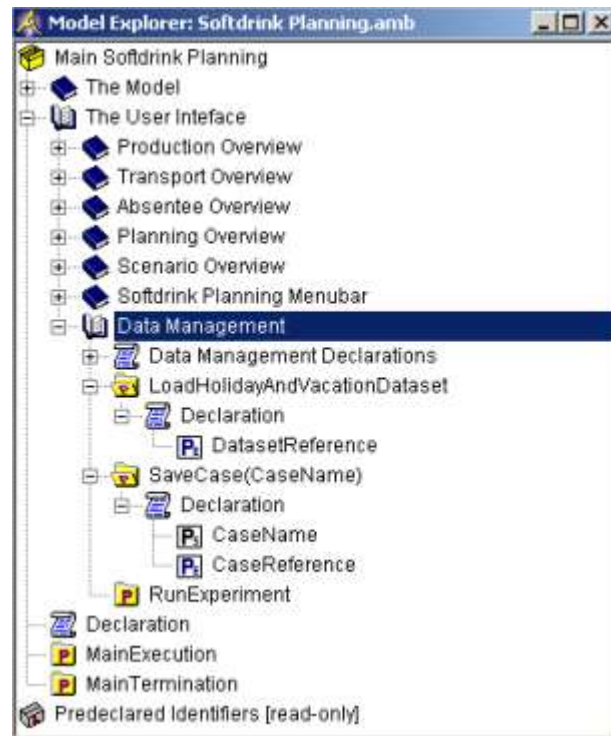


图 14.13: 最终的 Data Management 部分

上述实验的执行需要一点时间，而且可能运行一个晚上。为了启动这样的彻夜运行，你应该执行以下操作：

- 选择模型树的 RunExperiment 程序节点，然后
- 从鼠标右键的弹出菜单选择 **Run Procedure**


当实验结束时，容器树里应该有几个容器生成，如图 14.14 所示。如果你没有运行这个实验，你可以从你的项目目录的 ‘Cases and Datasets’ 目录导入容器。



图 14.14: 实验之后的容器树

现在你可以遵循以下步骤，创建一个表格显示实验中生成的所有容器的 AccumulatedTotalCost 参数值：

- 在页面树底部创建新的页面

- 输入 ‘Multiple Case Overview’ 作为它的名字
- 按 Enter 键，注册这个名字
- 在编辑模式下打开新的页面
- 按工具栏的 **New Table** 按钮 
- 拖放页面的矩形
- 选择参数 AccumulatedTotalCostInCurrentPeriod



要把这个表格转换为多重容器对象，你应该如下操作：

- 打开表格对象的 **Table Properties**
- 有需要的话，选择 **Table** 标签
- 勾选 ‘Multiple Case Object’ 选择框（见图 14.15），然后
- 按 **OK** 按钮



图 14.15: 创建一个多重容器表

这个表格应该可以用空列扩展。要设置多重容器选择，你应该执行以下步骤：

- 按工具栏的 **Page User Mode** 按钮 
- 从 **Data** 菜单选择 **Multiple Cases...** 指令
- 在 **Select Multiple Cases** 对话框的左边列表框选择容器” Length4” 至” Length 10”。
- 按 **Move Right** 按钮 ，把选择的容器移至右边的列表框（见图 14.16），然后
- 按 **OK** 按钮

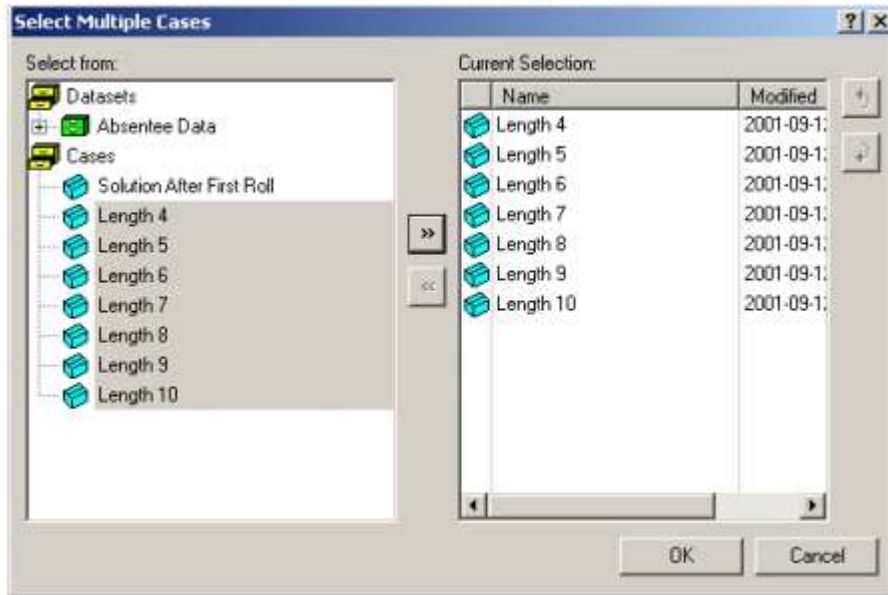


图 14.16: Select Multiple Cases 对话框

设置了多重容器选择后，AIMMS 将自动从容器中加载需要的数据，并完成如图 14.17 所示的表格。

	Length 4	Length 5	Length 6	Length 7	Length 8	Length 9	Length 10
AccumulatedTotalCostInCurrentPeriod	2071566		2074931		1990153	2033005	1978466

图 14.17: 显示多重容器数据的表格

有意思的是，表格的某些输入时空白的，表示实验的内在循环的某个子问题无法解决。而且，总成本没有随着规划范围的时期数量的增加而单调递减。这个实验似乎表示时期的数量应该大于 10。

A-4 可用的 AIMMS 文档列表

- AimmsUser' s Guide
- AimmsLanguage Reference
- AimmsFunction Reference
- AimmsOptimization Modeling
- AimmsCOM Object
- AimmsExcel Add-in
- AimmsMulti-Agent And Web Services
- AimmsOpen Solver Interface
- AimmsTutorial For Beginners
- AimmsTutorial For Professionals