第3章 初级GPSSWorld语言

随着系统仿真技术和计算机技术的发展，人们逐渐想要使用计算机来对系统进行仿真，初期人们采用计算机高级语言如 Fortran、C、Visual C、Visual Basic 等来对系统进行仿真，但是由于要详细描述各类基本事件的发生及其处理情况，并规定各类事件的处理顺序，即使是一个很简单的系统，程序也会很复杂，增加了仿真的难度，有时为了统计出必要的数据，在程序的各段还要加入各种统计所需的语句，最后还必须规定各种统计输出的打印格式。这无疑使得更多的人望而却步。随着系统仿真的进一步发展，人们开始将仿真程序中常用的程序段编成子程序或过程，用于仿真的各种不同的问题中，例如，将队列管理的相关功能（如何进入队列排队、如何统计当前队长及平均队长等）编制成一个名为 QUEUE 的过程，只要仿真的系统中有队列，就可以直接调用 QUEUE 这一过程。也有人将某一大类的仿真问题，编写成一个通用的主程序，用户只需将必要的参数填进去，经执行就能得到所需要的结果，这样专用的计算机仿真语言就出现了。

计算机仿真语言的种类有很多，但是它们都有一些其他高级语言所没有的优点：

（1）通用性好，提供了常用的功能模块，大大减少了编程的工作量。

（2）模块设计原则与要模拟的过程相仿，与用户的需求十分相近。

（3）具有标准输出，可以满足大部分用户的需要，可以省去输出设计的工作。

本书讨论的 GPSSWorld 是在 GPSS 语言的基础上推出的基于 Windows 平台的可视化仿真集成平台，GPSS 全名为 General Purpose Simulation System，即通用仿真系统，它是专用离散系统仿真语言中应用最为广泛的一种。GPSS 的全部目的就是为了预测现实世界中复杂系统的行为——过去许多代价高昂的项目之所以失败是因为缺乏对最终结果的精确预见，它的最早版本由戈登于 1961 年发表。GPSSWorld 以 GPSS 为基础极大地扩展了其功能和应用范围，它将可视化辅助编程、命令菜单辅助、系统动态图形输出、报表输出。参数方差筛选实验和参数优化实验等进行了集成，本书将以 GPSSWorld 为基础来讨论离散系统仿真。

真实系统在 GPSSWorld 中的表示称为 GPSSWorld 模型，分 GPSSWorld 框图式模型和 GPSSWorld 程序式模型两种，一般情况下，应是先绘制系统的 GPSSWorld 框图式模型（类似于高级语言的程序框图），它有结构清晰、一目了然、方便修改等优势，但它是不能被计算机所执行的，若要能被计算机执行，还必须要将 GPSSWorld 框图式模型转换成 GPSSWorld 程序式模型才能真正地在计算机上运行。在任何一台计算机上使用 GPSSWorld 的前提是这台计算机的软件中包括 GPSSWorld 处理器（我们可以通过安装 GPSSWorld 语言来达到这个目的）。GPSSWorld 处理器是一种计算机程序，它用解释方式

将系统模型(用 GPSSWorld 语言编写的程序)翻译成计算机可以直接执行的语句,并予以执行。

本章首先介绍了 GPSSWorld 的运行环境,GPSSWorld 中有关实体的分类,GPSSWorld 程序的基本结构、语句组成及自定义符号,接着按实体的分类分别介绍 GPSSWorld 的基本模块和其他相关的控制语句,然后结合实例着重介绍了如何使用这些语句编写 GPSSWorld 程序的方法和技巧,如仿真预热、仿真长度的控制、优先级的处理、模型状态的初始化、活动实体产生的控制等。最后为了帮助读者加深对 GPSSWorld 程序的了解,还介绍了 GPSSWorld 仿真程序在计算机中的执行过程。

3.1　GPSSWorld 的运行环境

GPSSWorld 虽然是一款商业软件,但用户可以从 http://www.minutemansoftware.com 上直接下载 GPSSWorld Student Version(学习版)进行学习,GPSSWorld 目前的版本为 5.2.2,使用学习版的限制表现为:在仿真模型中最多不超过 180 个模块(不包括命令语句),但这对大多数的仿真已足够了。

下面通过一个实例来了解一下 GPSSWorld 的运行环境。

【例 3.1】 理发店系统仿真。

有一个理发店,只有一位理发师 JOE,顾客以每隔 16±9 分钟(即服从均值为 16,半区间为 9 的均匀分布,在本书中均值为 A,半区间为 B 的均匀分布用 A±B 表示)的速度到达,当顾客到达时若理发师忙,则按先来先服务的规则排队等待直到理发师空闲,否则让理发师开始理发,理发时间为 15±7 分钟,理完发后离开理发店。

这是一个很常见的单窗口排队系统,下面先给出 GPSSWorld 的程序模型。

```
LINE    EQU         1       ;定义一个符号 LINE,它的编号是 1
        GENERATE    16,9    ;顾客以每隔 16±9 分钟的时间间隔到来
        QUEUE       LINE    ;顾客排入名为 LINE 的队中
        SEIZE       JOE     ;若 JOE 闲,则占用 JOE 理发,否则停在队中
        DEPART      LINE    ;离队
        ADVANCE     15,7    ;理发花了 15±7 分钟
        RELEASE     JOE     ;理发完毕释放理发师 JOE
        TERMINATE   1       ;顾客离开系统
        START       20      ;共仿真 20 个顾客
```

读者现在可能无法理解上面的程序,不过完全没有关系,关于程序编写的方法将在以后的章节中详细讨论,这里先通过这个简单的程序来了解一下 GPSSWorld 的语言环境。

若要在 GPSSWorld 中建立计算机仿真模型,首先要打开 GPSSWorld 软件,图 3.1 是打开 GPSSWorld 后的主界面。

GPSSWorld 的主界面由标题栏、菜单栏、工具栏和状态栏组成。系统常用的一些功能都可以在菜单中找到相应的菜单命令,工具栏是菜单的快捷方式,当然用户可以定制自己的工具栏。最下面的状态栏显示了系统运行过程中的一些系统状态或错误信息。

当要编制 GPSSWorld 程序时,首先要新建一个文件,并将程序内容写进去,选择 File|new 菜单命令,如图 3.2 所示。

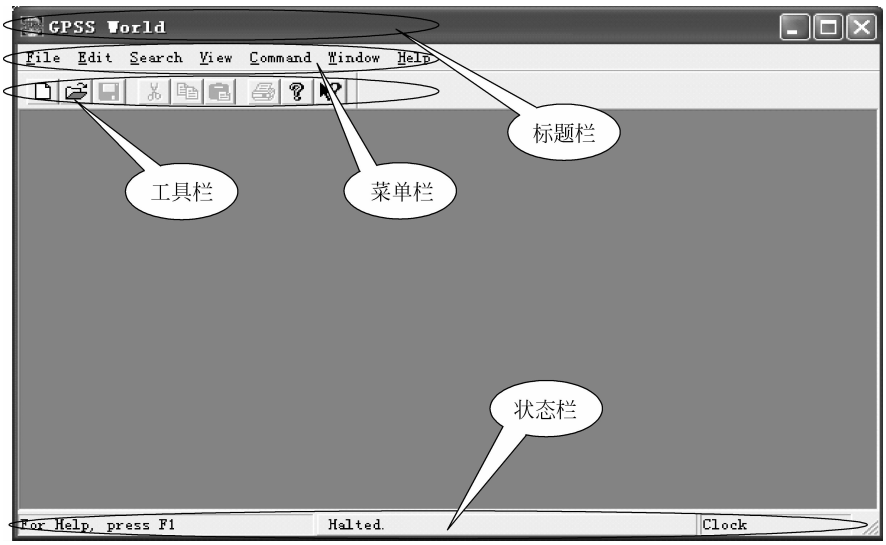

图 3.1 GPSSWorld 主界面

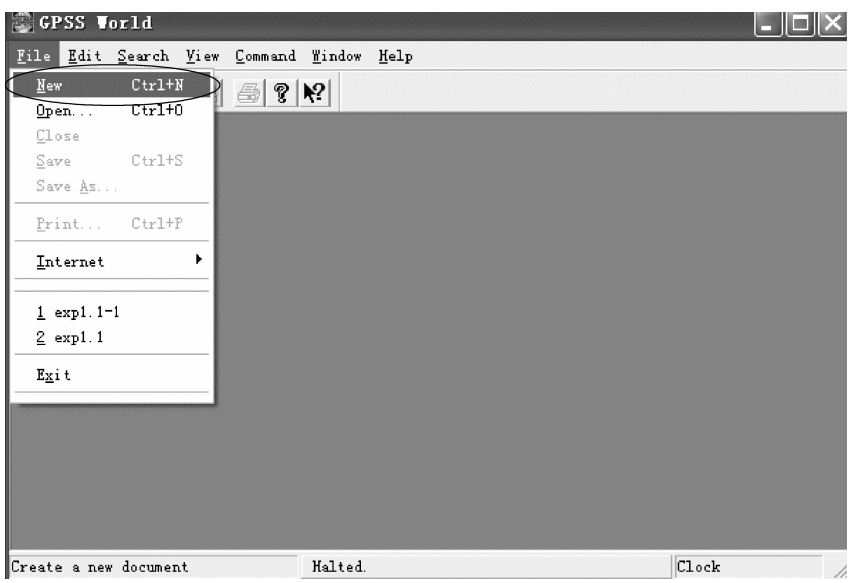

图 3.2 新建文件窗口

选择 New 菜单命令后会出现如图 3.3 所示的对话框，它询问用户想要创建的新文件的类型，如果是要新建仿真模型，应该选择 Model，然后单击“确定”按钮将打开一个空白的模型文件，如图 3.4 所示。

现在可以在这个空白文件中写入程序了，全部写好后单击“保存”按钮将程序保存到指定目录中。当然也可以打开一个已经存在的模型文件，选择 File|Open 命令，从对话框中选择需要编辑运行的文件名。

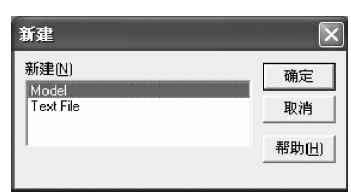

图 3.3 “新建”对话框

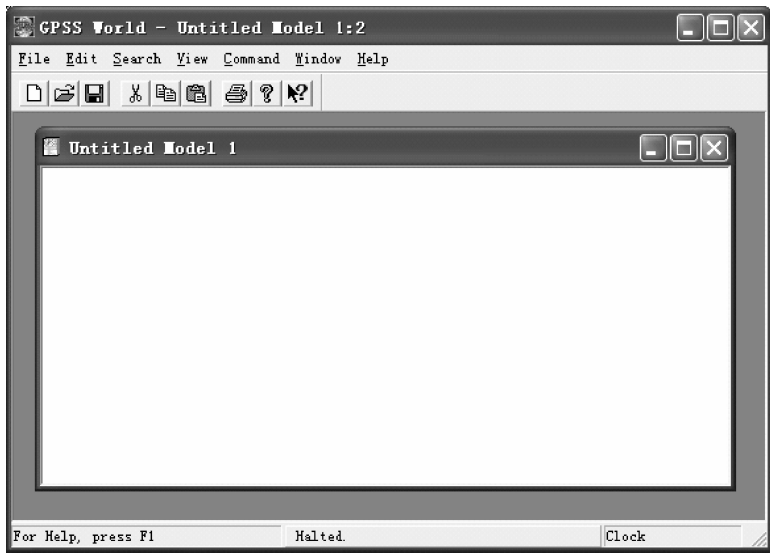

图 3.4　空白的模型文件

程序文件编辑好之后，便可以运行程序查看结果了，选择 Command|Create Simulation 命令，如图 3.5 所示。

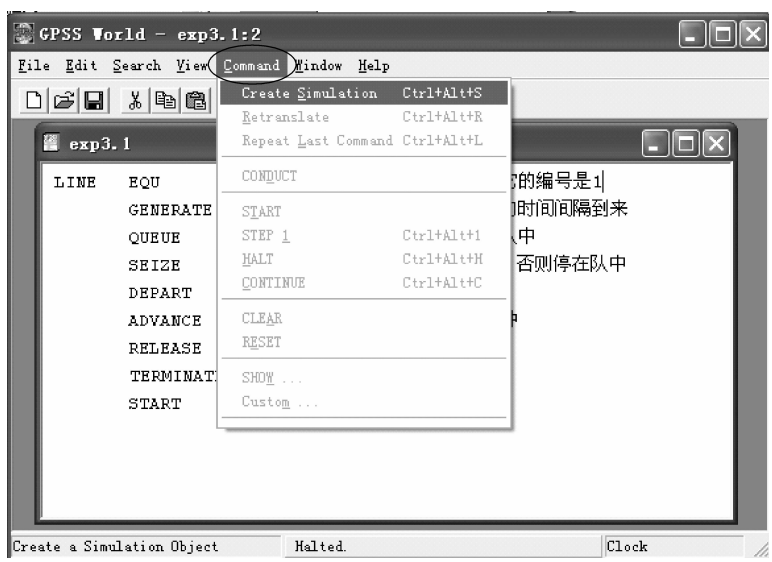

图 3.5　创建仿真

一旦用户选择了 Create Simulation 命令之后，GPSSWorld 处理器就会先对程序进行编译，若有语法错误，模型的运行会中断，并在运行日志文件中给出错误提示，包括错误所在行号和错误的内容等，如图 3.6 所示，读者可以根据系统的提示找出错误的原因，并修改正确。

若程序编译没有错误，则 GPSSWorld 处理器直接运行程序并给出程序运行的报告。前面已经提到过，GPSSWorld 语言会给用户一个标准的输出，这个标准的输出是由系统规

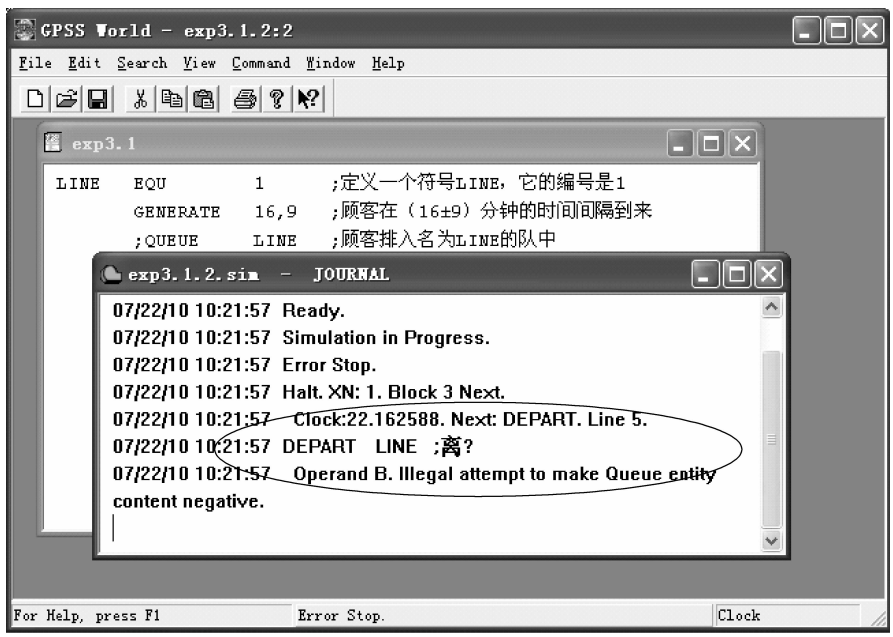

图 3.6 运行日志文件

划，并按指定格式输出的，当然用户也可以根据自己的需要自定义输出格式和内容。对于GPSSWorld读者不仅要会编制程序模型，还要学会去阅读仿真结果的输出，从而能在仿真结果输出中找到自己想要的统计数据。关于GPSSWorld标准输出的具体内容及含义，将在后面的章节中详细讲解，这里先让读者认识一下输出报告，如图3.7所示。

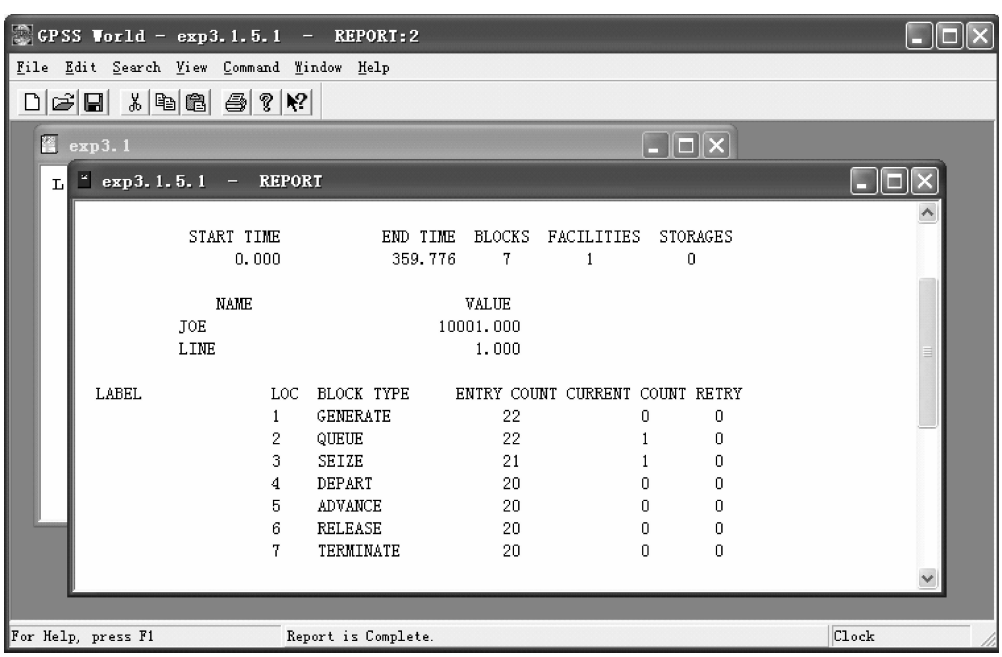

图 3.7 GPSSWorld的标准输出(运行报告)

到这里，这个简单程序的运行就结束了，用户对运行报告中的结果进行分析，从而根据结果来调整系统的模型。

在模型被编译好了之后仍然可以向模型中添加一些交互式的命令语句，它们有的会被立即执行，有的则被放入命令语句的队列中等待被处理。注意：模块语句是不可以交互式地添加的。

GPSSWorld 的程序除了以上自动运行的方式外，还可以单步调试模型程序，从而发现一些从最终运行结果中体现不出来的东西，比如，活动实体的参数值，可以在程序运行的过程中去检测，下面介绍一下单步调试的方法。

(1) 首先确认程序中没有 START 语句，若有 START 语句则模拟会自动运行下去，除非程序有错才会终止。

(2) 选择 Command|Create Simulation 命令，这时程序并没有真正运行，只是将程序编译好进入准备运行的状态。

(3) 选择 Command|STEP 1 命令，这时 GPSSWorld 处理器只执行一行语句，等到用户下次再按 STEP 1 时又执行一条语句，通过这样按一次命令执行一行语句的方式达到单步调试的目的。通过单步调试，用户可以进一步了解程序的运行，进一步了解活动实体在系统中的运动轨迹，并能够随时获得系统中各种资源实体的状态。用户可以选择 Command|SHOW 命令来显示系统中的标准数字属性。如果不需要一步步运行，也可以选择 Command|START 命令，然后在弹出的对话框中输入一个数值，GPSSWorld 处理器就自动运行到仿真结束，并给出运行报告。

GPSSWorld 语言还提供了丰富的 Windows，通过 Window|Simulation Window 命令，可以看到所有的资源实体都有一个 Window 在菜单中，在程序的运行过程中，如果想要获知某资源实体的状态，如队列的当前排队人数等，便可以打开 Queues Window 来获知，如图 3.8 所示。

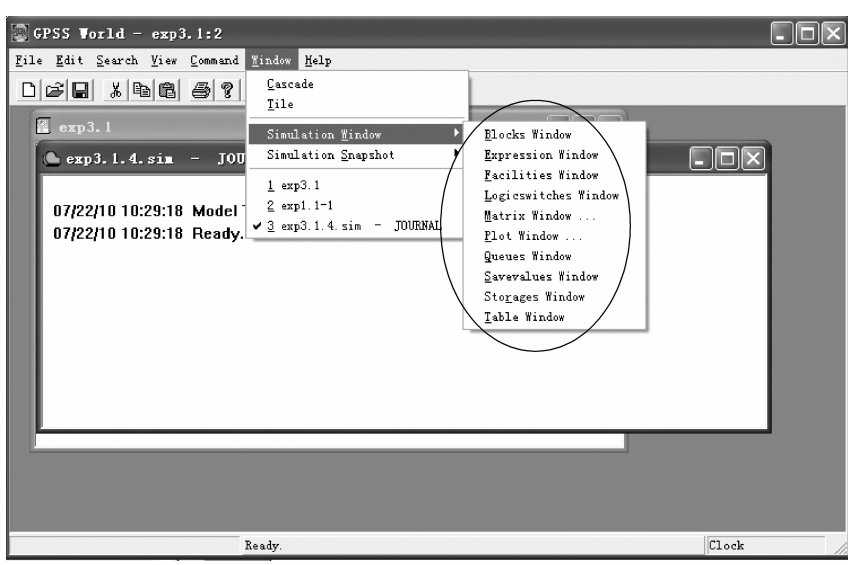

图 3.8　GPSSWorld 提供的观察固定实体状态的窗口

还可以对系统的参数进行设置，选择 Edit|Settings 命令，便可以看到如图 3.9 所示的界面，在这个界面中可以设置、修改一些系统的高级参数。如修改系统中使用的随机数发生器的相关参数(如图 3.10 所示)、自定义输出报告的格式内容(如图 3.11 所示)。

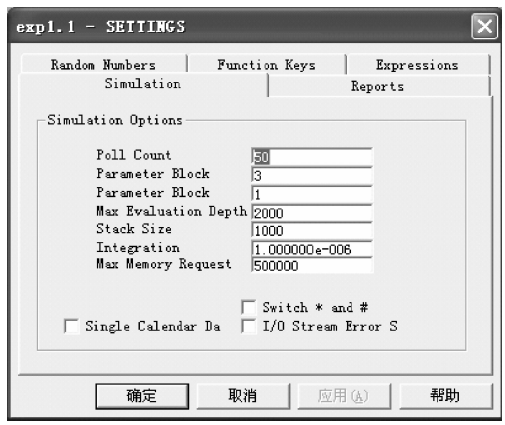

图 3.9　系统设置窗口

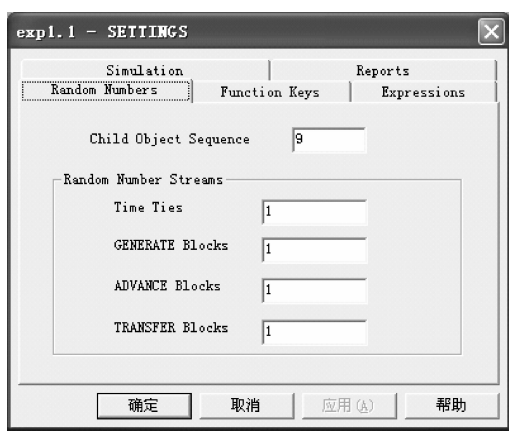

图 3.10　随机数设置窗口

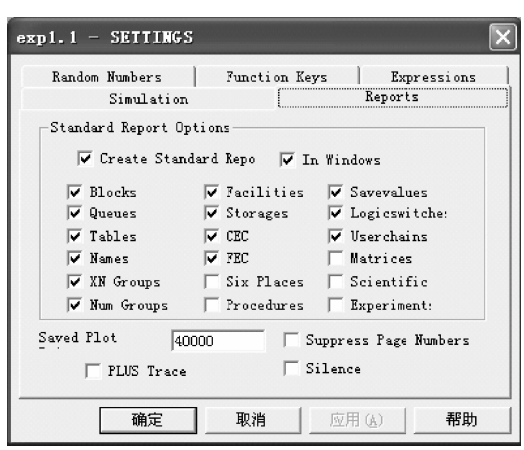

图 3.11　输出报告设置窗口

通过上面的介绍，读者对 GPSSWorld 的运行环境应该有了一个大致的认识。限于篇幅，这里介绍得还远远不够，读者可以自己在使用的过程中慢慢摸索每个菜单及命令的含义。

3.2　GPSSWorld 中实体的分类

在介绍 GPSSWorld 语言之前，有必要对系统中的实体及 GPSSWorld 中的模块进行分类，这不仅可以使读者对各类实体和模块加深理解，而且也可以进一步了解 GPSSWorld 语言的特点，了解它与计算机高级语言的区别，可以使读者尽快入门。根据实体在系统中的作用和行为，同样可以将 GPSSWorld 语言中涉及到的基本实体分成活动实体和固定实体两类。

活动实体是 GPSSWorld 语言中最具代表性的一种实体。活动实体是指系统中可以移动（或活动）的人、设备、设施等，它的行为是能够引起系统中其他实体的行为、属性或状态发生变化，从而导致系统状态发生变化，是驱动 GPSSWorld 程序运行的原动力，是 GPSSWorld 程序的灵魂。因此活动实体在系统中的活动/行为轨迹也就是编制 GPSSWorld 程序的脉络。在阅读 GPSSWorld 程序时，应按每个活动实体在系统中的活动/行为轨迹来阅读程序。

与活动实体相对应的是固定实体。顾名思义，这一类实体其位置在系统中是相对固定的，其不会决定或改变自身的行为、属性或状态，只有在活动实体的作用下才能改变自身的行为、属性或状态。比如车间内的各类车床、公路上的加油站、飞机场的跑道等。GPSSWorld 语言中的固定实体可进一步分为资源实体和统计实体。

（1）资源实体

资源实体是指那些系统中具有资源性能的设备或设施。一般来说，资源实体可以提供各类服务或控制的工作。在 GPSSWorld 语言中属于资源实体的常用固定实体有如下几类：

① 设施（FACILITY）。系统中可提供服务性质工作的人、设备或设施等，如公路上的加油站、飞机场的跑道、理发店的理发师、银行的出纳、食堂的卖饭口等。

② 队列（QUEUE）。系统中需要排队等候之处。这是管理系统中最常见的一种固定实体，几乎所有的管理系统中都存在排队问题。因此队列的应用在 GPSSWorld 中是最多的。

③ 存储器（STORAGE）。系统中可存储动态实体或某种系统元素的设备或设施，如仓库、物料场、煤仓等。有一些看上去像设施的实体，由于考虑问题的方法不同，或者模型建立的思路不同，也可以当成存储器来对待。比如一个有 4 个加油器的加油站可以作为 4 个设施来模拟，也可以用容量为 4 的一个存储器来模拟。

④ 逻辑开关（LOGIC）。系统中仅具有两种工作状态的开关型设备，比如道岔、红绿灯、电开关、机械开关等。

（2）统计实体

统计实体是系统中用来进行统计计算的各类工具，显然统计实体并不是一个具体的设备或元件。它是指各类统计表（TABLE）或用于某些特殊统计的自由变量（保存值）等。

按上述的分类原则，可以将 GPSSWorld 语言中的模块语句（简称模块）分成不同的类型。它们分别是：

① 与活动（流动）实体有关的模块。

② 与设施有关的模块。

③ 与队列有关的模块。

④ 与存储器有关的模块。

⑤ 与逻辑开关有关的模块。

⑥ 与统计实体有关的模块。

3.4 节和 3.5 节将按上述的分类来分别介绍 GPSSWorld 的基本模块语句（模块语句简称模块）。读者也可按其分类来记忆和使用这些模块。这里所谓的 GPSSWorld 模块是真正地用来模拟每一个典型的系统过程的，GPSSWorld 的标准输出中将对每一个这样的模块进行详细地统计。除了产生活动实体的模块（GENERATE 模块）外，所有模块的执行需要活动实体去触发。而 GPSSWorld 语言还有另外一部分语句，称为命令语句，比如控制语句、定义语句等，它们仅起控制仿真运行或定义仿真元素的作用，它们的执行是不需要活动实体去触发的。

3.3 GPSSWorld 程序的基本结构、语句组成及自定义符号

一个 GPSSWorld 的程序/模型应该是一个 GPSSWorld 语句序列，其中 GPSSWorld 语句又分为命令语句和模块语句。

3.3.1 GPSSWorld 程序的基本结构

一个 GPSSWorld 程序通常包含 3 个部分：说明部分、模块部分和控制部分。下面以例 3.1 为例来说明。

```
LINE    EQU         1       ;定义一个符号 LINE,它的编号是 1          说明部分
        GENERATE    16,9    ;顾客以每隔 16±9 分钟的时间间隔到来
        QUEUE       LINE    ;顾客排入名为 LINE 的队中
        SEIZE       JOE     ;若 JOE 闲,则占用 JOE 理发,否则停在队中
        DEPART      LINE    ;离队                                    模块部分
        ADVANCE     15,7    ;理发花了 15±7 分钟
        RELEASE     JOE     ;理发完毕释放理发师 JOE 离开
        TERMINATE   1       ;顾客离开系统
        START       20      ;共仿真 20 个顾客                        控制部分
```

第一部分：说明部分是由一些定义语句组成的，这一部分不可执行，它定义符号、函数、存储器、表格、初始化逻辑开关、保存值等，说明部分不是每个 GPSSWorld 程序都有，如不需要，说明部分便可以省略。

第二部分：模块部分是不可缺少的。它由可执行的模块语句组成，用于描述活动实体的行为轨迹/进程，是整个程序的关键。GPSSWorld 中大约有 50 个模块。它们是读者以后学习的重点。

第三部分：控制部分由一些命令语句组成，它也可以省略，可以在运行的时候通过交互式菜单命令来输入，方法是：仿真运行后，选择 Command|START 命令，然后在弹出的对话框中输入数字（控制模拟的次数），如图 3.12 所示，这样的方式和程序中写入 START 20 的效果是一样的。

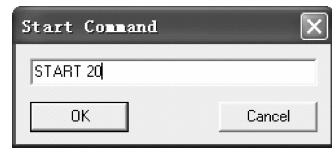

图 3.12　交互式的输入 START 语句

3.3.2　GPSSWorld 语句的组成

GPSSWorld 语句可以分为模块语句和命令语句，其中命令语句又可以分为定义语句和控制语句。GPSSWorld 模块语句与常见的高级语言有很大的不同，它的每个语句都表示一个过程，相当于高级语言中的一个子程序，且每个语句都有比较严格的格式要求。模块语句可以产生、消亡活动实体和改变活动实体的属性/状态，而命令语句不可以。命令语句用来定义除了活动实体以外的一些变量，函数存储器、逻辑开关等，以及控制仿真开始、仿真终止/结束等（通常位于说明部分和控制部分）。

GPSSWorld 语句必须位于同一行中，且最多包含 250 个字符。由 5 个数据域组成，每一个数据域之间用空格进行分割。

行号（可选）| 语句标签（可选）|操作码（必须的）|操作数（可选）|注释（可选）

```
⋮          ⋮          ⋮          ⋮          ⋮
5          AGN        ADVANCE    8,5        ;理发时间
⋮          ⋮          ⋮          ⋮          ⋮
```

（1）行号：这是一个可选项，在 GPSSWorld 版本中是不需要的，之所以保留主要是考虑到和 PC 版本的兼容性。如果要使用，必须从第一列开始输入。然而，它们在编译时是被忽略的，当模型中有错误时，错误信息提供的是语句在模型文件中的行数，而不是用户提供的行号。

（2）语句标签（地址码）：在 GPSSWorld 语言中语句标签的作用相当于高级语言中的语句号，其作用是使得程序在运行中可以根据语句标签来实现语句间的跳转，因此也并不是每个语句都需要的。语句标签必须遵循 GPSSWorld 语言中对于符号的命名规则，但是不可以用 SNA 码或系统关键字来命名。

（3）操作码（命令动词）：即模块语句或命令语句的名称，相当于高级语言中的关键字，它反映了语句的基本功能和所模拟的过程，由于这些操作码都是用英文全文书写的，因此非常地便于理解和记忆。GPSSWorld 中大约包含了 50 个模块（BLOCK STATEMENT）以及 25 条命令（COMMAND），以后将会在各章节中分别介绍这些模块和语句。

（4）操作数：很多的操作码在运行时要求有操作数，每个操作码对于操作数的要求都不一样，其操作数的意义也是不一样的，在 GPSSWorld 中对于操作数是只认位置不认数。

（5）注释：对于语句的说明，相当于高级语言中的注释语句，通常是为了增强程序的可读性。有两种注释的方式：①一整条语句都是注释：可以在语句的开始处用“;”或“*”注明；②在 GPSSWorld 语句的后面注释：可以在数据域的后面，注释开始处用“;”注明。

3.3.3 GPSSWorld 中的用户自定义符号

在 GPSSWorld 中允许用户自定义符号，允许给一个实体起一个指定的名字，但是这个名字必须遵守 GPSSWorld 中的命名规则。当需要使用标识的时候也可以使用 EQU 命令语句来定义。当然，也可以对自定义符号不加以说明。因为，当 GPSSWorld 碰到一个新的标识时系统会自动配给它一个唯一的编号，它是一个整数，通常都是从 10 000 开始计数。但是，如果希望自定义符号指向特殊的编号，那就必须得用 EQU 语句来说明了。例如，在例 3.1 中希望 JOE 指的是系统中的 1 号设施，这时就必须要用这样的语句来申明：JOE EQU 1。

可以给不同类型的资源实体起相同的名字，如同时给一个存储器和一个设施命名为 motor，但同一类型的实体则不可以同名。也可以为模块语句、用户自定义变量或 GPSSWorld 中的实体指定一个标识名字。GPSSWorld 的命名规则如下：必须以字母开头，由字母、数字和下划线“_”组成，且最多只能包含 200 个字符。命名时不可以使用系统关键字、SNA 码、模块语句或命令语句。如果无法记住系统中的关键字，可以有一个简便的方法避免同名：在自定义名字里使用“_”，或在名字的开头至少使用 3 个或以上的字母，然后再使用数字，从而和系统关键字或 SNA 码区分开来。

3.4 与活动实体有关的模块

活动实体是驱动 GPSSWorld 程序运行的原动力，是 GPSSWorld 程序的灵魂。从活动实体的产生到其消亡，活动实体在系统中的每一个行为/运动，系统的状态就会发生变化，建立 GPSSWorld 模型主要就是要用 GPSSWorld 的模块描述活动实体从到达到离开系统的行为过程。因此，活动实体在系统中行为/活动轨迹是编制 GPSSWorld 程序的脉络。

活动实体在仿真中将从 GENERATE 模块(产生活动实体的模块)开始，从一个模块语句移动到另一个模块语句，如果一个模块拒绝了活动实体的进入，则活动实体暂时延时等待，直到能进入为止。这时另一活动实体又开始从一个模块移动到另一个模块，从而一直到整个仿真结束。此外，从宏观上来看，GPSSWorld 建立的仿真系统对于活动实体来说是一个可以并行的系统。例如，在一个理发店中很可能会发生这样的情况：一个顾客可能正在洗头，另一个顾客可能正在理发，还有一个顾客已经理发完成正在付款等，每个活动实体分别沿着自己的行为轨迹向前运动，而不受其他顾客的干扰(除非发生对资源的竞争使用)。

活动实体在仿真系统中都有一个唯一的编号，这个编号是系统自动产生的，从 1 开始顺序编号。但如果执行了 CLEAR 命令以后，系统中活动实体将重新从 1 开始编号。

活动实体有一些很重要的属性，这些属性有些时候会影响活动实体的行动，下面介绍最常用的 6 个活动实体属性。

(1) 活动实体的参数：这些参数是依附于活动实体的，随着活动实体的移动而移动，活动实体生命周期结束了它们也就不存在了。我们可以将活动实体的一些属性、特性存入其所携带的参数中，如理发店系统中的顾客作为一个活动实体，其性别、年龄等数据可以作为其参数。活动实体的每个参数都有唯一的编号，从 1 号开始编号，也可以给参数起一个合法