

Справочное руководство по GPSS WORLD

Оглавление

Глава 1 - Введение	12
1.1 О главном	13
1.2 Концепция GPSS World	16
1.3 Архитектура	19
1.3.1 Многозадачность	19
1.3.2 Виртуальная память	19
1.3.3 Интерактивность.....	19
1.3.4 Визуализация	20
1.4 Язык моделирования	21
1.4.1 Что нового	22
1.4.2 Элементы языка	23
1.4.3 Встроенный язык программирования	24
1.4.4 Библиотеки процедур.....	25
1.4.5 Транслятор высокой производительности	27
1.4.6 Блок создания диалога	27
1.4.7 Автоматические генераторы экспериментов	27
1.5 Совместимость	27
1.5.1 Отличия от GPSS/PC	28
1.5.2 Модифицирование старых GPSS\PS программ	30
1.5.3 Строгое увеличение данных	31
Глава 2 - Управление GPSS World.....	33
2.1 Установка	33
2.1.1 Обзор.....	33
2.1.2 Сохранение устаревших объектов	33
2.1.3 Деинсталляция прежней GPSS World.....	34
2.1.4 Требования к установке	34
2.1.5 Папки/Директории.....	35
2.1.6 Процедура установки	36
2.1.7 Активация программного обеспечения.....	36
2.2 Среда GPSS World	37
2.2.1 Главное окно	37
2.2.2 Быстрый обзор	38
2.2.3 Установки	42
2.3 Контролирование сеанса.....	47
2.3.1 Использование главного окна	47
2.3.2 Построение моделей.....	49
2.3.3 Искусственные пределы	56
Глава 3 – Выражения модели	58
3.1 Использование выражений модели	58
3.1.1 Компиляция первичной модели (Initial Model Translation)	58
3.1.2 Интерактивное выражение	58
3.1.3 Последовательность блоков	59
3.1.4 Последовательность команд.....	59
3.1.5 Последовательность процедур	59
3.1.6 Сохранение объектов	60
3.2 Выражения GPSS.....	60
3.2.1 Команды GPSS.....	60
3.2.2 Выражения блока GPSS	61
3.3 Поля	62
3.3.1 Номера строк.....	62
3.3.2 Ярлыки.....	63

3.3.3 Действия (команды)	63
3.3.4 Операнды.....	63
3.3.5 Комментарии.....	63
3.4 Выражения	64
3.4.1 Выражения в выражениях GPSS	64
3.4.2 Система числовых атрибутов (SNA)	64
3.4.3 Выражения в режиме совместимости GPSS/PC	68
3.5 Имена	69
3.5.1 Ярлыки.....	69
3.5.2 Пользовательские переменные	70
3.5.3 Область имён.....	70
3.6 Числа.....	71
3.7 Использование строк.....	71
Глава 4 – Объекты GPSS.....	72
4.1 Объекты заявок	72
4.2 Блок объектов	74
4.3 Объект «Устройство».....	75
4.4 Функции	76
4.5 Объект «Логический переключатель» (Logicswitch)	77
4.6 Объект «Матрица».....	77
4.7 Объект «Очередь»	78
4.8 Объекты памяти.....	78
4.9 Объекты хранения	79
4.10 Объект «Таблица»	80
4.11 Пользовательские потоки	80
4.12 Переменные.....	80
4.13 Группы чисел	81
4.14 Группа заявок.....	81
4.15 Генераторы случайных чисел.....	81
4.16 Потоки данных.....	82
4.16.1 Операции потока данных.....	82
4.16.2 Использование потоков данных.....	84
4.16.3 Проверка на ошибки.....	84
4.17 Непрерывное моделирование.....	85
4.17.1 Как начать интегрирование	85
4.17.2 Базовые концепции.....	86
4.17.3 Переделы	86
4.17.4 Уравнения более высокого порядка	87
4.17.5 Непрерывное моделирование.....	87
4.17.6 Фазы.....	87
4.17.7 Ошибки интегрирования.....	88
4.17.8 Связанные команды.....	88
Глава 5 – Окна GPSS World	89
Глава 6 – Команды GPSS	102
BVARIABLE	103
CLEAR	104
CONDUCT	105
CONTINUE.....	105
EQU	106
EXIT	107
FUNCTION.....	107
FVARIABLE.....	111

HALT	111
INCLUDE.....	112
INITIAL	113
INTEGRATE.....	114
MATRIX	116
QTABLE	117
REPORT.....	118
RESET.....	119
RMULT	120
SHOW	121
START	121
STEP.....	122
STOP	123
STORAGE.....	124
TABLE	125
VARIABLE	126
Глава 7 – Блок выражений.....	128
ADOPT.....	130
ADVANCE.....	130
ALTER	131
ASSEMBLE	134
BUFFER	136
CLOSE	136
DEPART (ОТПРАВКА)	138
DISPLACE (ПЕРЕМЕЩЕНИЕ).....	139
ENTER (ПОСТУПЛЕНИЕ).....	141
EXAMINE (ПРОВЕРКА)	142
EXAMINE (ОБРАБОТКА).....	144
FAVAIL (ПРОВЕРКА ДОСТУПНОСТИ УСТРОЙСТВА)	144
FUNAVAIL (УСТАНОВЛЕНИЕ УСТРОЙСТВА В НЕДОСТУПНОЕ СОСТОЯНИЕ).145	
GATE (ШЛЮЗ).....	149
GATHER (НАКОПЛЕНИЕ).....	151
GENERATE (ГЕНЕРАЦИЯ).....	152
INDEX (ИНДЕКСИРОВАНИЕ).....	154
INTEGRATION (ИНТЕГРАЦИЯ).....	155
JOIN (ОБЪЕДИНЕНИЕ)	156
LEAVE (УХОД)	157
LINK (СОЕДИНЕНИЕ).....	158
LOGIC (ЛОГИКА).....	160
LOOP (СВЯЗЬ).....	161
MARK (МАРКИРОВКА).....	162
MATCH (СОГЛАСОВАНИЕ)	162
MSAVEVALUE (МАТРИЦА)	164
OPEN (ИНИЦИАЛИЗАЦИЯ).....	165
PLUS (СЛОЖЕНИЕ)	166
PREEMPT (ПРЕРЫВАНИЕ)	166
PRIORITY (ПРИОРИТЕТЫ)	170
QUEUE (ОЧЕРЕДЬ)	170
READ (ЧТЕНИЕ).....	171
RELEASE (ОСВОБОЖДЕНИЕ).....	172
REMOVE (УДАЛЕНИЕ).....	174
RETURN (ОСВОБОЖДЕНИЕ)	177

SAVAIL (ПРОВЕРКА ДОСТУПНОСТИ ПАМЯТИ)	178
SAVEVALUE (СОХРАНЕНИЕ)	179
SCAN (СКАНИРОВАНИЕ)	180
SEEK (ПОИСК)	182
SEIZE (ЗАХВАТ)	183
SELECT (ВЫБОР)	184
SPLIT (ДРОБЛЕНИЕ)	187
SUNAVAIL (ПРОВЕРКА НЕДОСТУПНОСТИ ПАМЯТИ)	189
TABULATE (ТАБУЛЯЦИЯ)	190
TERMINATE (ЗАВЕРШЕНИЕ)	191
TEST (ТЕСТИРОВАНИЕ)	191
TRACE (ТРАССИРОВКА)	193
TRANSFER (ПЕРЕДАЧА)	194
UNLINK (РАЗЪЕДИНЕНИЕ)	197
UNTRACE	198
WRITE	199
Глава 8 - PLUS: Язык программирования моделирования	201
8.1 Определение процедур PLUS	201
8.2 Язык	202
8.2.1 Набор символов	202
8.2.2 Имена	203
8.2.3 Выражения	204
Глава 9 – Дополнительные разделы	207
9.1 Цепи заявок	207
9.2 Планировщик заявок	209
9.3 Синхронизация	211
Глава 10 – Советы по увеличению производительности	212
10.1 Выделение памяти	212
10.2 Определение точек перегруженности	212
10.3 Советы по работе	212
10.4 Советы по построению эксперимента	213
Глава 11 – Стандартные отчеты	215
11.1 Управление отчетами	215
11.1.1 Окна отчётов	215
11.1.2 Управление содержимым отчета	215
11.2 Пример отчета	215
11.2.1 Модель	215
11.2.2 Отчет	216
Глава 12 – Статистика GPSS World	224
12.1 Введение	224
12.2 ANOVA (дисперсионный анализ)	224
12.3 RESET	225
12.4. Пространственно-временные продукты	225
Глава 13 - Экспериментирование	226
13.1 Введение	226
13.1.1 Эксперименты отбора	226
13.1.2 Пользовательские эксперименты	226
13.1.3 Оптимизация экспериментов	227
13.2 Экспериментирование и анализ вариантности	227
13.2.1 Мотивация	227
13.2.2 Номенклатура	228
13.2.3 ANOVA	229

13.2.4 Выбор факторов.....	232
13.2.5 Потоки случайных чисел	233
13.3 Функции GPSS World.....	233
13.3.1 Пользовательские эксперименты.....	233
13.3.2 Эксперименты PLUS	234
13.3.3 Матрица результатов.....	235
13.3.4 Библиотечная процедура ANOVA	237
13.3.5 Таблица остатков	239
13.4 Автоматические генераторы экспериментов.....	239
13.4.1 Процедура запуска.....	239
13.4.2 Фильтрующие эксперименты	240
13.4.3 Оптимизация экспериментов и поверхностная характеристика.....	245
13.5 Краткие указания по выполнению	249
Глава 14 – Сообщения об ошибках.....	251
14.1 Введение	251
14.2 Сообщения и Объяснения.....	251

Описание продукта

GPSS World - высокоуровневый, специально оснащенный, многоцелевой комплект, созданный для профессионального моделирования.

Этот современный инструмент позволяет осуществлять, как дискретные, так и непрерывные компьютерные модели с чрезвычайно высоким уровнем интерактивности и визуализации. Используя GPSS World, можно предсказать влияние пользовательских решений на особо сложные существующие мировые системы.

О данном руководстве

Данное руководство содержит информацию об использовании программы GPSS World и является основным указанием по использованию языка GPSS. Часть первая руководства показывает все, что необходимо знать, чтобы использовать данный продукт во всей полноте.

Последняя часть руководства содержит описание каждого выражения GPSS и различную информацию, которая может быть полезна, по мере того, как вы приобретаете опыт в овладении GPSS World. Так же имеются приложения с примерами моделей, где вы можете найти примеры, имеющие сходство с вашими проблемами моделирования.

Как начать

Если вы уже знакомы с GPSS и не хотите проходить подробно все этапы описания GPSS, то после установки GPSS World согласно инструкции в главе №2, вы можете приступить к руководству, называемому GPSS Tutorial.

Затем по мере возникновения вопросов обратитесь к главе 6,7 и 8 данного руководства.

Online-помощник поможет вам избежать подробного изучения выражений. Войти в этот помощник можно нажатием F1 или кнопки «помощь». Так же можно получить помощь по текущей проблеме, наведя на нее курсор. Чтобы максимально использовать возможности GPSS World необходимо подробно проработать данное руководство и затем руководство GPSS World Tutorial.

Части данного руководства

Данное руководство начинается с общей информации.

Далее следует более подробное описание. Главы 1 и 2 содержат представление о GPSS World, краткое описание и некоторые основные положения.

Главы 3 и 4 описывают наиболее важные аспекты языка GPSS.

Глава 5 содержит подробное описание окон, графического интерфейса.

Главы 6, 7, 8 являются частью справочника, они содержат подробное описание каждой команды, Block Statement и PLUS Facilities.

Глава 9 содержит множество подробностей о программировании в GPSS. В целом только опытные программисты используют материал из этой главы.

Глава 10 содержит информацию о том, как более полно использовать возможности моделирования. Глава содержит советы и другую информацию, которая может быть полезной для вас

Глава 11 описывает возможности и содержит отчеты, создаваемые GPSS World.

Глава 12 обсуждает статистику, автоматически создаваемую GPSS World.

Глава 13 содержит описание экспериментов в GPSS World. В дополнение к мультимедийным процедурам ANOVA она содержит описание высокопроизводительных генераторов оптимизации.

Глава 14 описывает ошибки, разъяснения и корректирующие действия.

И, наконец, приложение содержит грамматический справочник и краткий словарь. Читатели найдут в грамматическом справочнике большинство элементов GPSS World-диалекта. Формы операндов даны под соответствующим выражением в главах 6,7,8 данного описания.

Как данное руководство описывает действия

Понятия со значениями специфическими для языка GPSS World написаны заглавными буквами. Например, Цепочка Текущих Событий – специфическая конструкция с особым

значением. Когда слова употребляются с общим значением и не имеют отношения к конкретной области, то они не выделяются заглавными буквами.

Действия, которые вас просят выполнить немедленно, представлены глаголами в верхнем регистре. Например, когда вы видите надписи **PRESS**, **CHOOSE**, **CLICK**, **DOUBLE CLICK**, **SELECT**, **TYPE** вы сами выполняете соответствующие действия. Если действия выполняются, но вас не просят выполнить их немедленно, глаголы будут написаны в нижнем регистре.

Слова, отображающие предметы или понятия характерные для GPSS World часто показываются шрифтом *Italic*.

Нажатие клавиш, которое необходимо выполнить, обозначается специальным шрифтом, который содержит имя кнопки на клавиатуре.

Например, нажмите «Ввод», означает, что вы должны нажать клавишу с соответствующим именем. Когда надо использовать более одной клавиши, элементы клавиатуры объединены знаком «+».

Действия с мышью понадобятся, когда появляющиеся фразы **CLICK** и **DOUBLE CLICK** находятся в верхнем регистре. Это означает, что вам нужно использовать кнопку мыши, чтобы осуществить действие. Правая кнопка мыши вступает в действие для более подробных уточнений. К тому же вы можете не использовать OS\2.

Пункты меню указываются жирным шрифтом со стилем Times New Roman. Когда необходимо сделать многоступенчатый выбор, они показываются в определенном порядке: menu bar item / pull down menu item / cascade menu item. Например:

CHOOSE FILE / OPEN

Приведенное выше означает, что вы можете использовать как мышь, так и клавиатуру для выбора пункта меню «file». Затем нужно выбрать в раскрывшемся меню, нужный пункт, причем в данном примере не проиллюстрированы каскады пунктов меню. В данном руководстве «CHOOSE» используется для пунктов меню. «SELECT» для прочих выборов. Диалоговые окна так же показываются в стиле Times New Roman. Например:

SELECT OK

Это значит, что можно использовать как мышь, так и клавиатуру, чтобы выбрать кнопку «OK» в диалоговом окне.

Когда вы выбрали кнопку «OK» вы можете завершить это действие нажатием «Ввод» на клавиатуре. Существуют исключения, например, часто **Command / Custom** обеспечивают мультилинейное использование командных списков и процедур языка PLUS. Клавиша ввода используется в таких случаях только для прерывания текстовой строки, вместо

SELECT OK.

Символы, которые вы вводите в текстовую строку, отображаются жирным шрифтом стиля Courier New, например,

TYPE GENERATE 100

означает, что вам нужно напечатать буквы «Generate 100», убедившись, что текстовое окно активно и что курсор установлен в нужном окне.

Наконец, предметы операционной системы выделены подчеркиванием, чтобы отделить их от собственно GPSS World. Например,

DOUBLE CLICK on The GPSS World Icon

означает, что вам нужно выбрать ярлык на рабочем столе или в другой рабочей папке, который представляет объект. Далее вам нужно навести указатель мыши на объект и щелкнуть дважды левой кнопкой мыши.

Операции меню

Вы можете использовать как мышь, так и клавиатуру, чтобы выбрать пункты меню. В многоуровневом меню курсор выбора появляется, чтобы показать вам какой пункт меню вам выбрать. Курсор выбора обозначает объект пунктирной линией или как нажатую кнопку меню. Кнопка Alt, когда она нажата, будет выбирать первый пункт из главного

меню, выделяя кнопку меню, как нажатую. Указатель мыши, может быть использован для перемещения пунктов меню. Кнопка ввода делает окончательный выбор.

Использование мыши

Существует два пути использования мыши для выбора пункта меню. Вы можете установить указатель мыши над выбранным пунктом меню, затем нажать и отпустить левую кнопку мыши. Вы можете сделать это на панели меню, с помощью контекстного или каскадного меню. Вы также можете нажать левую кнопку мыши и, не отпуская ее сразу, перетащить пункт меню из одного в другой, затем отпустите кнопку. Вы можете использовать эти комбинированные операции для ускорения работы, хотя на клавиатуре это можно сделать так же быстро.

Использование клавиатуры

Клавиатура сама по себе может быть использована, чтобы сделать выбор в меню. Если любые действия в меню находятся в процессе, нажмите значок «^». Затем нажмите **a** для установления курсора выбора на первый пункт панели меню.

Теперь вы можете перемещать курсор выбора с помощью стрелок клавиатуры, чтобы сделать окончательный выбор в пунктах меню. Так же вы можете просто нажать кнопки запоминания. Эти клавиши обозначены подчеркнутыми буквами, появляющимися непосредственно в пунктах меню. После входа в клавиатурный выбор путем нажатия ALT вы можете нажать кнопки запоминания, чтобы выбрать интересующие вас пункты меню.

И наконец, некоторые пункты имеют короткие сочетания клавиш обозначенных в меню справа от пункта, вы можете тот час же выбрать пункт меню, соответствующий указанной комбинации. Конечно, чтобы вы были в состоянии это сделать окно должно быть активно.

Текстовые окна

Когда текстовое окно GPSS World активно, клавиши обычно вызывают появление букв в окне, кнопка Ins может использоваться для изменения режима ввода. Режим ввода обозначен мигающим текстовым курсором, который так же указывает на точку ввода в окне. Так же клавиатура или мышь могут перемещать текстовый курсор, нажимая левую кнопку мыши, вы передвигаете текстовый курсор в позицию под указатель мыши. Стрелки клавиатуры и клавиши [Home], [End], [PgUp], [PgDn] все перемещают текстовый курсор и позволяют ввести и стереть текст.

Новая строка начинается с нажатия клавиши ввода. Выбранный текст показан в инверсии и может участвовать в специальных операциях, вы можете выбрать текст с помощью протаскивания указателя, при зажатой левой кнопке мыши, щелчком левой кнопки мыши с зажатой клавишей «j» или двойным щелчком на выбранном слове. Вы можете уничтожить текст один символ за одно нажатие кнопки Del или Backspace. Нажимая Del, уничтожается буква справа от курсора, Backspace – слева. Уничтожение может производиться с помощью меню **Edit / Cut** или нажатием Del. Переход на один уровень редактирования назад может быть осуществлен с помощью команды меню **Edit / Undo**.

Графические окна

Графические окна поддерживают действия меню, базированные на выборе одного или более объектов в окне, например, сопряженное сочетание блоков может быть скопировано с формы блока окна ввода в буфер обмена.

В целом объект может быть выбран кликом на свою иконку, удалить выбор так же возможно. В дополнение некоторые окна поддерживают расширенное выделение. В начале выбирается объект, затем при нажатой клавише SHIFT осуществляется скроллинг до конечного объекта, выделение производится при клике на этот объект. Наконец, если вы удерживаете CTRL во время клика на иконку, то она либо выделяется, либо если выделение уже было установлено на объект до этого, то оно снимается.

Диалоговые окна

Хотя диалоговые окна GPSS World все различны, в целом они делают одни и те же вещи. Мышь используется для того, чтобы сделать активными поля ввода и сделать выбор в

каждом окне. Обычно, клавиша ввода выбирает OK, клавиша ESC выбирает CANCEL. Клавиша F1 вызывает помощь.

Получение помощи

Online-помощь доступна, когда вы используете помощь по выделенному объекту в диалоговом окне:

CHOOSE Help

или нажатием клавиши F1. Однажды активировав систему помощи online вы можете использовать все множество выбранных техник для информирования.

Если вы выбрали слово в текстовом окне окно помощи откроется прямо поверх в справочном руководстве.

Универсальные клавиши

Несколько клавиш и комбинаций клавиш эффективны, независимо от того, какое GPSS World окно активно, они обычно используются, чтобы контролировать текущее моделирование, быстро применять команды и активировать моделирующий сервер.

Горячие клавиши

Горячие клавиши инициируют интерактивные команды с единственной комбинации клавишей. Они описаны в главе 2.

Функциональные клавиши

Функциональные клавиши могут быть даны, как оперативные рекомендации. Вы можете сделать это в установках программы, набором новых предписаний рядом с соответствующей меткой функциональной клавиши. После этого, когда вы нажимаете соответствующую клавишу, подписанная ранее команда интерпретируется и посылается на сервер. Вы можете использовать в комплексе Command List и/или PLUS-процедуру для переопределения предписаний функциональным клавишам с помощью выражения INCLUDE.

Целевая аудитория

GPSS World - это полноценная моделирующая система с дизайном для профессионалов-программистов. Предписанное количество подготовки требуется для того, чтобы ваши ожидания использовать ее в полной мере оправдались. Это руководство для пользователей, которые уже знакомы с базовыми элементами GPSS-языка. В дополнение вы должны знать, как оперировать вашей персональной компьютерной системой. Ожидается, что вы знаете о файловой системе, дисковых драйверах, директориях, объектах рабочего стола, дискетах, и владеете мышкой.

Так же предполагается, что вы знаете, как работать с меню и диалоговыми окнами при помощи мышки или клавиатуры. Эта информация доступна в документации вашей операционной системы. Если вы GPSS/PC-пользователь, обязательно прочтите раздел называемый GPSS World Features в главе 1. Там приведено детальное описание преимуществ GPSS World перед старой GPSS/PC-системой. Более подробно по каждому пункту в следующих главах. Обязательно убедитесь, нет ли необходимости узнать больше по каждому отдельному пункту. Помощь в виде Online Help и online Reference Manual также доступна для вас. Если вы новичок в GPSS, Системе Моделирования Общего Назначения, вы должны сделать некоторую подготовительную работу, прежде чем вы продолжите знакомство с этим руководством.

Во-первых, вам следует прочитать одну из замечательных книг GPSS языка, затем вам нужно тщательно проработать сопутствующее руководство пользователя GPSS World Tutorial.

И наконец, вам следует прочитать первые четыре главы этого руководства и использовать оставшиеся главы как справочный материал. Этот курс обучения поместит всю интерактивную мощь GPSS World в ваш багаж знаний. Лучше всего использовать это руководство до тех пор, пока вы не приобретет практический опыт. Большинство GPSS программистов очень продуктивно работают без использования всех блоков доступных в этом языке. Однако, вы можете быть более эффективными, моделируя при помощи

методологии GPSS языка, и достигнуть больших вершин. Вы познакомитесь, как с online справочником, так и с руководством пользователя. Они оба легко достижимы и обеспечивают ответы на частные вопросы.

Благодарности

Minuteman Software с благодарностью отмечает вклад докторов Аверилла Лоу и Стефана Винсента в развитие многих различных направлений и алгоритмов используемых в GPSS World, и построение возможностей этой системы.

Глава 1 - Введение

Для системы GPSS World характерно предсказывание поведения сложных существующих моделей реальной предметной области. Многие дорогостоящие эксперименты прошлого провалились из-за того, что конечный их результат нельзя было охарактеризовать как точный. Исходя из максимального объема стоимости операции, существенно было сделать вывод о поведении систем, конструкции которых узнавались не сразу. Хотя чисто математические модели оценены слишком высоко и могут быть использованы там, где необходимо. Сложность большинства существующих мировых систем, требуют использования компьютерного моделирования, чтобы получить необходимые результаты. Вот где вступают в силу системы GPSS World.

GPSS World базируется на особом языке компьютерного моделирования GPSS, который стоит в основе Системы Моделирования Общего Назначения. Этот язык развивался изначально при участии Джеффри Гордона в IBM в 60-е годы и получил важный коммерческое развитие, как инструмент дискретно-событийного моделирования. GPSS World – это прямой последователь GPSS/PC, ранней реализации языка GPSS для персональных компьютеров.

Со времени релиза в 1984 году GPSS/PC, пользователи сэкономили тысячи миллионов долларов. Теперь реализации GPSS под операционную систему Windows простирает свои возможности и на Интернет. GPSS World создана, чтобы отвечать на вопросы быстро и надежно, с наименьшими усилиями достигать высочайшей надежности и результатов. Существование этих объективных, динамически визуализированных моделей высоко стилизовано и их эффективность статистически доказана.

Этот подход означает, что анимации свободны в использовании и не требуют дополнительных усилий для их создания, но они не так реалистичны как фото. Сильная сторона GPSS World – это не фотореализм. Третье поколение анимационных систем обеспечивает растровую анимацию, базирующуюся на моделях GPSS World.

Инвариантность очень важна по трем причинам. Первое, опасно полагаться на "Black-Box" моделирование (моделирование по схеме «Черный Ящик») чьи внутренние механизмы не могут находиться в полном обзоре, не только потому вы не можете быть уверенным в успешном результате в этой ситуации, но и потому что трудно уверенно сказать, что ваши ожидания совпадут с результатом. Во-вторых, успешное моделирование ценно и имеет долгую жизнь. Возможно, что новые пользователи будут все больше и больше знакомиться с внутренним устройством моделей, но это почти нереальная задача пока обеспечение еще находится на невысоком уровне. Третье, одно из наиболее эффективных, но часто забываемых преимуществ компьютерного моделирования это новый взгляд на поведение системы, который достигается, когда опытный программист профессионал, может видеть внутреннюю динамику в ключевые моменты времени. GPSS World был создан для того, чтобы удовлетворить всем этим трем пунктам. Его визуалистическая природа позволяет внутренним механизмам моделей быть рассмотренными в любой момент времени и его интерактивность позволяет вам делать любые манипуляции в моделировании. Его встроенная возможность работы с данными, позволяет подсчитывать доверительные интервалы и легко анализировать дисперсию.

И теперь GPSS World может создавать даже сложные фильтрующие и оптимизирующие эксперименты автоматически, при небольших усилиях со стороны пользователя.

Большинство систем могут быть смоделированы несколькими путями с использованием GPSS World. Обычно только малая часть возможностей системы используется для достижения цели. Однако серьезная цель требует знакомства со всеми теми возможностями, которые может предложить GPSS World. Это руководство является первейшим источником необходимой информации.

Эта глава состоит из пяти разделов. Первый – это короткий обзор наиболее важных аспектов GPSS World, второй – открывает вам несколько базовых концепций, которые

необходимы для перехода к дальнейшим главам, третий – обсуждает продвинутую архитектуру GPSS World, и что вы должны знать, чтобы использовать все его преимущества, четвертый раздел рассказывает о языке моделирования GPSS World и пятый раздел для GPSS/PS пользователей, которые хотят взять все достижения из GPSS World, этот последний раздел доступен так же в коммерческой версии GPSS World. Установка и операционные процедуры обсуждаются в главе 2.

1.1 О главном

GPSS World объектно-ориентирован, его элементы содержат Модели Объектов, которые используются для создания Объектов Моделирования. Модели Объектов в свою очередь используются, чтобы проводить моделирование и создавать Объекты Отчетов. Наконец, текстовые объекты могут быть использованы во включенных файлах для поддержки совместного использования кода и пользовательских ресурсов таких как библиотека кодов, и они так же часто используются как файлы которые могут быть прочитаны или написаны программистом. Моделирующие проекты требуют многоступенчатого подхода, они обычно включают построение моделей и создание коллекции данных, тестирование и контроль, моделирование, экспериментирование и анализ результатов. GPSS World имеет огромное количество мощностей, адресованных каждой из этих ступеней. В GPSS World вы можете создавать и модифицировать модели, используя полноэкранный текстовый редактор, если вы предпочитаете, вы можете включить выражения GPSS Block, используя специальный блок ведения диалогов, где вы можете просто заполнить формы. Блок результирующих выражений расположен в точке ввода вашего моделирующего объекта, вы затем создаете моделирующий объект путем выбора команды создания модели в главном окне меню **Command / Create Simulation**. После этого, в вашем распоряжении мощный набор команд для вашего контроля и руководства процессом моделирования. Вы можете вводить команды интерактивно или вы можете включить их в исходную модель обычным способом. Во время тестирования и контроля огромное количество типов окон достижимы для обозрения, чтобы видеть процесс создания моделей. Горячие клавиши и «Point» и «Shoot breakpoint» делают проверку и отладку достаточно легкой. После того как модель протестирована, вы можете использовать автоматический генератор экспериментов или создать свой собственный эксперимент. GPSS World поможет вам на каждой ступени этого пути. GPSS World был создан, чтобы усилить вычислительные мощности вашего компьютера. Использование виртуальной памяти позволяет вашим моделям расширяться до миллиардов байтов. Преимущество многозадачности и многопоточности обработки позволяет GPSS World быть высокопроизводительным и выполнять несколько действий одновременно. Это так же означает, что моделирование может использовать симметричную мультипроцессорную архитектуру, для повышения мощности.

Структура моделирующего языка была упрощена в GPSS World. Модель теперь обозначается как последовательность выражений модели. Каждая может быть определена как GPSS Statement или PLUS Procedure или PLUS Experiment.

Объект Модели используется для создания связанных объектов моделирования, которые затем могут быть активированы. Каждое выражение модели может быть послано в существующий моделирующий объект. Каждое выражение вызывается из «Interactive Statements». Эти концепции обсуждаются более подробно в следующих разделах.

Новое в GPSS World это блок создания диалогов. Это огромное достижение в обучающем элементе. Используйте

Edit / Insert Block

для открытия меню команд «Block Menu Window», где можно выбрать кнопки для любого элемента GPSS World. Навести мышь на каждый блок, все, что необходимо сделать, чтобы увидеть описание этого блока. Когда вы нажимаете кнопку, открывается диалоговое окно, со всеми элементами, которые вам необходимы для создания GPSS блока, остается только

заполнить форму. Кнопка помощи приведет вас в нужное место справочника с описанием блока. Когда вы кликните ОК, результирующий блок появляется в объекте модели.

Не беспокойтесь, любая ошибка синтаксиса исправляется немедленно. Для целей форматирования, встроенный блок выражений GPSS содержит таблицы. Вы можете настраивать параметры отступа в настройках модели объекта. В каждом случае Block-выражение и выражение языка PLUS является источником кода, который вы можете править, до того как будет создан объект моделирования.

Язык моделирования внутри GPSS World был расширен посредством языка PLUS, Языка Программирования Моделирования. Этот простой, но мощный язык программирования не содержит ограничений, которые существовали в более старых версиях GPSS интерпретаторов. Данные в этой среде разработки непечатаемые, с конверсией, происходящей автоматически в случае надобности. В дополнение мощные функции и библиотека с различными возможностями доступны в выражениях PLUS.

Процедурная библиотека поддерживает операции со строками, числовые вычисления и распределения вероятностей. Пользователь, написавший PLUS процедуру, может пользоваться ей различными путями.

Команда INCLUDE переносит в существующую библиотеку пользовательских процедур тест процедур PLUS для его последующего использования в моделировании.

Процедуры PLUS, определенные пользователем, могут быть использованы повсеместно в модели.

Выражения, определенные в PLUS могут включать в себя элементы SNA. PLUS процедуры могут использоваться в выражениях GPSS.

Фактически многие аргументы выражений GPSS могут находиться в форме выражений в круглых скобках. В версии 4 GPSS World язык PLUS был расширен до организации экспериментов. Это мощная особенность позволяет осуществлять программируемый контроль и даже может быть основана на результатах моделирования.

Однако, полностью автоматическая операция возможна в том числе и для изучения поверхностей отклика. Эксперименты PLUS, вызываемые командой CONDUCT могут быть использованы для контроля за ходом моделирования через параметры временного интервала. Эксперименты могут быть созданы с минимальным участием с вашей стороны. GPSS World теперь включает два мощных генератора экспериментов, которые могут быть доступны через меню Edit главного окна. Генератор контроля за экспериментом создаст подробный факторный эксперимент по вашим указаниям и вставит его в виде программного кода PLUS в вашу модель. Таким же образом генератор оптимизации экспериментов создаст сложную поверхность отклика для поиска максимума или минимума при получении результатов моделирования. Команда CONDUCT необходима, чтобы инициализировать эксперименты через запись их в Ключ Функции. Чтобы проанализировать результаты работы Screening или Optimizing Experiment все, что вам необходимо это нажать кнопку.

GPSS World имеет современные возможности дискретного и непрерывного моделирования. Тесная связь между дискретным и непрерывным моделированием обусловлена легким переходом из одной фазы моделирования в другую соответственно.

Из непрерывной фазы может быть установлен триггер создания транзакции для дискретной фазы. И наоборот, блок INTEGRATION и команда INTEGRATE контролирует процесс в непрерывной фазе.

Несколько новых GPSS блоков были добавлены, чтобы обеспечить контроль интеграции, перепланирование транзакций, изменений в «Assembly Sets», изменений в PLUS блоках, определенных пользователем и в «Data Streams» (потоках данных). Типы данных теперь включают integer и real форматы данных, так же как и строковые. Каждый тип автоматически приводится к требуемой форме. Новый тип данных "UNSPECIFIED" теперь доступен для определения типов данных, которые не являются стандартными, так же как и потерянные данные во время эксперимента.

Каждая матричная структура была улучшена. Сейчас они могут поддерживать до шести размерений. Новая мультимедийная процедурная библиотека ANOVA анализируют данные из результирующих матриц, которые не есть что-то большее, нежели матрицы GPSS, где результаты эксперимента были структурированы в традиционную форму. GPSS World легко оперировать. Полноэкранный текстовый редактор может быть использован в любом текстовом окне. Даже окна «Journal» и «Reports» могут быть переделаны и снабжены комментариями. Вы можете использовать пометки, как закладки в любом текстовом окне. Затем вы можете перемещаться от закладки к закладке в рамках большой модели путем тривиального нажатия клавиши. Когда вы моделируете, если обнаруживается какая-либо ошибка, она запоминается, поэтому вы можете исправить ее в любое время. Когда вы обращаетесь к последующей ошибке, курсор автоматически перемещается к месту в модели, где ошибка была замечена, чтобы помочь вам увидеть причину каждой проблемы. После завершения моделирования, несколько горячих клавиш могут быть использованы как интерактивные команды. И это не все, используя модель и установки моделирования, вы можете создавать свои собственные команды, присваивая их одной или более горячей клавише, и получать заданные действия простым нажатием клавиши.

Окна легко открывать. В целом, вам потребуется только меню по выделению. В процессе отладки, панель инструментов отладки позволяет прерывать управление для достижения совершенства в «Point» и «Shoot Fashion».

Процесс моделирования легко визуализировать. GPSS World позволяет создавать стилизованную анимацию в любых GPSS сущностях, путем нескольких нажатий кнопки мыши. Эти окна динамичны и показывают изменения состояний в процессе моделирования. Вы можете обозревать изменения элементов в любой GPSS сущности. В дополнение к данному просмотру вы, так же можете открывать специальные окна выражений и просматривать динамическое изменение значений любого параметра вашего выражения.

Для визуального представления переменных в вашей модели, можно открывать любое число PLOT-окон, чтобы показать графическую динамику. И наконец, «Table Window» позволяет изменять частоту распределения по вашему усмотрению.

В том числе, GPSS World поддерживает скриншоты. Эти продвинутые особенности созданы для профессионалов, которым необходимы детальные знания всех подсистем моделирования. Статические снимки экрана могут быть сделаны в любое время выполнения транзакции, взяты из потока будущих и текущих заявок или числовых или транзакционных групп.

GPSS World – высоко интерактивен. Все выражения модели могут использоваться в интерактивном режиме. Когда вы передаете команду, блоковое выражение, описание процедуры или описание эксперимента на существующие объекты моделирования, интерактивность используется для переопределения или манипуляции с состояниями конкретной модели. Этот метод контроля очень полезен для контроля за изменяющимися состояниями модели. По существу, вы можете делать почти все, исключая вставку Блоков в моделирование после того, как оно было создано.

Моделирование может взаимодействовать с внешней средой. Теперь вы можете использовать пять новых GPSS блоков и/или PLUS-процедур для управления потоками данных. Например, OPEN, CLOSE, READ, WRITE и SEEK. Последний, обеспечивает прямой доступ к базе данных. Потоки данных имеют много назначений. Вы можете использовать их для доступа к файлам данных, для создания результирующих файлов и составления отчетов, для обращения к внутренним данным напрямую. Теперь типовая библиотека PLUS, за исключением некоторых, может предоставить вам почти все функции. Это означает, что комплекс операций ввода-вывода может быть оснащен изнутри пользовательскими процедурами PLUS. GPSS World моделирование может даже связываться с другими продуктами напрямую. Библиотека процедур PLUS теперь

содержит процедуры **Call()**, **Call_Integer()**, **Call_String()** и **Call_Real()**, которые могут вызывать внешние функции в вашу систему из исполняемых файлов (EXE and DLL files). Эти возможности описаны в главе 8.

Анализ результатов теперь прост в GPSS World. Он имеет возможности захвата и печати графических окон. «Journal Window» записывает всяческую активность, связанную с объектом моделирования. Благодаря автоматической системе нумерации отчетов, гарантируется полная сохранность каждого стандартного отчета. Новая библиотека процедур ANOVA может выполнять полный многоцелевой анализ переменных и затем передавать результат в результирующую матрицу. Вся эта работа выполняется на основе таблиц ANOVA и доверительных интервалов. Теперь автоматический генератор экспериментов может создавать сложные проверочные и оптимизационные эксперименты, базирующиеся на диалоговых окнах, вставляемых вами. Базовый анализ этих экспериментов проводится автоматически, и сообщения об этом закладываются в «Journal Window».

Новый пакетный режим операций может быть использован для запуска моделирования в фоновом режиме.

Если вы устанавливаете файл модели или файл моделирования с помощью командной строки DOS введите команду BATCH и GPSS World запустится в минимизированном окне. Для начала откройте файл, а затем пропустите его через Create Command (если это объект модели) или CONTINUE Command (если это объект моделирования). В последнем случае вам остается только определить желаемый уровень контроля команд в модели. Вы можете использовать команду EXIT 1 (или библиотечную процедуру Exit(1)), чтобы закрыть текущую сессию, автоматически сохраняя все измененные и созданные объекты. EXIT обсуждается в главе 6.

GPSS World наследует некоторые характеристики от GPSS/PC. Например, он позволяет использовать символ «#» для обозначения умножения, наряду с символом «*» для не прямой адресации в GPSS. Вы можете поменять настройки так, чтобы какой-то из этих символов обозначал умножение, по вашему усмотрению.

Также выражения GPSS за исключением «Function Followers» должны располагаться в одной текстовой строке, максимальная длина которой не превышает 250 символов. Выражения PLUS не так ограничены. Они могут содержать любое количество строк.

1.2 Концепция GPSS World

GPSS World применяет четыре вида объектов. «Model», «Simulation» и «Report Objects form» являются тремя базовыми типами, которые используются во всех моделях GPSS World. Обычно модель строится путем создания выражений в объекте модели. Затем производится создание команды моделирования, посредством чего создается объект моделирования в соответствии с выражениями в модели. Моделирование осуществляется посредством передачи их с помощью команд GPSS или с помощью включения команд в объект модели. Обычно, когда моделирование завершается объект отчета создается автоматически. Полный контроль за процессом комплексного моделирования может осуществляться путем ручной вставки какой-либо структуры данных в эксперимент PLUS в модели объекта.

Команда CONDUCT может быть передана в объект моделирования, чтобы контролировать прогоны модели и выдавать результаты.

Текстовый объект включает четыре типа объектов GPSS World. Они именуются посредством выражения INCLUDE, чтобы помогать развитию широкомасштабного моделирования, так же используются для некоторых других целей, таких как создание библиотеки ресурсов кодов. В дополнение текстовые объекты часто используются для объектов моделирования, когда выполняются операции OPEN, CLOSE, READ, WRITE и SEEK.

Все объекты GPSS World, могут быть сохранены в любое время, в любом месте, и могут быть открыты в последующих сеансах работы. Они включают объекты моделирования, которые могут быть запущены в любой момент времени из сохраненной области, таким образом облегчается отладка и демонстрация. Объекты моделирования могут быть сохранены в середине моделирующего процесса, что позволяет вам закончить моделирование в другое время.

Все объекты GPSS World за исключением текстовых объектов содержат данные в дополнение к базовому ASCII-тексту. Поэтому вы можете использовать внешний текстовый редактор для создания текстовых объектов, но только для этого. Модель, имитация, отчет содержат закладки для упрощения навигации, список атрибутов значений вызывают настройки, которые могут быть изменены и переданы дочерним объектам. Когда создается объект модели, его создание начинается с копирования настроек модели, в которой он создан. То же самое происходит с объектами отчета, которые наследуют свои настройки от родительских объектов моделирования. В дополнение объекты модели запоминают все ошибки, которые происходили, когда вы пытались создать моделирующий объект из них. Пользуясь этим путем, вы можете осуществлять некоторую коррекцию и сохранить свое время в будущем.

Model Objects (Объект модели)

Объекты модели содержат преимущественно последовательности выражений модели и встроенных значений, сопряженных с настройками.

В дополнение, установка закладок и циклических списков результатов синтаксических ошибок есть также часть объекта модели. Когда моделирующий объект создается путем импорта выражения в модель, она наследует все настройки родительской модели. Команда меню **Command / Create Simulation** главного окна программы используется для создания новых имитационных объектов.

Model Statements (Выражения модели)

Модель это коллекция выражений содержащихся в объекте модели и его внутренних файлах. Выражение модели может быть также выражением GPSS или процедурой PLUS. Выражение GPSS в свою очередь может быть так же блоком выражений или команд. Выражения модели определяют объект моделирования, когда он в первый раз компилируется. Они могут быть переданы для преобразований в существующем моделировании.

Команды могут быть «Очередными» и «Безотлагательными». На самом деле все команды, за исключением HALT и SHOW – очередные. Каждый объект моделирования имеет специальную последовательность, называемую очередью команд, которая содержит все очередные команды, полученные для объекта моделирования. Команды, взятые из очереди, выполняются одна за другой.

Команда HALT – особый случай. Это безотлагательная команда, и по этой причине она не размещается в очереди команд моделирования.

Однако, когда HALT выполняется, все оставшиеся команды в очереди команд удаляются. Это гарантирует, что все действия моделирования прекращаются, когда передается команда HALT. Обычно, сброшенное моделирование возобновляется путем передачи ему команды CONTINUE.

Multiple Model Files (составные файлы модели)

Обычно модель представляет собой набор выражений в единичных объектах модели.

Некоторые выражения можно помещать в отдельные текстовые файлы, называемыми текстовыми объектами. Это иногда полезно для сочетания одного или более выражения или процедуры PLUS и/или PLUS-экспериментов, совмещенных в текстовом объекте для последовательного моделирования. Текстовые объекты, которые являются простыми текстовыми файлами, могут быть названы в выражении INCLUDE в объекте моделирования. Выполняя таким образом все выражения в текстовом объекте, вы можете вставлять их в места определенные командой INCLUDE, когда объект моделирования

создан. Текстовые объекты могут сами по себе содержать выражения INCLUDE, максимум вложений – пять уровней. Команда INCLUDE может быть использована интерактивно так же как внутри модели. Они могут даже быть загружены в функциональные клавиши, и таким образом вызывать комплекс действий одним нажатием клавиши. Это в подробностях обсуждается в главе 2.

Модель, построенная из выражений, в точности соответствует одному объекту моделирования и может содержать сколько угодно текстовых объектов, которые запоминаются в выражениях INCLUDE, впервые проявляющих себя при компиляции модели. Только объекты модели могут иметь настройки связанные непосредственно с ними, которые могут быть доступны через меню главного окна **Edit / Settings**.

Эти установки наследуются всеми созданными объектами моделирования из объекта модели. Для предотвращения возникновения сложноструктурированных установок, объекты модели не могут использоваться в выражениях INCLUDE, только в текстовых объектах.

Есть два типа нумерации файлов. Модель и включенные в нее файлы нумеруются внутренне, таким образом, создаются ссылки в окне статуса имитации. Транслятор после моделирования объекта производит нумерацию с нуля. К примеру, главная модель ассоциируется с числом ноль, но включенные файлы считаются с единицы. Эта нумерация только для распознавания ошибок. Разные системы нумерации уже применены к новым сделанным моделированиям и к файловым отчетам, чтобы держать имена файлов индивидуальными. Вы можете установить нумерацию имен файлов для потомка такие, как и для родителя. К примеру, чтобы поменять нумерацию имен файлов имитационных моделей, вам нужно изменить настройки директории объекта моделирования. Подобным образом изменяются настройки для отчета объекта потомка.

Настройки модели

Каждая модель хранит настройки выборов называемые настройками модели или просто «Настройками». В главном меню в опциях **Edit/Settings** вы можете изменять настройки. Настройки наследуются имитационной моделью, а так же результатами отчета модели. Некоторые настройки могут быть изменены даже после создания имитационной модели. Детали в части 2.

Имитационные объекты

Имитационный объект создается после разбора формулировки моделированного объекта. В основе используются **Commands/Create simulation** опции меню. После того, как имитационная модель успешно создана, команды используются для дополнительных настроек формы имитации. Эти команды могут быть частью моделированного объекта, или же могут быть посланы существующей имитационной модели как интеракции.

Транслятор

Высокий уровень транслятора - это часть GPSS World программы которая создает имитационные объекты. Все формулы модели транслируются перед регистрацией в имитации. Подобным образом симуляции транслируются в общие рамки перед посылкой в существующую имитационную модель.

Когда случаются ошибки, которые предотвращают создание имитационной модели, список сообщений помещается в моделированный объект. Затем, вы сможете переместиться к следующему сообщению об ошибке кликом мыши в **Search/Next Error** опции меню или используя кнопки доступа. Каждая ошибка описана в статусной линии внизу главного окна, и курсор мыши прыгает к расположению текста в модели, где вы можете исправить ошибку. Корректирование и трансляция быстры и просты. Есть специальные команды, **Commands/Retranslate** для ретрансляции модели в существующую имитационную модель.

Отчеты объектов

Одна из главных особенностей GPSS это стандартная форма отчетов. По существу, можно без усилий для разработчика моделей, имитировать автоматический запуск отчетов на всех существующих GPSS моделях, когда счетчик имитации достигнет нуля.

Текстовые объекты

Текстовые объекты это простой путь представить понятный текстовый файл в GPSS World. Их главное использование направлено на команды INCLUDE. Это позволяет сделать доступным утверждение модели из моделированного объекта. Вы даже можете загрузить утверждение INCLUDE в ключ функции. В этом случае комплекс списка команд и/или PLUS процедура в текстовом файле может быть послана имитации одним нажатием клавиши.

Текстовый объект разворачивается и в других местах. Потоки данных позволяют имитации прочитать и записать данные из текстового файла, и сделать результирующий файл для дальнейшего анализа. Как вы могли ожидать эти файлы представляются, как текстовые объекты.

1.3 Архитектура

На уровне интерфейса, GPSS World это введение Document/View архитектуры попросту в Windowstm приложения. Объекты могут быть открыты многими способами, изменены и сохранены на постоянную память путем последовательных интуитивных действий пользователя. Хорошо знакомо меню главного окна с «выколотыми» редко используемыми командами меню, ведет пользователя к его/ее цели с минимальными отвлечениями внимания. Внутренне GPSS World был разработан с преимуществом для близкой интерактивности даже для мультизадачной виртуальной памяти системы. Это обсуждается дальше.

1.3.1 Многозадачность

Многозадачная система GPSS World позволяет запускать много имитаций и экспериментировать одновременно. Не только улучшен экран, ввод пользователя, дисковые вводы/выводы, печать, и имитация так же, а любое число имитаций может быть запущено одновременно.

1.3.2 Виртуальная память

Имитации не ограничены размером физического доступа случайной памяти(RAM), где запускается имитация объекта. Использование виртуальной памяти имитации может быть расширено до гигабайта. Нет других прямых существующих пределов налагаемых на оперативную память, при условии предоставляемой адекватной замены места. Для оптимального представления, вообще, реальной памяти может быть достаточно. Все существующие размещения и управления невидимы для пользователя. Объекты создаются автоматически, если дополнительной информации не требуются. REALLOCATE утверждения не используются.

1.3.3 Интерактивность

GPSS World поддерживает высокий уровень интерактивности ваших имитаций. Вы можете посылать любые утверждения модели в существующую имитационную модель используя опцию Command в главном меню окна моделирования, Accelerator-ключи, или используя настройки модели, чтобы загрузить ключи функции с вашими самодельными командами. Вы можете использовать обычные диалоговые команды для выбора выражений в выпадающем вниз меню, и, используя INCLUDE команду, вы можете

посылать интеракцию любой сложности в имитацию. Даже INCLUDE команда может быть загружена в ключи функций, таким образом, обеспечивая любую комплексность, вызываемую нажатием клавиши.

Блок выражений, когда использован интерактивно, приводит активные транзакции к попытке ввода во временный блок, который затем отбрасывается. Этот режим называется «режимом управления моделированием» и очень полезен в течение экспериментов, обеспечивает доступность всех функций для имитации.

PLUS процедура и выражения экспериментов регистрируют установленные пользователем процедуры или эксперименты в имитационную процедурную библиотеку. Любые одноименные или однотипные процедуры будут заменены.

Команды, за исключением HALT и SHOW, когда получают существующий имитационный объект, помещаются в очередь имитационных команд для выполнения. HALT и SHOW незамедлительные команды, которые не помещаются в очередь, а исполняются сразу же. В дополнении, HALT убирает все ожидающие команды из командной очереди. CONDUCT команда инициирует PLUS эксперименты, которые берут под контроль имитационный объект. Эксперимент может основывать решения на результатах и вызывать статистический анализ, однако, это уменьшает интерактивность в течении эксперимента.

1.3.4 Визуализация

GPSS World отличается графическим представлением. Главная особенность GPSS World в том, что может быть одновременно запущено до двадцати разных окон, и с ними можно проводить различные действия. Нет особой сложности в управлении окнами, сохранении и печати для визуального представления состояния имитации. Поскольку снимок это картинка состояния на текущий момент времени, то имитационное окно обновляется оперативно, динамично меняет данные с их сохранением. Хотя динамичное окно показывает изменения состояния вводов, они так же заставляют имитацию запускаться более медленно.

Многие окна имеют не детализированный и детальный вид, показывают альтернативные настройки информации. Другие позволяют вам показывать запущенные часы модельного времени внизу окна. Вы можете выбирать показываемые опции через главное меню. Детали на появлении и операции для каждого окна в части пятой.

Снимки

Снимки доступны для:

- Данной цепи событий
- Будущей цепи событий
- Индивидуальных транзакций
- Нумераций групп
- Транзакций групп

Динамические окна

Любые нумерации неавтономных имитационных окон могут быть открыты следующими вводными типами:

- Блоки
- Выражения
- Логические переключатели
- Матрицы
- Очереди
- Сохраняемые входные данные
- Базы
- Таблицы

Таблицы окон представляют динамические гистограммы, которые очень важны для отображения накопленных входных данных, наблюдения выходных данных, и для обзора конвергенции начального проблемного распределения. Любые разноцветные неавтономные графики могут быть распечатаны, сохранены путем открытия одного или более дополнительных окон. Каждое окно прослеживает значения для до 8 главных выражений, включающие интегрированные значения. Графики могут быть просмотрены вертикальной и горизонтальной прокруткой в любых направлениях.

«Expression Window» может быть сильно увеличено. Используя этот вид, вы можете менять значения любых количеств PLUS выражений ваших собственных разработок. В любой точки имитационного моделирования вы можете открыть «Expression Window». «Blocks Window» показывает вводы транзакций в блоках. Это позволяет вам установить и убрать точки остановки, используя мышь или клавиатуру и визуально прошагать через имитацию.

Многие «Interaction Windows» имеют панель отладки, которая позволяет «заметить и исправить» ошибки, используя только мышь. Панель отладки наиболее точно обсуждается в части 5.

Часы модельного времени

Каждый имитационный объект имеет интернальный тумблер, переключение которого ведет к выводу меняющихся значений системных часов на самой правой панели статуса внизу окна. Это обеспечивает нормальную работу имитационной модели. Когда вы нажимаете на любой вид имитационной модели, то появляются часы, если вы их поставили в опциях. Это очень полезно для наблюдения статуса эксперимента.

Анимация

Разнообразная анимация поддерживается GPSS World. Уровень реализма варьируется от абстракции визуализации до высоко реалистичного динамического представления элементов.

Абстракция анимации

Всесторонне стилизованная анимация встроена в GPSS World. Эта возможность обеспечивается большим набором настроек имитационных окон, которые отображают GPSS сущности динамической имитации в течение всей имитации. Нет особых сложностей для данной анимации, только открыть окно. Графические изображения могут быть сохранены для включения в отчеты и/или печати.

Послепроцессная анимация

GPSS World включает дополнительный интерфейс, который обеспечивает имитацию пакетов анимаций по потокам данных. Фотореалистичные анимации могут быть изложены таким манером. Чтобы использовать этот интерфейс, постройте выходящий поток данных, повинующийся редактированию и управлению содержания в 3й части анимационного пакета. Поток данных рассматривается в следующем разделе.

Неавтономная анимация

GPSS World включает настройку динамического вызова PLUS процедур, которые позволяют вам осуществлять вызов функции в третьей части исполняемых файлов. Это может обеспечить неавтономные связи к третьей части анимационного пакета. Детали в части 8.

1.4 Язык моделирования

GPSS World содержит реализацию языка GPSS, универсальной системы моделирования, усиленной встроенным языком программирования PLUS (язык программирования для моделирования). Язык GPSS имеет различные включения, как-то свыше пятидесяти операционных блоков, и двадцати пяти команд. Так же включает в себя свыше тридцати пяти систем счисления, которые обуславливают мгновенную и полную конвертацию структур переменных и повсеместную доступность в рамках модели.

PLUS – маленький, но очень мощный язык процедурного программирования, продуцирующий более двенадцати типов выражений.

Многое от этой мощности взято из большой библиотеки процедур, содержащей математические и строковые функции и строго ориентированной на создание дистрибутивов.

В GPSS WORLD модель определена как последовательность выражений модели. Как то любое выражение GPSS, процедурные выражения или экспериментальные выражения PLUS. Выражениями GPSS также в свою очередь считаются блоки выражений и команды. За исключением функций повторителей выражений, все выражения GPSS должны записываться в одной текстовой строке, не превышающей 250 символов.

Любые выражения моделирования могут быть частью модели, записанной в файл модели, или быть переданы в интерактивном режиме для модификации существующего моделирования.

Включение выражений в язык PLUS и язык GPSS сильно увеличивает мощность моделирования и облегчает использование языка. Использование символа «*» для обозначения умножения определено по умолчанию. Очевидное отличие этой операции от операции GPSS – это не прямая адресация операции, которая обозначается указанным символом «*». Между прочим, вы можете переключить назначение этой операции, если захотите. Это выполняется на первой странице установок объектной модели.

Команды обсуждаются в главе 6, блоковые выражения в главе 7 и язык PLUS, включая процедурную библиотеку в главе 8.

1.4.1 Что нового

Много изменений было сделано в языке моделирования. Различие управляющих выражений было ликвидировано. Все выражения GPSS, которые не являются блоковыми, не есть просто вызываемые команды. Линии чисел игнорируются в GPSS WORLD, если используются, то должны начинаться в первой колонке текстовой строки. Автоматическое округление было удалено из GPSS World. Теперь вы должны использовать процедуру INT() или какой-либо другой метод, если вы хотите, чтобы промежуточный числовой результат был сокращен. Это выполняется для всех выражений, даже для VARIABLE и BVARIABLE. Так же, числовые типы, возвращающие значение в соответствии с системой числовых атрибутов теперь могут быть типа integer, real или string, в соответствии с протоколами SNA.

PLUS-выражения могут записываться в текстовые строки вне зависимости от их номера. Хотя выражения GPSS за исключением функциональных последовательностей должны быть записаны в единичную текстовую строку, максимальной длиной 250 символов.

Вставляемые математические выражения теперь могут использоваться где угодно. Выражения являются встроенной частью языка PLUS, но они также используются для расширения мощности операндов в GPSS World.

Допустимые формы для каждого операнда приведены в описании выражений в главе 6 и 7.

К тому же было создано много новых типов выражений. Они рассмотрены далее.

Некоторые новые возможности доступны как в студенческой, так и коммерческой версии.

- Генератор автоматической оптимизации экспериментов
- Генератор автоматической сортировки экспериментов.
- Захваты, доступные в студенческой версии.
- Мультиканальная библиотека процедур «ANOVA»
- «CALL» - библиотечная процедура для вызова внешних функций.
- Настройки трассировки PLUS
- Сеансный режим BATCH
- Автосохранение при выходе.

- Блок создания диалогов.
- Регулируемая остановка модели.
- Потокные процедуры данных PLUS.
- Новые выражения GPSS

Были добавлены некоторые новые блоки.

Новые GPSS-блоки:

- **ADOPT** – изменение сборочных комплектов.
- **DISPLACE** – изменить следующий составляющий блок данной транзакции. Может быть использовано как переназначающий элемент.
- **PLUS** – исполняет выражения языка PLUS и сохраняет их как параметр.
- **INTEGRATION** – включает или выключает интеграцию пользовательских переменных
- **OPEN** – инициализирует, поток данных для экспериментов.
- **CLOSE** – окончания операций с потоком данных.
- **READ** – перенесения следующей строки данных из потока данных в параметр текущей транзакции.
- **WRITE** – отправка строки, представляющей оценку параметра транзакции в следующий слот потока данных.
- **SEEK** – Изменение текущей позиции строки данных в потоке данных.

Новые команды

Добавлены выражения для контроля за интеграцией непрерывной структуры переменных, для поддержки составной структуры модельных файлов и библиотек процедур, а так же осуществления экспериментов PLUS.

- **CONDUCT** – запускает первоначальную регистрацию PLUS-экспериментов.
- **EXIT** – завершает работу, по выбору сохраняет объекты.
- **INTEGRATE** – автоматически интегрирует простейшее дифференциальное уравнение с опциональной транзакцией генератора значений триггера.
- **INCLUDE** – вставляет текстовый объект для транзакции.

1.4.2 Элементы языка

GPSS WORLD предоставляет целый ряд языковых элементов для удобства осуществления комплексного моделирования.

Полиморфные типы данных

Переменные теперь могут принимать значения одного из четырех типов. Переменные, контролируемые пользователем, такие как сохраненные значения, матричные элементы, параметры транзакции, и пользовательские переменные могут принимать значения типа integer, string, real, или неспецифические значения. Переменные времени принимают тип integer с двойной точностью.

Конвертация между переменными из одного типа в другой происходит автоматически. Процедуры, которые требуют специфических типов данных для аргумента, приводят аргумент к соответствующему типу. Если вы передаете строку процедуре, которая требует числового аргумента, то будет использоваться числовой эквивалент строки.

Так же, если вы пытаетесь применить команду «чтение» к числовому значению, то оно конвертируется в строковое автоматически.

Строковые данные имеют много применений. Они могут использоваться в потоках данных для форматирования отчетов и файлов результатов, а так же непосредственно для прямого доступа к данным. Строковые константы отмечаются парными двойными кавычками.

Вы можете использовать две двойные кавычки вместе для отображения единичных кавычек внутри строки. Например, вы хотите использовать совокупность из шести

символов двойных кавычек для представления строк внутри строки. Внутренние строки представляются как начинка «бутерброда», двойные парные кавычки, как «хлеб», формируется структура бутерброда в бутерброде. Библиотека процедур содержит много строковых процедур, которые могут быть использованы для создания и управления строками. Теперь вы можете передавать неспецифические типы данных в объекты сохранения значений и в матричные объекты. Команда INITIAL позволяет вам назначать тип значениям сохранения, матричным элементам и даже самим матрицам. Если нестандартная величина используется в операции, которая требует стандартного значения, то происходит остановка и выдается сообщение об ошибке.

Инициализация структур данных

Переменные и элементы матрицы могут быть инициализированы несколькими путями. Команда INITIAL может использоваться для присвоения значений элементам Logicswitch, Savevalues и матричным объектам.

- SAVEVALUE присваивает значения объектам Savevalue.
- MSFVEVALUE присваивает значения матричным объектам.
- LOGIK присваивает значения объектам Logicswitch
- READ и ASSIGN присваивает значения параметрам транзакции.
- Выражения присваивания языка PLUS могут быть использованы для присвоения значений пользовательским переменным и матрицам.
- Команда CLEAR сбрасывает значения объектов Savevalues, Logicswitches и матриц в целочисленный ноль.

Это действие может пресекаться параметром OFF операнда В в CLEAR. Скорее всего вы будете использовать опцию OFF в CLEAR при проведении экспериментов, иначе результирующие матрицы не будут сохранены.

Часто полезно изолировать инициализационные выражения в текстовый объект. Тогда, только команда INCLUDE понадобится для проведения полной инициализации. Это может быть выполнено в диалоговом режиме.

PLUS эксперименты могут контролировать инициализацию на любом уровне детализации. Любая команда инициализации может запрашиваться в ходе эксперимента посредством использования библиотеки процедур DoCommand.

Многомерные матрицы

Матрицы могут иметь до шести измерений. «Dynamic Matrix Windows» отображают любое поперечное сечение матрицы, и могут быть пронумерованы любым номером окон матриц. Временные матрицы могут создаваться для использования во время вызова процедур. Операторы присваивания PLUS используются для инициализации матриц размерностью три и более.

Выражения

GPSS World поддерживает широко распространенные выражения. Они могут быть вставлены в процедуры PLUS, или, заключены в круглые скобки в выражения GPSS. Это означает, что высокий уровень мощности вычислений может быть достигнут уже в операндах блоков и команд. Выражения могут выполнять простые вычисления, вызывая процедуры, которые выполняют операции математического характера или операции над строками, выборку возможных распределений или осуществляют реализацию пользовательских алгоритмов, включая файлы ввода-вывода.

1.4.3 Встроенный язык программирования

Язык программирования моделирования PLUStm, это простой, но мощный встроенный язык программирования, способный удовлетворить вычислительным потребностям пользователей, которые требуют хороший уровень управления структурами данных, вычислительных алгоритмов и специальных блоковых процессов.

Процедурные выражения PLUS могут быть использованы для определения процедур, как части первоначальной модели или они могут быть импортированы для использования в уже исполняемом моделировании. Это хорошо применяется в экспериментах PLUS выражений.

GPSS-блок, PLUS, позволяют пользователям вызывать процедуры PLUS определенные как блок. В результате пользователь может создавать его собственные блоки с комплексными методами, как ему будет угодно.

Процедуры PLUS просты в создании и доступны повсеместно в рамках модели и могут быть включены в библиотеку процедур. Так же они могут быть переопределены в любое время, даже интерактивно.

Типы выражений PLUS обсуждаются в главе 8. Они включают:

Assignment·BEGIN

DO...WHILE

END

EXPERIMENT

GOTO

IF ...THEN...ELSE· PROCEDURE

Procedure Call

RETURN

TEMPORARY

1.4.4 Библиотеки процедур

GPSS World имеет широкий набор функций, которые могут быть доступны, как вызываемые процедуры.

Библиотека строковых процедур

Для облегчения манипуляций с новыми строковыми типами, доступна встроенная библиотека строковых процедур. Она включает:

- **Align** – Возвращает копию одной строки, перемещенной в другую, выравнивает по правому краю.
- **Catenate** – Возвращает копию двух строк совмещенных в одну.
- **Copies** – Создает строки из многих копий ее же самой.
- **Datatype** – Возвращает строку означающую тип данных аргумента.
- **Find** – Возвращает смещение одной строки в другой.
- **Left** – Возвращает копию подстроки, начиная слева.
- **Length** – Возвращает длину строки в символах.
- **Lowercase** – Возвращает строку с заглавными символами.
- **Place** – Перемещает одну строку в другую. Выравнивание по левому краю.
- **PolyCatenate** – Возвращает копию одной или более строк, объединенных в одну.
- **Right** – Возвращает копию подстроки, начиная справа.
- **String** – Конвертирует данные в их строковый эквивалент.
- **StringCompare** – Возвращает целочисленный результат сравнения строк.
- **Substring** – Возвращает копию подстроки строкового аргумента.
- **Trim** – Удаляет пробелы между словами.
- **Uppercase** – Возвращает эквивалент строки в верхнем регистре.
- **Value** – Возвращает числовой эквивалент строки.
- **Word** – Возвращает копию одного из слов в строке

Библиотека математических процедур

Общая математическая библиотека обычно включает:

- **ABS** – Абсолютное значение
- **ATN** - Арктангенс

- **COS** - Косинус
- **INT** - Округление
- **EXP** - Экспонента
- **LOG** – Натуральный логарифм
- **SIN** - Синус
- **SQR** – Квадратный корень
- **TAN** – Тангенс

Новые процедуры транзакционных запросов

Они включают:

- **QueryXNExist** – Определяет существование транзакции.
- **QueryXNParameter** – отыскивает значение параметра транзакции.
- **QueryXNAssemblySet** – отыскивает установочный комплект транзакции.
- **QueryXNPriority** – находит приоритеты транзакции.
- **QueryXNM1** – находит временную метку транзакции.

Новые сервисные процедуры

Новые сервисные процедуры являются инструментом поддержки экспериментов:

- **DoCommand** – Переводит командную строку в глобальную среду и посылает ее в объект моделирования.
- **ANOVA** – Выполняет анализ переменных. Новые мультисканальные процедуры ANOVA теперь могут перемещаться вручную, и ограничены факторами взаимодействий.
- **Effects** - Вычисляет эффекты от полученных выборок эксперимента.
- **Exit** – Заканчивает сеанс работы GPSS World сохраняет все объекты, и элементы не относящиеся к ним, или запрашивает пользователя об обратном.

Новые сервисные процедуры есть средство коммуникации с программами во внешних выполняемых модулях.

- **Call- Invoke** - функция в выполняемом модуле. Не передает аргументов. По умолчанию возвращает значения типа integer.
- **Call_Integer- Invoke** функция в выполняемом модуле. Передает целочисленный аргумент. По умолчанию возвращает значение целого типа.
- **Call_Real- Invoke** функция в выполняемом модуле. Передает аргумент типа real. По умолчанию возвращает значения типа integer.
- **Call_String- Invoke** функция в выполняемом модуле. Передает строковый аргумент. Возвращает значение целочисленного типа.

Свыше двадцати распределений доступны:

- | | |
|---------------------------|----------------------------|
| • Beta | • LogLaplace |
| • Binomial | • LogLogistic |
| • Discrete Uniform | • LogNormal |
| • Exponential | • Negative Binomial |
| • Extreme Value A | • Normal |
| • Extreme Value B | • Pareto |
| • Gamma | • Pearson Type V |
| • Geometric | • Pearson Type VI |
| • Inverse Gaussian | • Poisson |
| • Inverse Weibull | • Triangular |
| • Laplace | • Uniform |
| • Logistic | • Weibull |

1.4.5 Транслятор высокой производительности

Характерная черта GPSS World это новый высокопроизводительный транслятор моделей, который в сотни раз быстрее, чем его процессор. Самые большие модели преобразуются не более чем за несколько секунд. Любые обнаруженные ошибки размещаются в циклической очереди сообщений, так что они могут быть скорректированы с помощью полноэкранного редактора в текстовом просмотре, находящемся в объекте моделей. Коррекция ошибок помогает автоматическому размещению курсора на следующем проблемном лексическом элементе в модели. Листинг ошибок легко проследить с помощью прокрутки мыши или клавиатуры. К тому же эти листинги сохраняются вместе с объектом модели.

1.4.6 Блок создания диалога

Создание GPSS диалогов в GPSS World может выполняться произвольно с помощью мастера, путем заполнения форм. Вам доступна эта опция через меню **Edit / Insert Block**, которая открывается в GPSS Block Menu Window. Вы можете открыть «создание диалога» для любого GPSS Block, щелкнув соответствующую кнопку. Когда вы щелкаете на кнопку ОК после заполнения информации необходимой в блоке, результирующее выражение проверяется на ошибки синтаксиса, если все верно, оно внедряется в модель сразу после строки содержащей мерцающий курсор.

1.4.7 Автоматические генераторы экспериментов

Теперь GPSS World имеет возможность генерации экспериментов, записываемых на языке PLUS и вставляемых в объект модели. Все что вам нужно сделать это заполнить форму в нескольких диалоговых окнах, доступных через команду **Edit** главного меню. Вы можете создать любой фильтрующий или оптимизирующий эксперимент автоматически. По умолчанию команда CONDUCT загружается в ключ функции, поэтому все, что вам необходимо сделать, это запустить эксперимент для создания объекта моделирования и нажать функциональный ключ. Это все и более, описано в главе 13 и в уроке 19 и 20 документа *GPSS World Tutorial Manual*.

1.5 Совместимость

Последующий раздел содержит широкий сравнительный обзор GPSS World и его предшественника GPSS/PC. Рекомендуется пропустить этот раздел, если вы не имели опыта работы с GPSS/PC.

GPSS World совместима с GPSS/PC, и дает статистически неразличимые результаты. Этот уровень совместимости доступен по причине простой коррекции нескольких различий.

Добавим, что высокий уровень совместимости доступен благодаря возможности вызова режима совместимости GPSS/PC. В большинстве случаев вы можете достигать точного дублирования результатов. К тому же GPSS World использует новую «run time» библиотеку. Числа с плавающей точкой при округлении мало отличаются от используемых в GPSS/PC. К тому же большинство моделей GPSS/PC могут быть незначительно модифицированы для получения аналогичных результатов при запуске коммерческой версии GPSS World в режиме совместимости GPSS/PC. Процедура, которой необходимо следовать, детализирована ниже после обсуждения разницы между GPSS World и GPSS/PC.

1.5.1 Отличия от GPSS/PC

GPSS World имеет много отличий от GPSS/PC. GPSS World базируется на идее, что текстовая модель объекта построена и/или модифицирована и затем скомпилирована в порядке создания имитационного объекта. Это отличает GPSS/PC от взглядов «Program File and Simulation». В GPSS World операции загрузки модели GPSS/PC были полностью заменены. Вместо сканирования каждой линии одновременно, GPSS World сканирует сразу полный экран и создает меню команд имитации. Любые ошибки, обнаруженные в течение компиляции сохраняются в очередь сообщений ошибок, таким образом они могут быть легко найдены и скорректированы. Скорость транслятора гарантирует быстрое обнаружение/корректировку ошибок.

GPSS World не исполняет предотвращения ошибок нажатия клавиш, как было в GPSS/PC. Вместо этого, GPSS World использует модельный компилятор для создания имитации. Это уменьшает время, затраченное на загрузку в сотни раз. Однако, это меняет механизм обнаружения ошибок. Сейчас ошибки отслеживаются в течение трансляции, и корректируются выбором «следующая ошибка» из поиска подменю в главном окне. Курсор затем автоматически перемещается на ошибку и пишется сообщение об ошибке в статусной линии внизу главного окна.

Возможно следующее заметное изменение - это отпадение необходимости в линии чисел, как факт, который игнорируется в GPSS World. Это значит, что позиционирование блоков больше не специфицируется как линия чисел блоков, только относительное положение передается в транслятор. С нового выражения INCLUDE, позволяющего модельному объекту содержать другие ясные текстовые файлы, эта последовательность блоковых выражений читается транслятором как диктовка позиций блоков имитации. Так же блоки больше не вставляются в течение имитации, они поворачиваются назад так, что транслятор их задерживает, пока не наступит наиболее благоприятное время для их вставки. Даже высокоуровневая интерактивность задерживается. Это значит, что любая сущность модели может быть использована в течение имитации, как интерактивная модель.

Уровень интерактивности GPSS World был сохранен. Любая модельная сущность может быть передана в существующую имитацию для исполнения. По факту, PLUS Procedures может быть объявлена, или даже переобъявлена «на лету». В случае команд GPSS, это обычно переобъявленные входы, или контроллеры запуска имитации. В случае выражений блока, так же как и в GPSS/PC «главная имитация» блок входа ограничивается активной транзакцией. Все это используется для подменю команд после того, как модель скомпилирована.

Автоматическое усечение было убрано из GPSS World. Теперь вы должны использовать подробно метод INT() процедуры, или нескольких других методов, если вы желаете урезать промежуточные численные результаты. В зависимости от типа данных, возвращаемый System Numeric Attribute, может быть теперь integer, real, string. Даже эта SNA возвращает real с двойную точность между 0 и 1000, включительно. Старые исключения и integer SNA могут быть включены, используя режим совместимости GPSS/PC.

Многозадачная архитектура GPSS World привела к другим изменениям, начиная с сообщений, которые выводятся для представления команд или изменения статуса. Поэтому, обновление интерактивного окна происходит с помощью очереди сообщений принятых от имитации. Таким же образом, большинство команд принимаемых объектом моделирования имитационной командной очереди до начала их выполнения одной за другой. Существуют два исключения, HALT и SHOW которые выполняются немедленно. Помимо этого, HALT очищает командную очередь.

Остальные изменения приведены здесь, и более подробно рассматриваются в других местах этого руководства.

- Утверждения PLUS могут измерять количество текстовых строк. При этом утверждения GPSS должны быть записаны с помощью единичной текстовой строки, максимальная длина строки составляет 250 символов.
- Заключение математических выражений в скобки может быть использовано теперь во всех случаях использования SNA.
- Численные атрибуты системы теперь могут возвращать целые, вещественные или строковые значения в зависимости от самой SNA. В режиме совместимости GPSS\PS, SNA возвращает только целые значения, за исключением модификаторов функций.
- Новый числовой атрибут системы A1, был создан для того, что бы возвращать набор активных транзакций.
- Блок помощи был удалён. Он заменён на блок PLUS, который помогает добавлению завершённых процедур в модель, и блоки потоков данных которые помогают взаимодействовать с внешними файлами и программами.
- Выражение управления MICROWINDOW, было заменено на окно представления, которое открывается из подменю **Window/Simulation Window** главного меню главного окна.
- Выражение управления PLOT, было заменено на графическое окно, которое открывается из подменю **Window/Simulation Window** главного меню главного окна
- Выражение управления END, было заменено на EXIT которое может заканчивать сессию. END теперь ключевое слово языка PLUS.
- Выражение управления ANOVA, было заменено на библиотеку процедур ANOVA.
- Выражение управления EVENTS, было заменено на окна FEC и CEC, которые открываются из подменю **Window/Simulation Snapshot** главного меню главного окна
- Выражение управления GROUPS, было заменено на моментальные окна числовой и транзакционной групп, которые открываются из подменю **Window/Simulation Window** главного меню главного окна
- Выражение управления RESULT, было заменено на блоки потока входа-выхода, которые автоматически записывают данные в файл отчёта.
- Выражение управления WINDOW, было заменено на подменю «Window», оконного меню модели.
- Выражение управления USERCHAINS, было заменено на окно «Userchain», которое открывается из подменю **Window/Simulation Window** главного меню главного окна
- Выражение управления ANITRACE было удалено. Анимация GPSS World обеспечивается потоками данных и внешним постпроцессором.
- Блок MOVIE был удалён. Анимация GPSS World обеспечивается потоками данных и внешним постпроцессором.
- Position Window и файл POSITION.GPS были удалены. Анимация GPSS World обеспечивается потоками данных и внешними анимационными пакетами.
- Символ включения файла @ был заменён на INCLUDE, который использует взятую в двойные кавычки спецификацию файла как операнд.
- Численный атрибут системы Z1 больше не отображает доступность общего объёма физической памяти. Теперь он возвращает значение максимального объёма памяти, которая может быть локализована, как операционная система.
- Следующие функции управления GPSS\PC были заменены графическим интерфейсом пользователя с полноэкранным редактором и файловыми опциями: DELETE, DISPLAY, DOS, EDIT, RENUMBER, и AVE.

- Целые, вещественные и строковые типы данных поддерживаются в «сохранённых значениях», матрицах, значениях имени и параметрах транзакции. Значения часов может быть целым и вещественным. Неограниченная точность числовой арифметики GPSS\PS была модифицирована.
- Команды REPORT выполняются теперь в режиме NOW. Операнд A более не используется и должен быть равен нулю.
- Библиотека процедур PLUS теперь может использоваться в PLUS выражениях. Поддерживаются процедуры управления строковым типом данных и более 20-ю встроенными возможностями.
- Матрицы GPSS не могут иметь более 6 измерений. Любой двумерный поперечный разрез может отображаться динамически в окне матрицы.
- Поменялся приоритет операторов. Новый приоритет рассмотрен в главе 3.

1.5.2 Модифицирование старых GPSS\PS программ

Этот раздел для пользователей GPSS/PC, которые желают перевести существующие модели GPSS/PC в среду GPSS World. Он содержит список настроек, которые вы должны произвести для ваших программных файлов GPSS/PC. Это обеспечит вас данными, незначительно отличающимися от данных полученных в моделированиях GPSS/PC. Вам не потребуется режим совместимости GPSS/PC для доведения до конца этой степени соответствия. Несмотря на это, если вы пожелаете добиться точного копирования данных, вы должны выполнить действия, указанные в следующем разделе.

Существуют множество деталей, предназначенных обеспечить легкий переход к новой системе. Все пользователи GPSS/PC должны записать небольшой номер расширений, который может быть запрошен для соответствующих настроек ваших имеющихся массивов данных модели. Другая часть состоит в том, что пользователи, желающие преумножить в определенном количестве данные GPSS/PC, должны установить достоверность модели при работе с GPSS World. После этого важно переключиться к новому режиму управления, включающему в себя часы с плавающей запятой и множество других деталей.

Опустим сейчас номера строк. Не используйте старые программные файлы, которые могут переместить GPSS/PC-узлы по номерам строк. Если вы используете их в качестве включаемых файлов (Include-files), они добавятся к размеру имитации, если вы используете их как интерактивные команды, они одновременно будут выполнять функции управления выражениями моделирования. Вы должны произвести следующие настройки программы:

1. Если Вы используете файл по предотвращению ошибок при нажатии на клавиатуру (Keystroke Error Prevention) GPSS/PC для автоматического набора ключевых слов, вы должны набирать текст в прежнем порядке до того, как переходящая модель будет под GPSS World.
 2. Поменять местами @ и INCLUDE.
 3. Спецификации файла должны быть в двойных кавычках в INCLUDE.
 4. Удалите все команды REPORT.
 5. Удалите команды END и какие-либо метки, такие как BEGIN, COUNT, NORMAL и так далее, которые сейчас могут конфликтовать с ключевыми словами GPSS World.
 6. Поменяйте местами блоки HELP и PLUS и/или Data Stream Blocks (блоки потока данных).
 7. Удалите блоки MOVE. Позиции Windows не поддерживаются длительно.
 8. Заклучите в скобки все выражения в VARIABLE, FVARIABLE и BVARIABLE.
- Вероятно, безопаснее удалить все старые пусковые команды со старого программного файла GPSS/PC до того, как вы проверили свою измененную модель. Не используйте

номера строк при перемещении блоков в «Savable Programm». Это не займет много времени.

1.5.3 Строгое увеличение данных

Большинство программных файлов GPSS/PC могут быть составлены для выработки таких же результатов, что и при запуске GPSS World. Однако, из-за различий в окружении вещественных чисел вычисления с плавающей запятой, вам может потребоваться сделать несколько добавочных модификаций к программному файлу GPSS/PC, в дополнение к предыдущему разделу. Если вам необходимы только данные, которые будут статистически неразличимы от полученных в GPSS/PC, вам не понадобится исполнять следующие пункты использования коммерческой версии GPSS World.

1. Первый шаг направлен на режим совместимости в свойствах модели GPSS/PC. Он определяет возможность GPSS World использовать целочисленный счетчик и завершать процесс вычислений данных, точно так же, как делала это GPSS/PC.

ВЫБЕРИТЕ Файл/Открыть,

чтобы открыть Модельный Объект (Model Object)

ВЫБЕРИТЕ Правка/Установки.

Затем на первой странице проверьте чекбокс, помеченный как GPSS/PC совместимость (Compatibility).

2. Теперь установите случайные потоки чисел из GENERATE, ADVANCE и TRANSFER, чтобы установить соответствие с SETTINGS. GPS-файл вы используете с GPSS/PC. Далее, установите случайные потоки чисел временных связей к 1.

ВЫБЕРИТЕ Правка/Установки,

затем выберите случайную числовую страницу. Установите входные поля случайного числового потока для GENERATE, ADVANCE, TRANSFER, и временные связи.

3. Убедитесь, что все составляющие в любых RMULT-командах меньше, чем 100,000,000.

4. Удалите все зависимости на идентификаторе, приводящие к невыполнению системных значений. Например, RN\$IDENTIFIER-составляющие потока случайных чисел с четкой системой неправильных значений. Эта составляющая определенно нуждается в том, чтобы быть измененной.

5. Не запрашивайте случайную функцию (Random Function) от случайной функции. Замените любые такие запросы.

6. Замените все дроби, случающиеся в TRANSFER-блоках на целочисленные.

7. Убедитесь, что случайные функциональные категории GPSS (Function Entities) не позволяют любой паре функциональных повторителей (Function Follower) ассоциироваться с вероятностью $1/n$, где n - число пар. Например, если функция определяется 4 пунктами, не допустимо любой вероятности быть 25. Вы можете сделать это, добавляя дополнительные пункты, прежде чем $1/n$ станет повторяться, или не ассоциироваться с любыми парами функциональных повторителей. FN\$SNORM и FN\$XPDIS не должен меняться.

Специальная DOS-программа, именуемая PCAID.EXE, предусмотрена, чтобы помочь вам модифицировать ваш программный файл GPSS/PC. Она откорректирует номера строк и напомнит вам, если у вас имеется GPSS функция, заявлением, что она нуждается в проверке. Вы можете начать работу с этой программой в DOS-окне.

В любом случае, помните, что модельные объекты - не простые текстовые файлы. Они содержат установки, закладки, и списки данных, которые не могут быть прочитаны внешними программами для обработки текста. По этой причине, когда вы завершите модифицировать ваш программный файл GPSS/PC, вы должны использовать программу для обработки текста, чтобы скопировать это в окна буфера обмена. Затем откройте в

GPSS World новый модельный объект, выберите **Файл/Создать Новый** и вставьте текст в предназначенное место. Выберите **Правка/Вставить**. Этот процесс создает действительный модельный объект GPSS World, основанный на вашей первоначальной модели. Сейчас вы должны быть способны получить соответствующие данные для моделирования запуска GPSS/PC и GPSS World.

Глава 2 - Управление GPSS World

Эта глава содержит обсуждение организации возможностей, доступных с GPSS World, и информацию по управлению ими. Сначала мы проходим через процедуру установки, а затем обсуждаем нормальное управление установленного пакета GPSS World. Специфические окна обсуждаются кратко. Дополнительная обработка каждого типа окна может быть найдена в главе 5. Объяснения сообщений об ошибке - в главе 14.

2.1 Установка

Нормальная установка GPSS World размещает копию исполняемых файлов, файл помощи, и модели образцов GPSS в папки, иногда называемые директориями, в вашем компьютере. Сетевая HTML-документация, как, например, это справочное руководство и учебное пособие по GPSS World нормально включаются в установку коммерческой версии GPSS World. Для установок обучающей версии HTML-справочники должны быть загружены и/или скопированы отдельно. Оба справочника имеются в распоряжении в печатной форме. Все установки включают Справочное руководство в виде онлайн помощи.

2.1.1 Обзор

Установка начнется, когда вы двойным щелчком загрузите «Microsoft Windows Installer» (msi) файл с ресурса Minutemansoftware.com, или когда вы запустите стартовую программу, подлинную» GPSS World Commercial Version Distribution CD». К сожалению, мы не способны поддерживать программное обеспечение, полученное другим путем.

Существуют 2 важных пункта, с которыми необходимо ознакомиться вначале. Во-первых, вы должны убедиться, что если ваши существующие модельные объекты устарели, они будут сохранены, т.е. удобны и годны к работе, после установки. Во-вторых, в этом случае вы должны переустановить любую существующую установку такого же продукта GPSS World.

2.1.2 Сохранение устаревших объектов

Если вы заменяете существующую установку GPSS World, вы должны сначала определить усовершенствованы ли ваши старые модельные объекты. Хотя мы стараемся не навязать вам это неудобство, но иногда внутренний формат модельных объектов должен быть изменен. Перед тем, как вы начинаете установку, загрузите текущие уведомления или просмотрите страницу общей информации на Minutemansoftware.com, чтобы определить совместимость.

Если модельные объекты новой версии, которые вы установили, несовместимы с вашими уже существующими модельными объектами, вам необходимо сохранить ваш текст для каждого существующего модельного объекта перед тем, как вы продолжите установку. Чтобы это сделать, откройте каждый из ваших модельных объектов и используйте буфер обмена Windows, чтобы сохранить текст в каждом модельном объекте (.gps файл) в только что созданном текстовом объекте. Затем, после установки, вы можете снова использовать буфер обмена, чтобы подать модельные операторы в только что созданные модельные объекты нового вида. Вы можете не обновлять существующие текстовые объекты. Если вы не используете по умолчанию модельные установки, вам будет необходимо, изменить установочные параметры в каждом новом модельном объекте. Для этого, вы можете пожелать записать все старые установочные параметры перед инсталляцией. установочные параметры могут быть рассмотрены или могут быть изменены, с помощью пункта меню **Правка/Установочные параметры** в главном окне. Обычно, новые

модельные объекты становятся совместимыми со старыми, поэтому вам не придется проделывать все это. Хотя внутренний формат модельных объектов обычно не изменяется от одного выпуска GPSS World к следующему, это не верно для объектов моделирования. Обычно, но не всегда, вы должны вновь ретранслировать ваши модели, используя новую версию GPSS World, тем самым, создавая новые объекты моделирования. До тех пор пока ваши старые модельные объекты совместимы с новыми, это просто. Единственное, из-за чего надо беспокоиться, это если вы внесли изменения в параметры настройки старых объектов моделирования. Если так, вы должны будете изменить также и параметры настройки в новом объекте моделирования, используя меню **Правка/Установочные параметры**. Вы можете захотеть записать старые значения, перед тем как приступить к изменениям.

2.1.3 Деинсталляция прежней GPSS World

Перед тем как начинать установку, мы рекомендуем, чтобы вы удалили любую существующую старую версию того же продукта GPSS World. (Student и Commercial version - это различные продукты). Это не удалит ваши собственные объекты, только те, которые были установлены первоначально с GPSS World. Операция деинсталляции осуществляется с помощью кнопки **Добавить/Удалить папку** в контрольной панели вашей операционной системы. Как правило, ключ программного обеспечения вашей старой Commercial version сохранится, если вы установите в ту же директорию. Вы можете установить Student version GPSS World, даже если вы уже установили Commercial version, и наоборот.

После того, как любая другая старая установка удалена, вы готовы установить GPSS World. Если вы устанавливаете его с CD в систему с возможностью автозапуска, просто со вставкой CD начнется установка программы. В противном случае, откройте CD-иконку в папке «Мой Компьютер» и двойным щелчком по ней начните установку. Это даже легче, чем если вы устанавливаете программу с World.exe или Student.exe самоизвлекающегося файла, загруженного с ресурса Minutemansoftware.com. Всего лишь двойной щелчок на этой иконе и установка GPSS World начнется.

2.1.4 Требования к установке

Сначала, вы должны проверить, что требования к установке были соблюдены.

Требования к оборудованию:

1. Использование пакета GPSS World предполагает использование совместимого компьютера IBM с операционной системой Windows 98 или совместимой операционной системой.
2. Мы рекомендуем Intel Pentium III CPU.
3. Математический сопроцессор с плавающей запятой особенно рекомендуется, но не требуется. Большинство микросхем центрального процессора, продающихся сегодня, имеют таковые.
4. Мышь особенно рекомендуется. Хотя моделирование может быть создано и может быть запущено без мыши, некоторые дополнительные операции требуют ее использования.
5. Рекомендуется по крайней мере 32 Мб RAM.
6. Требуется по крайней мере 15 Мб свободного пространства жесткого диска для установки всех компонентов Коммерческой Версии. Больше пространство рекомендуется для того, чтобы содержать Модельные Файлы. Кроме того, соответствующее обменное пространство должно быть доступно в менеджере виртуальной памяти Windows.
7. CD привод требуется, только если установка происходит с диска.

8. 16-значный пользовательский ключ требуется для управления программным обеспечением. Этот числовой ключ должен быть получен от Minutemansoftware.com после того, как основной процесс установки завершится.

Ограничения версии

1. Student Version GPSS World ограничена для моделирования, содержащего не более 180 блоковых объектов GPSS, и Personal Version ограничена для 2000. Commercial Version может запустить моделирования миллионов блоков, вследствие этого она ограничена только размером виртуальной памяти, предусмотренной вашей операционной системой. Для оптимального исполнения вы должны иметь достаточно RAM на вашей ПК, чтобы все моделирования могли быть запущены в одно и то же время.

2. Student Version не может открыть моделирования, сохраненные Personal Version или Commercial Version GPSS World. Commercial Version может открыть любое моделирование такого же номера версии, как и другого совместимого с номером версии.

3. Student Version GPSS World не поддерживает режим совместимости GPSS/PC. А Personal Version и Commercial Version поддерживают.

2.1.5 Папки/Директории

Хотя вы можете выбрать и иначе, установочная программа GPSS World обычно устанавливает программное обеспечение в папку Minuteman Software в папке Program Files на вашем компьютере. Далее мы будем использовать "папку" и "директорию" синонимично. Commercial Version GPSS World будет размещаться в папке, именуемой Commercial GPSS World в папке Minuteman Software. Папка Samples, которая содержит все примеры, размещается здесь же. Файлы, относящиеся к управлению GPSS World обычно размещаются в "модульной директории", т.е. в папке, содержащей исполняемые файлы GPSS World. Другие совместно используемые исполняемые файлы могут быть размещены в одной из системных директорий вашего компьютера, если они уже не используются. Управление GPSS World приводит к созданию нескольких различных файлов на вашем компьютере. Файлы для уведомлений, регистрации информации, и пользовательские ключи создаются в модульной директории. Вы контролируете размещения других файлов, когда открываете и сохраняете объекты GPSS World. Точно так, как и объекты GPSS World, потоки данных могут быть направлены к специфическим директориям вашего открытого выражения GPSS. Если вы устанавливаете только файловое имя, а не путь, используется директория, содержащая ваш объект моделирования.

Иначе, если только вы не пользуетесь пунктом меню Файл/Сохранить как... , Объекты Моделирования будут размещаться с их родительскими Модельными Объектами, и Объекты Уведомлений будут размещаться с их родительскими Объектами Моделирования. Аналогично, поиск Включенных файлов начинается в папке Модельного Объекта, или поиск Текстовых Объектов Потока Данных - в папке Объекта Моделирования. GPSS World может сейчас активизировать внешние программы непосредственно динамической процедурой Call(). Правила для размещения исполняемых файлов, содержащих целевые функции, детализируются в главе 8.

Когда вы начинаете строить свои собственные модели, вы должны создать директории, которые могут опознаваться по отношению к проекту, с которым вы работаете. Например, если вы исследуете две очень различных стратегии контроля запасов, которые требуют две очень различных модификации, чтобы смоделировать их, вы можете найти преимущества в размещении их в отдельных папках. С другой стороны, варианты по теме могут быть лучше организованы в подпапках с тем же родителем.

2.1.6 Процедура установки

Пользование всеми продуктами GPSS World подтверждено соглашением о передаче прав пользования, которое может быть показано во время установки или выбора меню **Помощь/О программе** в главном окне. Как только вы начнете установку, установочная программа GPSS World проведет вас через процесс. После короткой инициализации появится первое окно установки GPSS World.

2.1.7 Активация программного обеспечения

После того, как вы установили файл GPSS World Microsoft Windows Installer (msi), загруженный с www.minutemansoftware.com, вы должны активировать программное обеспечение. Это делается, как только вы получили шестнадцатизначный ключ, и программное обеспечение может быть запущено в месте, где вы установили это. Если вы хотите использовать онлайн-метод активации, этот процесс почти автоматический.

Вы можете активировать установку GPSS World либо через онлайн-активацию, либо через электронную почту. В любом случае, сначала вы должны сделать предварительную оплату для активации, щелкая кнопку "Buy Now" на нашем сайте www.minutemansoftware.com/downloads. Поочередно, пользователи Personal Version или Commercial Version GPSS World могут связаться с нами непосредственно по электронной почте (sales@minutemansoftware.com) для предоплаты.

Когда вы активируете GPSS World, вы должны указать, каким образом вы совершили предоплату. Вы делаете это, уточняя номер счета ("Minuteman Invoice Number"), протокол счета PayPal ("PayPal Transaction Number") или свой авторизованный номер ("Minuteman Authorization number").

Первое из этого - номер счета ("Minuteman Invoice Number") постоянно хранится в cookie и, если не заблокирован на вашем компьютере, позволяет вам произвести активацию просто щелчком по одной единственной кнопке. Протокол счета PayPal ("PayPal Transaction Number") может быть использован также. Вы можете найти это при подтверждении приема электронной почты, посланной вам от PayPal. Наконец, иногда удобнее получить авторизованный номер ("Minuteman Authorization number") непосредственно от Minuteman Software, который позволяет вам приступить к активации.

После того, как вы указали ваш метод оплаты, вы можете выбрать либо онлайн-активацию, либо активацию по электронной почте. Он-лайн активация немедленно "попытается получить" ваш 16-значный ключ программного обеспечения с нашего регистрационного сервера. Если это пройдет успешно, вы сможете запустить моделирование на GPSS World немедленно. Если нет, вы можете повторить онлайн-активацию или, возможно, можете вместо этого активировать программное обеспечение по электронной почте.

Активация по электронной почте вызовет зашифрованное сообщение для отправки на наш регистрационный сервер. Если все будет в порядке, электронное письмо, содержащее 16-значный ключ ПО, будет послано на электронный адрес, который вы указали при регистрации. Если у вас возникли трудности при отправке электронного письма с компьютера, где был установлен GPSS World, существует опция, которая размещает зашифрованное сообщение в буфер обмена вашей операционной системы Windows. Вы можете тогда использовать текстовый редактор, чтобы переместить сообщение к месту, где вы можете создать электронное письмо для отправки на Activation@minutemansoftware.com. Ваш 16-значный ключ программного обеспечения будет послан на электронный адрес, который вы указали при регистрации.

2.2 Среда GPSS World

Сейчас мы начинаем обсуждение способов, с помощью которых вы можете контролировать вашу новую среду моделирования.

2.2.1 Главное окно

Среда моделирования GPSS World - это коллекция всех действий, имеющихся в вашем распоряжении, когда вы начинаете сеанс. Все контролируется из главного окна GPSS World, один пример которых представлен ниже.



Рисунок 2.1 – Главное окно GPSS World

Главное окно имеет несколько компонентов. Строка заголовка (Title Bar) находится вверху. Ниже нее располагается меню, ниже которого находится панель инструментов (Toolbar) (который может быть скрыто, выбрав команду меню вид/панель инструментов). Большинство действий происходит в большой пустой пользовательской области.

В самом низу главного окна располагается строка состояния, которая разделена на 3 части. Левая часть ее показывает подсказки команд, которые дают вам больше информации о пунктах меню, которые вы собираетесь использовать. Средняя часть строки состояния показывает состояние и сообщения ошибки, посланные от объектов GPSS World, с которыми вы взаимодействуете. И, с правой стороны находится маленькая часть строки состояния, которая используется для демонстрации изменений системных часов работающего моделирования. Вы можете включить или выключить их для каждого объекта моделирования путем использования команды меню **Вид/Показать часы**.

2.2.2 Быстрый обзор

Используя меню главного окна, вы можете создавать и манипулировать объектами GPSS World. Это: модельные, отчетные, текстовые и имитационные объекты. Каждый из них будет иметь один или более видов, которые могут быть открыты в пределах главного окна GPSS World.

Типичный проект начинается с создания и модификации модельного объекта. Модель - это только список модельных операторов, определенных в главах 6, 7, и 8. После модели составляется следующий шаг, который должен создать объект имитации. Каждый раз вы выбираете из главного меню пункты **Команда/Создать имитацию** или **Команда/Транслирование**, GPSS World переводит модельные операторы в работоспособный объект моделирования. Если ошибки нуждаются в исправлении, единственное нажатие клавиши (представляющее собой выбор в меню **Поиск/Следующая Ошибка**) может циклически повторяться через список ошибок, устанавливая курсор в текстовый вид модели перед каждой ошибкой.

Итак, объект моделирования успешно создан, и вы можете послать любое модельное выражение к нему, чтобы изменить структуру имитации. Обычно, как правило, вы пошлете команды, такие, как НАЧАТЬ (START), но вы не сможете повторно определить объекты или PLUS-Процедуры.

Вы можете послать модельные операторы к существующим имитационным объектам, используя один из нескольких методов.

Первый: некоторые из самых общих команд перечисляются в меню команды главного окна. Большинство имеют ускорители работы клавиатуры.

Второй: общий командный диалог вызывается через меню **Команда/Общее** в главном окне. Здесь вы можете записать любой модельный оператор - даже PLUS-Процедуру - и отправить его к существующему имитационному объекту.

Третий: вы можете загрузить набор функциональных клавиш со своим собственным набором команд. Это делается через меню **Правка/Установки** в главном окне. Впоследствии, нажатие функционального ключа посылает команду к только что выбранному объекту моделирования. Сложные PLUS-Процедуры и длинные командные списки могут быть отправлены, используя оператор INCLUDE, связанное с ранее созданным Текстовым Объектом. Вы можете вводить INCLUDE в качестве общей команды (Команда/Общие) или загружать это в качестве функциональной клавиши. После того, как INCLUDE-выражение загружается в функциональную клавишу (**Правка/Установки**), одним нажатием клавиши посылается целая часть к объекту моделирования для обработки данных.

Во время фазы отладки вашего проекта, уверенно используйте все инструменты визуализации, находящиеся в вашем распоряжении. Здесь имеются 10 динамических окон и 7 кадровых окон, отображающих статическое состояние набора данных, которые позволяют вам рассматривать работающую имитацию. Не требуется большого усилия с вашей стороны, кроме того, чтоб открыть их. Некоторые, подобно окнам блоков, добавляют запятую и отмечают отладочные средства для управления ограничителем условий и для контроля продвижения имитации в течение отладки.

Новая PLUS-трассировочная метка может быть включена в Блокноте установок. Она разместит описания каждого вызова PLUS-процедуры в вашем регистровом окне. Это полезная метка: если вы определяете PLUS-процедуру, которая принимает аргументы, но не делает ничего, вы можете разместить ее вызовы где угодно в вашем PLUS-коде, чтобы оценить и проследить свои собственные пользовательские переменные - глобальные и временные.

Среди динамических окон окна диаграммы и выражения позволяют вам видеть оценку любого PLUS-выражения в его динамическом изменении. Окно таблиц может также делать видимым конвергенцию распределений частоты. В дополнение имеется 7 кадровых

окон, отображающих статическое состояние набора данных, которые представляют подробное состояние информации в некоторый момент в пределах моделирования. GPSS World сильна в экстенсивности своих стандартных отчетов. Если вы не выбираете иначе, когда объект моделирования завершает имитацию (т.е. конечный номер достигает нуля), она создает объект отчета, содержащий стандартный отчет, описывающий окончательное состояние имитации. Вы можете видоизменить содержание стандартных отчетов, редактируя установки объекта моделирования. Промежуточные отчеты могут часто действительно использоваться во время отладки/испытания процесса. При взаимодействии с объектом имитации вы можете использовать команду ОТЧЕТ (REPORT), чтобы создать объект отчета, описывающий все внутренние GPSS- объекты. Вы можете извлечь любую информацию, которую желаете получить от имитации, путем использования потоков данных. Это сопоставляется с набором GPSS-блоков и PLUS-библиотеками (ОТКРЫТЬ (OPEN), ЗАКРЫТЬ(CLOSE), ПРОЧИТАТЬ(READ), НАПИСАТЬ(WRITE), ИСКАТЬ(SEEK)), что позволяет вам оставить метки и собрать полученную информацию от имитации и приписать ее к планируемым текстовым файлам. Такие метки могут быть обычно использованы для запуска фото-реалистичных анимаций в частности.

В пределах GPSS World, текстовые файлы, используемые в совокупности с потоками данных, представлены как текстовые объекты. Новый вызов PLUS-процедуры обеспечивает прямой онлайн доступ к внешним программам, таким, как, например, специальные аниматоры. После того, как модель пройдет тщательное испытание, настанет время определить эксперимент по определению количества данных и испытать их статистическое значение. GPSS World выделяется в этом. Его встроенный язык программирования, PLUS, поддерживает особый вид оператора PROCEDURE, названного оператором EXPERIMENT. В пределах эксперимента вы можете запустить и проанализировать комплекс экспериментов автоматически. Вы можете даже получить автоматический анализ наборов данных. Эти детали разработаны в образце Oneway.gps, и далее в уроках 19 и 20 справочника по GPSS World. Глава 13 этого справочника рассказывает об ANOVA и автоматических генераторах эксперимента (Automatic Experiment Generators) в подробностях.

Получение помощи

GPSS World поддерживает несколько вспомогательных средств помощи. Наиболее быстро помощь можно получить онлайн. Онлайн помощь доступна через GPSS World. [F1] - функциональная клавиша - вызовет панель помощи для активного окна или диалога. Также, если вы нажимаете клавишу [F1], когда пункт меню подсветится, контекстная панель помощи появится с информацией, относящейся к этому пункту меню. В пределах текстового поля объекта, выберите слово и нажмите [F1], чтобы переместиться к описанию ключевого слова в языковой справочной секции файла помощи GPSS World. И наконец, онлайн помощь может быть получена нажатием на кнопку помощи в некоторых диалоговых блоках.

Интернет предлагает больше помощи по GPSS World. HTML онлайн документацию вы найдете на сайте Minuteman Software, Minutemansoftware.com. Этот сайт предлагает множество других возможностей получения помощи. Вы можете войти на этот сайт следующим образом: выберите в главном меню **Файл/Интернет/Веб-страница GPSS**.

Minutemansoftware.com содержит другие полезные ресурсы и пути связи с другими пользователями и консультантами.

Student Version GPSS World может быть загружена с сайта бесплатно. Официальная документация GPSS World включает данное руководство, и справочник GPSS World. Они как доступны на сайте, так и имеются в печатной копии.

Не пропустите обширный набор образцов моделей. Большое количество профессионалов обнаружили, что один из наиболее быстрых путей к приобретению навыков имитации состоит в рассматривании и изучении работ других. Образцы моделей в каждой установке

GPSS World включают широкое разнообразие применений имитаций к проблемам реального мира.

Главное Меню

Большинство действий начинается в меню главного окна. Меню верхнего уровня: файл, правка, поиск, вид, окно, помощь, каждое из которых имеет субменю. Чтобы помочь направлять ваши действия, GPSS World блокирует несоответствующие меню команд. Заблокированные, меню команд бледные и не могут быть активированы. Когда имеется неоднозначность, касательно которой объект должен получить команду, вы должны будете выбрать одно из окон объекта мышью или клавиатурой для того, чтобы активировать соответствующее меню команды.

Давайте рассмотрим каждый пункт меню верхнего уровня.

Меню Файл

Левее всех располагается меню файл. Оно используется, чтобы открывать и сохранять объекты, печатать виды объектов, а также для доступа в Интернет для загрузки уведомлений и интернет-страниц, и для регистрации ПО для пользователей Commercial Version.

Меню Правка

Верхние пункты в меню «Правка» предназначены для работы с текстом. Все объекты GPSS World имеют текстовый вид, который может быть видоизменен при использовании команд этого меню. Команда «Отменить» это одноуровневое средство для удаления эффектов последнего действия в текстовой модели. Команды «Вырезать», «Копировать» и «Вставить» определяют переход к буферу обмена и от него. «Вставьте Строку» и «Удалите Строку» управляют расположением курсора в текстовом виде объекта. Команда «Шрифт» открывает диалоговое окно, позволяющее создать вид.

Оставшиеся пять пунктов позволяют вам модифицировать атрибуты объектов GPSS World. Окно выражений и окно диаграмм открывают диалоговое окно, которое позволяет вам изменить содержание существующего динамического окна выражений или окна диаграмм. Команда «Вставить Блок» открывает блок вводного меню, из которого вы можете открыть блок создания диалога. Эти диалоги доступны для направления создания GPSS-блока. Когда вы завершите блок создания диалога и кликните ОК, новый блочный оператор вставляется после пункта вставки в модели.

Блочные операторы создали этот путь, содержащий табуляции. Вы можете отрегулировать позиции табуляции в странице отчета установок модельного объекта. Команда «Вставить Эксперимент» открывает диалог, который позволяет вам автоматически создавать либо эффективный экранирующий эксперимент, либо сложный оптимизационный эксперимент и вставлять его в модельный объект. Посмотрите главу 13 этого справочника для получения информации об автоматических генераторах экспериментов.

Команда **Правка/Установки** позволяет вам рассмотреть и видоизменить настройки модели, имитацию, или объект отчета. Эти атрибуты обсуждаются детально.

Меню Поиск

Меню «Поиск» помогает вам передвигаться внутри текстового вида объекта.

Первый пункт **Найти/Переместить** открывает диалоговое окно, позволяющее вам переходить к следующему событию особого текста, и выборочно перемещать его. Перемещение всех пунктов возможно также.

Пункт «Переход» к строке переместит вас к строке необходимого номера. Это бывает полезно, когда сообщение об ошибке в имитации ссылается на номер строки.

Следующий "набор" пунктов – «Закладки». Они позволяют вам размещать невидимые закладки и метки, которые сохраняются с объектом. Они образуют циклический список, который вы можете увидеть через соответствующие команды меню или с помощью нажатия на необходимые клавиши. Команда меню «Следующая Закладка» переместит вас к позиции в тексте, соответствующей следующей закладке в списке. «Отменить метку»

удалит текущую закладку.

»Отменить все метки» - удалит все закладки. «Перейти к закладке» - переместит вас в место позиции текущей закладки с текущей позиции курсора.

Последние 2 пункта в меню «Поиск» работают с преобразователем сообщений об ошибке. Когда вы пытаетесь создать объект моделирования из модели, которая содержит синтаксические ошибки, циклический список сообщений об ошибках и размещениях создается в модельном объекте, заменяя все другие, которые были в нем до этого. Этот список становится частью модельного объекта и сохраняется с ним. Чтобы переместить список в каком-либо направлении используйте команды «Следующая ошибка» и «Предыдущая ошибка», чтобы отобразить сообщения об ошибках в строке состояния внизу главного окна. В то же время курсор позиционируется до ошибки в текстовом виде, так что вы можете корректировать его. Комбинации клавиш [Ctrl]+[Alt]+[N] и [Ctrl]+[Alt]+[P] позволяют вам просто управлять списком с клавиатуры.

Меню Вид

Оно включает в себя команды для контролирования того, что отображается в окнах.

Первый пункт, «Заметки», позволяет вам видеть заметки, касающиеся GPSS World. Вы можете обновить их с minutemansoftware.com, если пожелаете.

Используя второй пункт, «Панель инструментов» вы можете скрыть или отобразить панель инструментов главного окна, переключая первый пункт в панели инструментов.

Третий пункт, «Детали Объектов», позволяет вам изменить формат некоторой динамической имитации окна, чтобы показать больше или меньше деталей на экране.

Последний пункт меню, «Часы», переключается включателем в существующем имитационном объекте. Если этот переключатель включен, изменения в системных часах имитации покажутся в нижней правой части строки состояния, когда один из этих имитационных объектов выглядит как активное окно. Это простой путь, чтобы удостовериться, что моделирование действительно запущено, но он снижает скорость имитации.

Командное меню

Связано с созданием и управлением имитационным объектом. Команда «Создать симуляцию» запускает высокопроизводительный преобразователь, чтобы создать объект моделирования, основанный на модельном операторе в существующем модельном объекте, и это является вложенным файлом. Вы столкнетесь, как правило, с ошибками, которые должны быть исправлены (смотри Поиск/Следующая ошибка выше). Команда «Ретранслировать» доступна, чтобы запустить преобразователь снова, когда ошибки будут исправлены.

«Повторить последнюю команду» - простой путь, чтобы сделать какое-либо действие от того же объекта. Оставшиеся команды меню представляют собой формальные команды GPSS. Большинство из них "быстрые" клавиши для легкого пользования и большинству может быть задана функциональная клавиша. Эти команды не действуют, если текущий объект моделирования не готов для них.

Вы можете нажать на последний пункт меню «Общие» в командном субменю для того, чтобы написать свой собственный модельный оператор, не внесенный ни в какие списки.

Меню Окно

Первые два пункта его могут быть использованы, чтобы упорядочить все, когда много окон открыто в пользовательской площади главного окна. «Элемент изображения» покрывает пользовательскую площадь главного окна маленькими частями всех открытых окон, а «Каскад» скомплектует их в таком порядке, когда только все строки заголовка будут видны.

Следующие 2 пункта имеют свои субменю. Они используются, чтобы открыть окна и кадры в существующем имитационном объекте. Эти окна обсуждаются ниже. Они не действуют, пока текущий объект моделирования не готов для этого.

Меню *Помощь*

Меню «Помощь» может быть использовано, чтобы запустить систему помощи онлайн. Нижний пункт о GPSS World показывает версию ПО и пользовательское соглашение.

2.2.3 Установки

Установки - это набор значений, которые формируют цепочку выполняемых задач во всех объектах GPSS World. Они используются для контролирования имитации, возникновения отчетов, и содержимого окон и загрузочных функциональных клавиш.

Установки перенимаются. Когда вы создаете объект имитации, он перенимает исходные установки от своего модельного объекта. Подобным образом, отчетные объекты перенимают свои установки от имитационного объекта, который установил их. Отчет содержит только читаемые записи установок, которые действуют, когда запущено моделирование.

Текстовые объекты различны. Они всего лишь отображение простых текстовых файлов и не содержат копию установок. Это причина, по которой они - единственные объекты GPSS World, которые могут быть открыты внешними текстовыми редакторами. Другие объекты содержат установки, закладки, и в случае модельных объектов, списков данных, они сохраняются с объектами и мешают внешним текстовым процессорам. Если вам потребуется, вы можете переместиться из текста в текстовый вид объекта GPSS World, используя буфер обмена Windows.

Чтобы открыть окно установок, вы должны сначала открыть объект GPSS World. Затем выберете **Edit / Settings**, покажутся установки, и после нажатия на объект GPSS World они могут быть изменены. Ниже приведен пример окна установок для объекта с 5 закладками.

Страница Моделирования

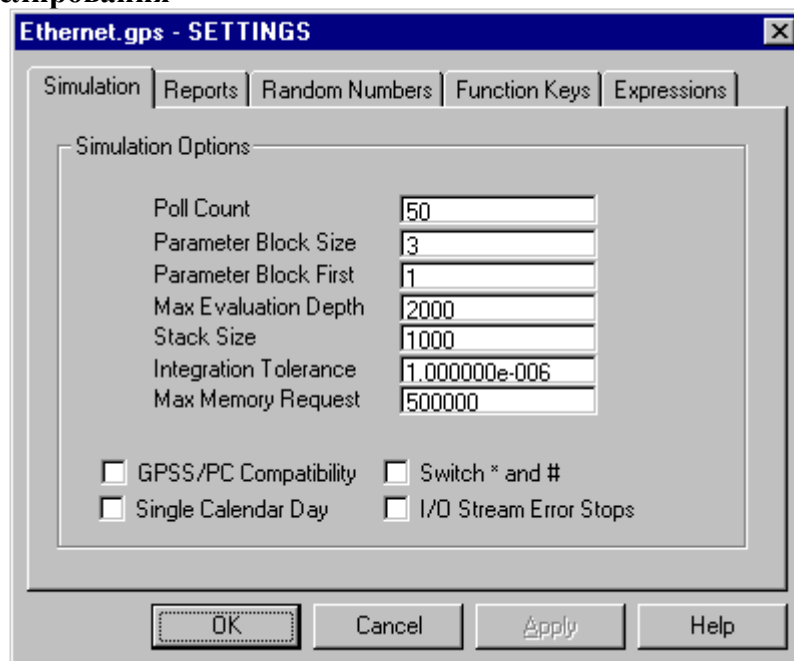


Рисунок 2.2 – Страница моделирования окна установок

Страница имитации включает несколько значений, которые вы можете изменить, что затронет ограничения и поведение работающей имитации.

Установка Poll Count - это число блоковых входов, которые могут предприниматься до проверки любого вида прерываний (пауз), таких как, например, команда от пользователя. Если вы используете большое количество операционных параметров GPSS в своей модели, вы можете немного увеличить скорость моделирования, перенося их в блок параметров (Parameter Block). Чтобы так сделать, номер параметра должен быть

заблокирован в поле «Parameter Block Size», и номер первого параметра должен быть заблокирован в поле «Parameter Block First». Если вы используете имена взамен чисел, чтобы ссылаться на эти параметры, вы должны контролировать значения, дающиеся имени параметра, используя операторы EQU.

Максимальная величина значений (Max Evaluation Depth) используется для защиты против циклических ссылок в наборе PLUS-процедур. Если ваше моделирование с большим количеством вложений без циклических ссылок, вы можете увеличить значения, чтобы вместить любой уровень вложенных вызовов процедуры.

Размер стека (Stack Size) используется, чтобы распределить стековое пространство для вложенных PLUS-процедур и библиотек. Вы можете сохранить пространство, уменьшая эти значения, но если у вас возникла ошибка вследствие переполнения стека во время моделирования, вы можете увеличить его.

Поле допуска режима интегрирования (Integration Tolerance) используется для дальнейшей фазы моделирования. Если вы ослабили этот допуск, сделав его больше, интегрирование будет быстрее, но может потерять точность.

Требование максимальной памяти (Max Memory Request) указывает лимит на охватываемую RAM-память для таких структур, как таблицы и матрицы. Такой лимит полезен для отслеживания ошибок, но вы можете повысить его, если необходимо. Если ваша RAM-память запрашивает больше, чем доступно на вашем компьютере, администратор виртуальной памяти вашей операционной системы увеличит эту дисковую функцию и, возможно, это дисковое пространство, используемое в различных целях.

4 чекбокса внизу страницы контролируют совместимость GPSS/PC, доступной только в Commercial Version, и показывают, каким образом обрабатываются потоки данных в моделировании.

Выберите чекбокс «GPSS/PC Compatibility», чтобы управлять имитацией в режиме совместимости GPSS/PC, как указано в разделе 1.5. Эта опция автоматически запустит опцию отдельный календарный день (Single Calendar Day), описанную далее.

Выберите чекбокс «Switch * and #», чтобы обусловить [*] чертами, используемыми для мультипликации, и [#] для не прямой адресации GPSS внутри приблизительно другого пути.

Выберите чекбокс «Single Calendar Day», чтобы обусловить имитацию для использования отдельного FEC-time сегментной таблицы внутри многочисленных объектов. В определенных имитациях это может быть немного быстрее, но не изменит результаты имитации.

Выберите чекбокс «I/O Stream Error Stops», если вы хотите, чтобы какие-либо ошибки, например, "File Not Found" (файл не обнаружен) и "End of File" (конец файла), выявленные в потоках данных в моделировании, остановили его.

Если этот бокс не выбран, вы можете обработать эти ошибки самостоятельно в моделировании. Смотрите раздел 4.16 для более подробной информации о потоках данных.

Страница отчета

Страница отчета связана исключительно с появлением отчета или диаграммы.

Стандартные отчетные опции групповых блоков включают в себя классифицированные подотчеты объектов GPSS, включенные в каждый стандартный отчет. Выберите их для внесения.

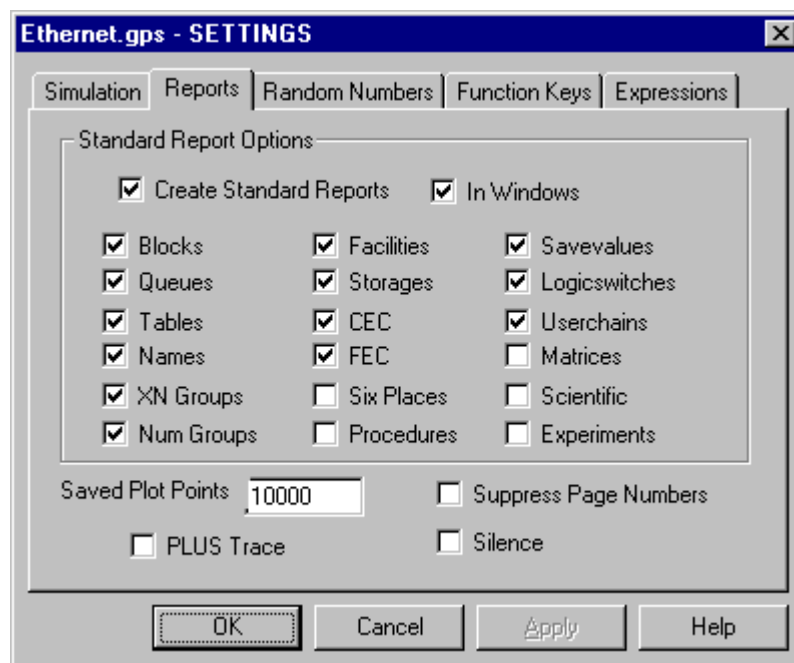


Рисунок 2.3 – Страница отчета окна установок

Выберите «Создать стандартный отчет» (Create Standard Reports) для возможности создания стандартного отчета автоматически, когда каждая имитация заканчивается. Команда «Отчет» (REPORT) все еще доступна, даже если она не выбрана.

Выберите чекбокс в Windows, чтобы открыть каждый отчетный объект в текстовом виде вместо каждого сохранения их в файле. Эти чекбоксы зависят от предыдущих открытых чекбоксов.

Выберите чекбокс «Six Places», чтобы установить точность действительный дробных чисел до 6 разрядов десятичной дроби взамен 3 в отчетах и сообщений о состоянии.

Выберите чекбокс «Scientific», чтобы использовать экспоненциальный способ задания функции предельных чисел в отчете и в потоке данных.

Ниже группового блока опций стандартных отчетов располагаются еще 4 установки.

Значения «Saved Plot Points» сообщают GPSS World, сколько пространства назначено циклическому буферу для последних точек диаграммы. Если вы прокрутите или напечатаете окно диаграммы, эти значения используются, чтобы восстановить диаграмму. Если вам недостаточно этого, левый сегмент прокрученной или напечатанной диаграммы может пропасть. Слишком большое значение может "засорить" виртуальную память.

Выберите чекбокс «Suppress Page Numbers», чтобы предохранить GPSS World от вставки номера страницы в заголовке, когда текстовый вид объекта печатается.

Выберите чекбокс «PLUS Trace», чтобы разблокировать введение регистра трассировочных меток для каждой PLUS-процедуры, вызванной выполнением имитационного объекта. Аргументы оцениваются до внесения их в регистровое окно. Если вы вставите вызовы в какую-нибудь простую процедуру и включите «PLUS Trace», вы получите трассировочную метку ваших переменных в регистровом окне.

Выберите чекбокс «Silence», если ваши ошибки взаимодействуют с объектами, чтобы записать их беззвучно, без звукового сигнала.

Страница случайных чисел

Страница случайных чисел позволяет вам контролировать внутренние потоки случайных чисел моделирования и внутреннюю нумерацию дочерних объектов.

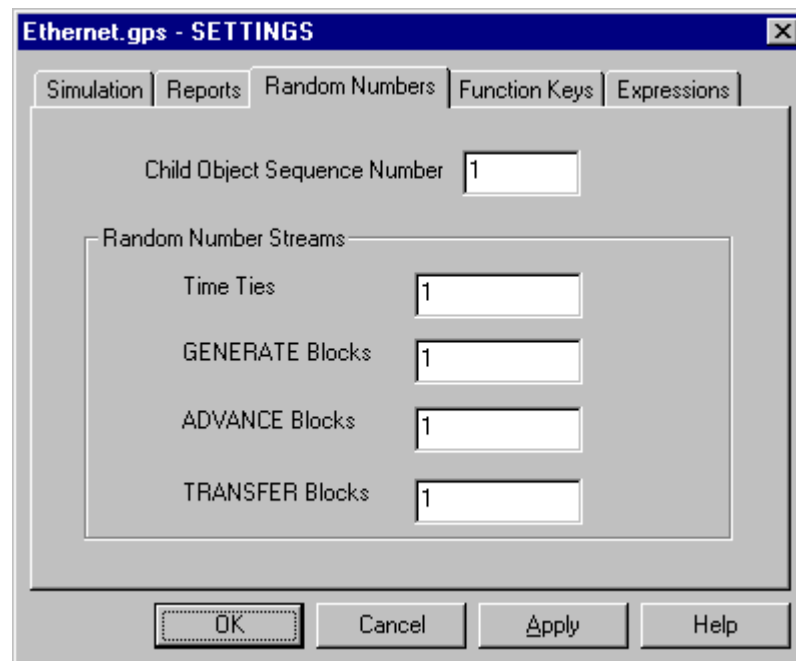


Рисунок 2.4 – Страница случайных чисел в окне установок

Чтобы сохранить уникальные имена объектов, GPSS World устанавливает последовательность и соответствие чисел именам новых созданных дочерних объектов. Чтобы изменить его, измените поле «Child Object Sequence Number».

Группа блоков потоков случайных чисел включает в себя остающиеся на странице установки. Они определяют, какой поток случайных чисел используется для цели разрешения временных связей (Time Ties), время перемешивания в блоках GENERATE и ADVANCE, и следующих блоков в блоке TRANSFER.

Поле ввода «Time Ties» позволяет вам устанавливать, когда ближайшее событие произойдет, и следующее событие может быть выбрано случайно. Чтобы это сделать, поместите номер объекта генератора случайных чисел в поле ввода «Time Ties». Если вы используете значение 0, временные связи не будут случайны. Приблизительное значение одной части из миллиарда используется в эквивалентном критерии действительных временных значений.

Поле ввода «GENERATE Blocks» позволяет вам установить, какой генератор случайных чисел должен использоваться, в то время как «GENERATE Block» подсчитывает время между операциями А и Б. Если вы устанавливаете неположительное число, генератор случайных чисел использует 1.

Поле ввода «ADVANCE Blocks» позволяет устанавливать какой генератор случайных чисел используется, в то время как «ADVANCE Block» подсчитывает время выдержки от операций А и Б. Если вы устанавливаете неположительное число, генератор случайных чисел использует 1.

Поле ввода «TRANSFER Blocks» позволяет устанавливать, какой генератор случайных чисел используется, в то время как «TRANSFER Block» выбирает вероятное назначение блока. Если вы устанавливаете неположительное число, генератор случайных чисел использует 1.

Страница функциональных клавиш

Вы можете ввести операторы GPSS в поля на этой странице, чтобы загрузить их в функциональную клавишу, и/или вы можете изменить позиции табуляции в модельном объекте.

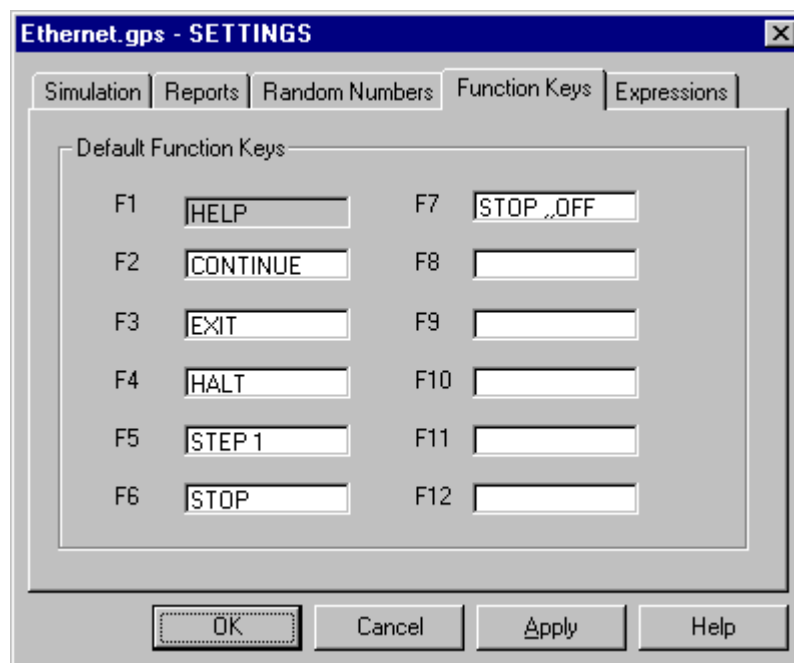


Рисунок 2.5 – Страница функциональных клавиш окна установок

Когда объект моделирования будет выбран, все, что вам потребуется сделать, чтобы послать к загрузке оператор GPSS - это нажать функциональную клавишу. Использование оператора INCLUDE позволяет вам отправить список операторов любой сложности, включая PLUS-процедуры и экспериментальные определения.

Секция «модельные позиции табуляции» (The Model Tabstops) позволяет вам установить табуляцию модельных объектов. Это желательно, когда вы используете блок создания диалогов (Block Creation Dialogs), который использует табуляции в блоках операторов, которые он создает.

Позиции табуляции блокируются для имитационных или отчетных объектов. Они должны быть ненулевыми и точно увеличенными от первой (крайней левой) до последней границы.

Страница выражений

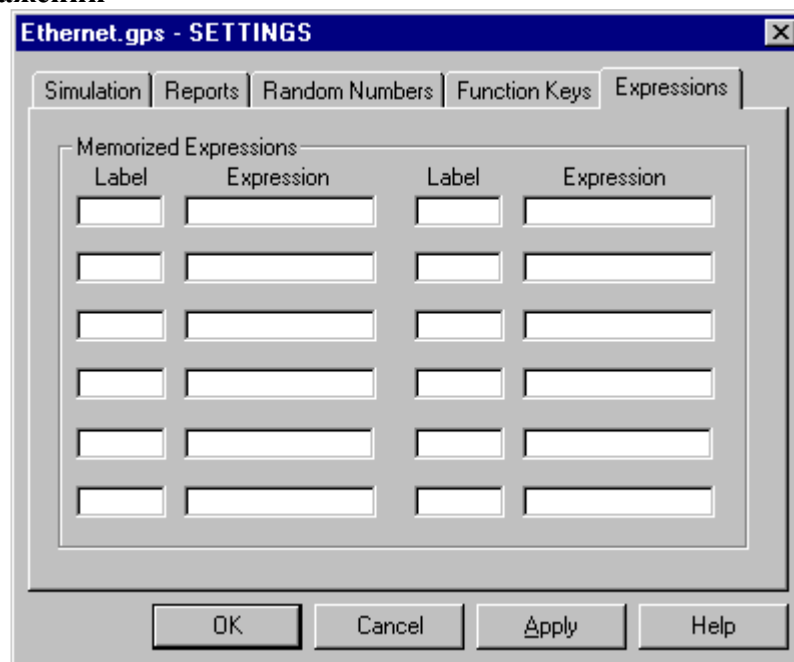


Рисунок. 2.6 – Страница выражений в окне установок

Когда вы открываете окно выражений или окно диаграмм в имитационном объекте, вы должны установить выражение, которое должно быть просмотрено, также как и название, которое используется как ярлык. Эта страница позволяет вам сохранить комплекс выражений с имитационным объектом. Затем, когда вы откроете окно, использующие выражения, вместо того, чтобы переписывать их.

Как окно выражений, так и окно диаграмм дают вам шанс сохранить выражения непосредственно в рабочем окне. Вы можете сделать это следующим образом: выберите команду меню **Edit / Expression Window** и **Edit / Plot Window**. И затем, выбрав выражение, нажмите «**Memorize**».

Вы можете записать ярлык и выражение в окно настроек. Эффект такой же. Вы будете способны перенести обратно помеченные выражения легко, без выделения.

2.3 Контролирование сеанса

Нормальная установка GPSS World помещает строку в меню **Пуск/Программы** вашей операционной системы, которую вы нажимаете для запуска программы. Вы можете создать ярлык для GPSS World и разместить его на рабочем столе, или в любом другом месте по вашему желанию. Простое перетаскивание иконки GPSS World из меню Пуск. Вы можете открыть объект в сеансе GPSS World просто двойным щелчком на нем. Когда вы запускаете GPSS World в первый раз, программа регистрирует файловые типы и расширения GPSS World с операционной системой Windows.

Это позволяет Windows ассоциировать GPSS World с любым из его объектных файлов. Впоследствии вы можете, путем двойного нажатия на любой файловой иконке открыть сеанс GPSS World.

2.3.1 Использование главного окна

Когда вы начали сеанс, вы можете контролировать его, используя мышь и клавиатуру. Вы можете выбрать в меню пункты с помощью мыши, используя ускоряющие клавиши, указанные в меню. Многие взаимосвязи требуют от вас записать информацию, используя клавиатуру, и установить функциональные клавиши на вашей клавиатуре, которые могут быть загружены с любой командой, которую вы выберете. Как было описано ранее, вы редактируете настройки объекта для того, чтобы загрузить команды на функциональные клавиши.

Перед тем как начать, вы должны познакомиться с главным меню. Быстрый обзор представлен в разделе 2.2.2.

Большинство пунктов главного меню заблокированы и появляются серыми, до того как они могут быть использованы.

Поскольку вы открываете и создаете объект GPSS World, рабочее командное меню будет заблокировано, как следует.

Использование главного меню

Хотя все объекты GPSS World имеют маленькие меню в своих окнах, большинство действий начинается в главном меню. Все меню используются однообразно. Некоторые меню согласовывают данные в создании диалогового окна, содержащего некоторые контроллеры, таких как кнопки и поля ввода. Поля ввода обычно требуют от вас использования клавиатуры, чтобы записать дополнительную информацию.

Чтобы запустить действия меню, у вас есть три способа. Первый: вы можете просто использовать мышку для нажатия на требуемый пункт меню. Второй: вы можете нажать клавишу Alt, чтобы активировать меню, и затем нажать быструю кнопку в каждом пункте меню. Третий: если пункт меню нижнего уровня имеет быструю клавишу, показанную с права от имени пункта, вы можете просто нажать эту комбинацию клавиш, чтобы начать действие, если она заблокирована.

Быстрые клавиши

Быстрые клавиши - это комбинации клавиш, которые мгновенно вводят команды меню в GPSS World. Обычно они указаны справа от названия пунктов меню. Они заблокированы, только когда они выбраны в меню. Чтобы активировать быструю клавишу, удерживайте все клавиши в комбинации, исключая последние, затем нажмите последнюю клавишу, затем отпустите все клавиши.

Быстрые клавиши - это быстрый путь для того, чтобы отправить GPSS World прямо к имитационному объекту. Примеры:

[Ctrl]+[Alt]+[S] - создать имитацию
[Ctrl]+[Alt]+[R] - ретранслировать
[Ctrl]+[Alt]+[L] - повторить последнюю команду
[Ctrl]+[Alt]+[C] - продолжить
[Ctrl]+[Alt]+[H] - остановить
[Ctrl]+[Alt]+[1] – шаг 1

Управление текстовым видом

Клавиши управления текстовым видом позволяют вам переходить к специальным разделам текстового вида

Пример:

[Ctrl]+[Alt]+[B] – Следующая закладка
[Ctrl]+[Alt]+[G] – Переход к строке
[Ctrl]+[Alt]+[N] – Следующая ошибка
[Ctrl]+[Alt]+[P] – Предыдущая ошибка
[Ctrl]+[Alt]+[F] – Найти переместить

Ускорители правки текста

Редактирование это быстрое использование ускорителей правки текста вместо выбора меню.

Пример:

[Ctrl]+[Z] – Вставьте строку
[Ctrl]+[D] – Удалите строку
[Ctrl]+[I] - Отменить
[Ctrl]+[C] – Копировать в буфер обмена
[Ctrl]+[X] – Вырезать в буфер
[Ctrl]+[V] – Вставить из буфера
Также,
[Ctrl]+[Ins] - Копировать в буфер
[Shift]+[Del] - Вырезать в буфер
[Shift]+[Ins] - Вставить из буфера

Основные ускорители

Есть некоторые другие ускорители.

Пример:

[Ctrl]+[O] – Открыть файл
[Ctrl]+[S] – Сохранить файл
[Ctrl]+[P] – Распечатать файл
[Alt]+[F6] – Следующее окно
[Shift]+[F6] - Предыдущее окно

Помощь онлайн

Клавиша [F1] открывает службу справки и поддержки. Вы можете рассматривать содержимое этой помощи в справочном руководстве GPSS World . Если пункт меню выбран, эта клавиша откроет контекстную «чувствительную» помощь в этом пункте. Или, если курсор вставлен в ключевое слово GPSS World, такое как например «Command Name» в текстовом виде, если вы нажмёте [F1] вы откроете справочное руководство онлайн по информации об этом.

Сообщения об ошибках

Когда операция не может быть завершена программа показывает сообщение об ошибке в диалоговом или текстовом окне. Большинство сообщений предназначены для самообъяснения. Исходя из этого, дальнейшее объяснение каждого сообщения может быть обнаружено в разделе 14 этого руководства.

Использование текстового окна

Текстовое окно всех объектов имеет свои возможности правки текста. Вы можете вводить текст в текстовое окно, открывая файл, печатая или вставляя из буфера обмена. Текст может быть изменён различными способами.

Существуют два выделительных индикатора, которые вы можете использовать, когда текстовый поле ввода активно. В текстовом поле курсор выглядит, как I-черта. Курсор - это мигающая вертикальная линия в текстовом окне, которая отмечает место ввода текста.

Доступные правочные операции:

Переместите курсор в позицию указателя мышки нажимая кнопку мыши 1.

Переместите курсор с помощью стрелочки на клавиатуре.

Переместите курсор и прокрутке с помощью клавиш [PgUp] [PgDn].

Переместите курсор к началу и концу текста, нажимая [Home] и [End] .

Прокрутите, используя панель прокруток в текстовом окне.

Перейдите на следующую строку с помощью [Enter] .

Удалите выбранный текст с помощью [Del] .

Отмените последнюю правку с помощью Edit / Undo .

Удалите строку с курсором Edit / Delete Line.

Вставьте новую строку после курсора с помощью Edit / Insert Line .

Операции буфера обмена включают:

Вырезка текста, после чего удаленный текст размещается в буфере обмена. Вы также можете использовать комбинацию клавиш [Ctrl] + [X].

Копирование выделенного текста, после чего он размещается в буфере обмена.

Комбинация клавиш [Ctrl] + [C].

Вставка текста из буфера обмена. Комбинация клавиш [Ctrl] + [V].

Другие операции, использующие быстрые клавиши, описывались ранее.

Вы можете поменять шрифт текста. Сначала выделите текст, затем выберите **Edit / Font**. Откроется диалоговое окно правки текста, где вы можете установить тип шрифта, размер и вид для выбранного текста. GPSS World использует по умолчанию шрифт Courier. Он имеет преимущество, так как является моноширинным с определенной шириной. Еще одно преимущество состоит в том, что в операторах сохраняется выравнивание.

2.3.2 Построение моделей

Физический процесс построения модели начинается с создания модельного объекта, содержащего модельные операторы, написанные вами. Модельные операторы описаны в главе 6, 7, 8 этого руководства. Часто, можно просто начать построение моделирования с изменения существующей модели. Папка Samples - богатый ресурс маленьких модельных объектов. Модельные операторы могут существовать, как текстовые объекты (.txt), так же

как и модельные объекты (.gps), позволяющие вам совместно использовать коды через проекты. Хотя только один модельный объект может быть в имитации, здесь также могут присутствовать множество текстовых объектов. Они связываются с модельным объектом через оператор **INCLUDE**, который рассматривает имя файла текстового объекта как аргумент.

Развитие модели начинается с открытия текстового вида модельного объекта. Полный набор операций текстовой правки доступен для изменения и добавления модельного оператора. Если вы предпочтете, использование блока ввода меню и одного или более блока создания диалогового окна, то можете быстрее установить блоковые операторы в вашем модельном объекте. Онлайн помощь поможет вам избежать всех ошибок. Модель должна быть транслирована в соответствии с созданием моделируемого объекта. Трансляция начинается через меню **Command/Create Simulation** в окне модели. Транслятор определяет, есть ли какие-нибудь синтаксические ошибки в модели. Если да, создается циклический список сообщений об ошибке, и затем ошибки могут быть обнаружены и исправлены быстрее. Если ошибок нет, результирующее моделирование автоматически отправит ее к объекту моделирования, перемещая остальные, которые уже существуют. Когда объект моделирования принимает имитацию, он немедленно разрешает все вставленные команды в порядке, указанном транслятором.

Команды INCLUDE

Вы можете использовать эти команды где угодно в модели, исключая PLUS-процедуру. Обычно, они используются в модельном объекте, чтобы включать в себя операторы одного или нескольких текстовых объектов. Оператор **INCLUDE** может включать до 5 вложений, но он не может быть использован внутри PLUS-процедуры.

Синтаксис этой команды: **INCLUDE "FileSpec"**.

В месте, где вы располагаете слово, написанное курсивом FileSpec со своей файловой спецификацией, включая расширение (normally .txt), ставьте двойные кавычки. Директория по умолчанию такая же, как и у модельного объекта. Команды **INCLUDE** имеют и другие применения. Вы можете записать команду **INCLUDE** как общую интерактивную команду. Это обусловит тот факт, что модельный оператор должен быть транслирован и отправлен к объекту моделирования. Команда **INCLUDE** может быть задана и функциональной клавишей, как и другие команды. Поэтому отдельные клавиши могут переопределять комплекс командных файлов или PLUS-процедур.

До использования файла в команде **INCLUDE**, вы должны протранслировать его, чтобы удалить ошибки. Для этого скопируйте его модельный оператор в буфер обмена, откройте новый шаблон модельного объекта, вставьте в него из буфера обмена скопированное и протранслируйте это используя меню **Command / Create Simulation**. Это позволит вам исправить все синтаксические ошибки быстро до комбинирования текстового объекта в более совершенную модель.

Транслированные модели

Объекты моделирования начинаются как транслированные модельные объекты. Когда вы транслируете модель, выбирая **Command/Create Simulation**, транслятор программы проверяет синтаксис, создает имитацию и отправляет ее в новый созданный модельный объект. Если все успешно, транслирование активирует все интерактивные команды и имитации, так что состояние новой имитации может быть контролируемо. После этого вы можете отправить модельный оператор в существующую имитацию для обработки данных.

Программа включает в себя новый высокопроизводительный модельный транслятор, который быстрее предшествующего в сотни раз. Огромные модели транслируются не более, чем в считанные секунды. Когда транслятор обнаруживает одну или более ошибок в модели, он создает список сообщений об ошибках, который может быть доступен из текстового вида модельного объекта. Вы можете быстро проследить ошибку, используя **Search / Next Error** или **Search / Previous Error**. Обе имеют быстрые клавиши. Каждая

остановка в списке размещает сообщение об ошибке в строке состояния главного окна и делает вставку в текстовом виде модели до того синтаксического элемента, который обусловил ошибку. Затем несложно исправить каждую и перейти к тексту.

Если ошибка возникла в текстовом объекте INCLUDE, и вы не можете обнаружить ее быстро вы можете вставить этот оператор в новый шаблон модельного объекта и протранслировать его самого. Лучше всего размещать только проверенные операторы в файл Include. Блоки размещаются в имитации в таком же порядке, что и в трансляторе. Команды, накопленные в списке, отправляются к имитационному объекту. Список сканируется для неотложных команд (SHOW, HALT, и CONDUCT), и остатки размещаются в командном списке моделирования и представляются один за другим. Командный список интерактивно отправляется к существующей имитации, интерпретированной по такому же принципу.

Динамическое окно моделирования (не кадр) автоматически обновляется, когда вы транслируете модель, чтобы убедиться, что вид текущий. Если объект не может долгое время быть обнаружен, окно будет автоматически закрыто.

Контролирование моделирования

Команды программы используются для контроля работы имитации. Они обсуждаются подробно в главе 6. Они могут быть вставлены в модель или могут быть отправлены как интерактивные операторы.

Любой модельный оператор может быть отправлен в существующую имитацию. Такие операторы называются интерактивными. Вы можете контролировать и видеть работающую имитацию сжато с операциями, которые доступны. Для этого используйте командное меню в главном окне. Если нужной вам команды нет в списке, выберите **Command/Custom** и запишите её в поле ввода, и нажмите ОК. Это действие протранслирует команду и отправит ее в моделирование. Командное меню заблокировано, пока есть имитация, принимающая взаимосвязи.

Вы можете использовать определения блочных операторов и PLUS-процедур в интерактивных командах. Блочные операторы отправляются интерактивно в существующую симуляцию, так как шаблонный блок должен быть создан и немедленно уничтожен после активного входного сообщения, изменяющего файл, пытающегося ввести его. Этот режим называется ручной имитацией. Процедурные операторы отправляются к существующим имитационным объектам, регистрирующим пользователя, создавшего PLUS-процедуру, переименовывая те названия, которые уже существуют.

Взаимосвязи зарегистрированы в системном журнале имитационного объекта.

Работа

Обычно, команда START используется для начала имитации. Она может быть введена интерактивно или может быть частью модельного объекта. Оператор START устанавливает окончное устройство подсчета, которое является важной составляющей переменных имитации. Оконечное устройство подсчета должно быть уменьшено до нуля для имитации к концу. Любой оператор TERMINATE в имитации может уменьшать такое значение.

Когда объект моделирования выполнит команду START, входное сообщение начнет повторяться в имитации. Обычно моделирование завершается само собой и производит стандартный отчет, когда окончное устройство подсчета достигнет 0. Если ошибка остановит это, отчет не составит. Отчетные опции могут быть изменены в настройках (Settings) имитационного объекта. Они описываются в разделе 2.2.3.

Многочисленные имитации могут работать автоматически с использованием RESET, CLEAR и START в последовательности команд. Они обычно используются в PLUS-эксперименте, который может контролировать и анализировать экстенсивные настройки работы имитации. PLUS-эксперименты рассмотрены в этом руководстве в разделе 8.2.4. и в уроке 19 учебного пособия по GPSS World.

Остановка имитации

Вы можете остановить работающую имитацию несколькими путями.

Первый: вы можете прервать имитацию посылая команду остановки HALT. HALT может быть представлена быстрыми клавишами [Ctrl]+[Alt]+[H], функциональными клавишами, интерактивными командами из командного меню, или частью командного файла.

Второй: вы можете ввести одно или более остановочных условий с помощью команды STOP до того, как начнете или продолжите имитацию. Команда STOP устанавливает условия, которые, будучи обнаружены, становятся причиной для остановки имитации. Когда она останавливается, предыдущие условия остановки не удаляются. Если вы продолжаете имитацию после того, как условия остановки были обнаружены, начальное условие, ставшее причиной остановки имитации, не позволяет остановленному входному сообщению продолжаться. Условия STOP могут быть проконтролированы операциями "Point and Shoot" на панели инструментов отладки, используя только мышь. Это рассматривается ниже, в главе 5.

Третий: вы можете возобновить имитацию с помощью команды STEP. После определения числа попыток ввода блоков, моделирование остановится автоматически.

Четвертый: когда состояние ошибки будет обнаружено, работа имитации прекратится. Сообщения об этом будут отправлены для установления причин.

Пятый: вы можете просто подождать, пока моделирование не завершится. Это случится, когда окончное устройство подсчета достигнет 0.

В каждом из этих случаев при остановке имитации, ее состояние заносится в регистр имитационного объекта.

Интерактивные операторы

Любой модельный оператор может быть отправлен в существующую имитацию. Такие операторы называются интерактивными. Они могут быть командами, PLUS-процедурными описаниями или блочными операторами.

Каждая моделирование имеет командный список, связанный с ней. Объект моделирования представляет каждую команду в списке одну за другой, пока они не были остановлены или пока они не завершили свою работу иначе. Команды в модели, другие, нежели HALT и SHOW, размещаются в командном списке до того, как они осуществляются.

Интерактивные команды являются неотложными и очередными. Неотложные команды, такие как HALT и SHOW, осуществляются, как только объект моделирования принимает их. Все другие команды очередны. Они располагаются в конце списка команд, который все еще не завершается.

Команда HALT - это особый случай. Это не единственная неотложная команда, но она также уничтожает любые остающиеся очередные команды, находящиеся в командном списке. После осуществления команды HALT, объект моделирования больше ничего не может сделать от имени этой имитации до момента, пока командный список станет пустым.

Команды являются не только операторами, которые должны использоваться интерактивно. Когда оператор PLUS послан интерактивно, это заставляет данную процедуру зарегистрироваться внутри имитации. Затем процедура может быть запущена из любого PLUS-выражения. Если процедура с таким же именем уже существует внутри имитации, она уже предопределена.

PLUS-эксперимент является особым видом процесса. PLUS-эксперименты могут быть отправлены также интерактивно. Если любая названная PLUS-процедура (или эксперимент) уже зарегистрирована с объектом моделирования, она заменяется. Затем эксперимент может быть определен с помощью команды CONDUCT, которая обсуждалась в главе 6.

Интерактивные блочные операторы называются ручными моделирующими операторами. Когда они принимаются объектом моделирования, они вызывают создание временного

блока, и попытку появления внутри входного сообщения. Затем блокировка уничтожается.

В этом виде GPSS блокировки могут быть использованы как команды, способствующие лучшему уровню контроля моделирования. Данные «Data Stream I/O» (потoki данных ввода-вывода) поддерживаются интерактивно. Потoki данных обсуждались в разделе 4.16.

Блоки генерации единственные, которые не могут быть использованы в ручном моделирующем режиме. Вместо них операторы «SPLIT Block» могут быть использованы интерактивно для создания входящих сообщений.

Есть несколько способов послать моделированное сообщение в моделирование. Вы можете использовать командное субменю главного окна, вы можете использовать быстрые клавиши, и вы можете ассоциировать GPSS оператор с функциональными клавишами в окне настроек модели «Model Settings Notebook».

Чтобы послать список команд в моделированный объект выберите **COMMAND / CUSTOM**, наберите список команд и выберите **ОК**. Это переведет список команд и отправит его на моделирование.

Очень удобно поместить список часто используемых команд в маленький текстовый файл. Затем вы можете использовать команду **ВКЛЮЧИТЬ** чтобы отправить всю последовательность к моделируемому объекту. Даже легче, вы можете загрузить функциональную клавишу с командой **ВКЛЮЧИТЬ**, и получить весь командный список подготовленный для единственного нажатия клавиши.

Предопределенные объекты

Существует 15 разных типов, которые вы можете создать. Предопределенные объекты - это входящие сообщения, блоки, средства, функции GPSS, Logicswitches, матрицы, очереди, памяти, savevalues, таблицы, пользовательские цепи, вариации, нумерационные группы, входящие сообщения, и генераторы случайных чисел. Чем лучше вы понимаете эти модели и их свойства, тем более мощными будут ваши моделирования. Глава 4 посвящена этим модулям.

За исключение входных сообщений, модели никогда не стираются с моделирования. Хотя, некоторые модели могут быть определены интерактивно. Вы можете использовать **STORAGE**, **TABLE**, **QTABLE**, **MATRIX** или команды **VARIABLE**. Похоже, и **PLUS** процессы могут быть предопределены интерактивно.

Локация - это имя, которое используется в блоковом операторе. Эта область не востребована в блоковом операторе. В отличие от др. модульных типов, блоки могут не быть определены в GPSS. Похожая гибкость может быть достигнута путем использования боков **EXECUTE** и определения его операнда **A**.

Просмотр

Широкое разнообразие окон доступно для просмотра моделирования. Многие окна одних типов могут быть открытыми на разных моделированиях. Все окна могут быть напечатаны, или сохранены для дальнейшего использования.

Симуляционные окна делят на 2 категории: «Snapshots» и «динамические окна». Все «Snapshots» и окна открываются, используя субменю **Windows** в главном меню. Исп. **Window/Simulation Snapshot** для «Snapshot» и **Window/Simulation Window** для динамического окна. Меню отключены, пока видны моделирования.

«Snapshots» не обновляются online. Они берут одну картинку, представляющую собой вид моделирования в данный момент времени.

Вы можете использовать Snapshots для следующих предметов:

- Остановка текущих состояний
- Любые вход. сообщения в имитации
- Текущие цепочки событий
- Будущие цепочки событий

- нумерационные группы
- Цепи пользователей
- Группы вход. сообщений.

Динамические окна обновляются как имитационные изменения. Они обновляются, когда вы переводите модель, или вызываете событие RESET. Вы можете открыть просмотр online:

- Блочную структуру модели
- Произвольный список выражений
- средства объектов

Динамические окна составляют нагрузку на систему, так как большое количество сообщений, полученных от моделирования, должно быть обработано. Мы рекомендуем использовать малое количество окон, когда моделирование должно работать. Намного легче открыть большое количество кадровых окон, когда моделирование остановится.

Распечатка окон

Любое имитационное окно может быть распечатано при выборе из главного меню **File / Print**.

Тестирование

После того, как вы создали одну или более моделей, вы можете начать тестировать и отслеживать ваше моделирование. GPSS World обеспечивает вас большим количеством тестирующего оборудования, включая визуальные вспомогательные средства. Высокий уровень согласованности действий, обеспеченный программой, позволяет вам управлять моделированиями с целью определения проблемы.

Во-первых, вы должны проконтролировать, что моделирование ведет себя, как вы и предполагали. Графическое окно программы идеально для этой цели. Мы надеемся, что вы начнете моделирование и затем начнете исследовать общее поведение вашей модели, поскольку они должны быть устойчивыми.

Одно из наиболее полезных окон для отладки - это регистровое окно объекта моделирования. Когда вы обуславливаете индикаторы трассировочных меток входных сообщений включенными, все блочные входы этих входных сообщений отображают сообщение о трассировочных метках в регистре. Вы можете разместить блок TRACE внутри модели для этой цели, или вы можете увидеть оператор TRACE в режиме ручного моделирования, чтобы проконтролировать индикатор трассировочных меток отдельных входных сообщений. Настройка PLUS Trace сейчас может быть использована для поддержки развития PLUS-процедур. Это рассматривается ниже.

Для серьезной отладки вам необходимо разместить сигнал «стоп» в определенных местах моделирования, а затем обследовать состояние моделирования, открывая динамические окна. Кадровые окна CEC и FEC вкупе с командой STEP, часто достаточны для определения причины, по которой произошли определенные случаи. FEC и CEC могут быть видны в кадровых окнах или показаны с помощью команды REPORT.

Во всех SNA доступно использование команды SHOW. Они могут быть динамически отображены в окне выражений графического окна. Иногда весьма полезно делать поправки во время процесса отладки. GPSS World позволяет вам переопределять сущности, переопределять процедуры PLUS, изменять значения имён заданных пользователем, и управлять процессом моделирования посредством выражений управления моделированием. Затем вы можете продолжать имитацию далее без перезапуска.

Ошибочные остановки

Ошибочные остановки, это ошибочные условия, которые вынуждают утверждения работать неправильно внутри имитации. Вы должны исправить все ошибки, до того как начнёте процесс моделирования.

Панель инструментов отладчика

Большинство динамических окон имеют дополнительную панель, которая называется панель инструментов отладчика и позволяющая вам выполнять процесс отладки, используя лишь мышь.

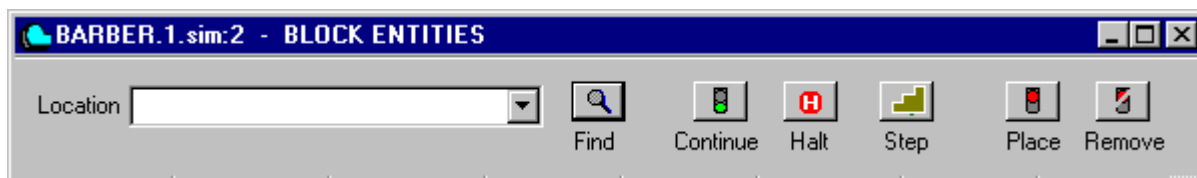


Рисунок 2.7 – Панель отладчика окна блоков

Окно блоков имеет улучшенную панель отладчика, потому что вы можете выбирать иконку блока с помощью мышки, затем нажимать на кнопку «Place» для размещения условия остановки, которое остановит симуляцию, когда активная транзакция войдёт в этот блок. Когда блок выбран, и вы нажимаете на кнопку «Remove», то условие остановки удаляется.

Множество других окон имеют панели без кнопок «Place» и «Remove». Используйте кнопку «Halt» для того, что бы послать команду HALT объекту моделирования. Кнопка «Продолжить» продолжает имитацию, а кнопка «Step» устанавливает в конце каждого блока команду HALT. Вы можете увидеть набор условий остановки на снимке пользовательских остановок. Выберите **Window / Simulation Snapshots / User Stops**

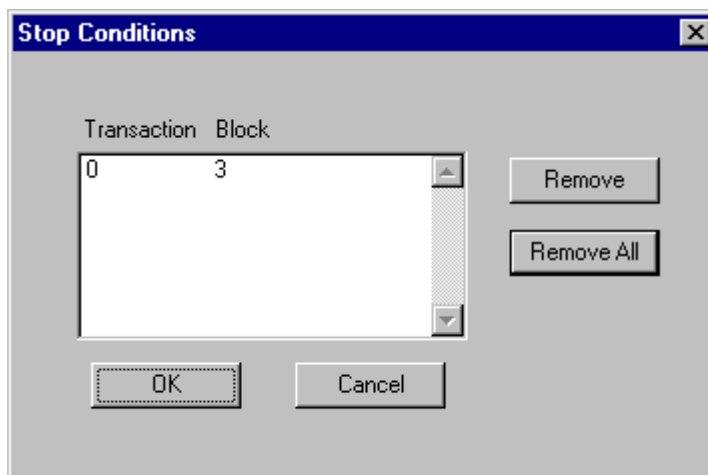


Рисунок 2.8 – Снимок пользовательских остановок

Снимок пользовательских остановок показывает набор всех активных условий остановки. Условия остановки определены номером транзакции и номером блока. Любая попытка транзакции с заданным номером войти в блок с заданным номером вызовет команду HALT. Значение 0 особенное. Оно означает, что условие остановки применяется ко всем транзакция или блокам, имеющим значение 0. Например, на рисунке показано, что если любая транзакция попытается войти в блок 3 то произойдет остановка. Условия остановки добавляются к списку использования команд STOP либо через панель отладчика. Вы можете удалить любое или все условия остановки с помощью команды STOP OFF или через панель отладчика, или нажав кнопку «Remove» или «Remove All» на снимке пользовательских остановок.

Отслеживание PLUS

Когда вы разрабатываете модель объектов, которая включает ваши собственные процедуры PLUS, вам следует рассмотреть временное включение отслеживания PLUS на второй странице параметров записной книжки (Notebook). Используйте меню **Edit / Settings** в главном окне, для того чтобы сделать это. Далее, вызовы процедур PLUS будут записываться посредством записи трассировочных сообщений в окно журнала моделирования (Simulation's Journal). Вы можете найти это полезным для того, чтобы определить процедуру PLUS, которая берёт один или более аргументов, но ничего больше не делает. Далее, вы можете вставить вызовы в ваш PLUS-код, которые обеспечат отслеживание пользовательских переменных и их отображение в окне журнала. Отслеживание может быть легко удалено, после того, как ваша PLUS процедура будет полностью протестирована.

Экспериментирование

После того, как вы установили правдоподобие в вашей GPSS модели, вы обычно начинаете экспериментальную фазу проекта. Во время этого периода, вы неоднократно запускаете вашу модель для того, что бы поэкспериментировать с эффектами от предложенных действий в вашей моделируемой системе. Это позволит вам установить эффекты, наблюдаемые вами в вашей имитации, которые выше уровня статического шума. Анализ изменений, обеспечиваемый библиотекой процедур ANOVA, может дать вам уверенность в том, что ваши результаты больше чем результат случайной изменчивости. Мощность экспериментов PLUS позволит вам запускать комплекс автоматических экспериментов, каждое исследование экспериментального отклика выявляет и собирает данные для продвинутой статистической обработки. Автоматические генераторы экспериментов (Automatic Experiment Generators) GPSS в вашем полном распоряжении. Прочитайте главу 13 этого руководства и изучите уроки 19 и 20 в руководстве The GPSS World для того, чтобы начать ваше экспериментирование.

Пакетный режим

Есть фоновый пакетный режим действий, который полезен в том случае, когда требуется большое количество вычислений, и который не требует действий с вашей стороны. GPSS World Module Directory это либо конкретная директория DOS, либо запись в переменной PATH. В командной строке DOS **GPSSW FileSpec BATCH, FileSpec** имя модели объекта или объекта моделирования. Если объект расположен в другой папке, нежели GPSS World модуль, то вы должны указать полный путь к этому объекту. В пакетном режиме GPSS World открывает объект и пробует продолжить обработку в минимизированном окне. Если объект это модель, то GPSS World автоматически запускает меню создания имитации для её трансляции. Если объект это имитация, то GPSS World пересылает ей команду CONTINUE. Для завершения сессии и сохранения всех файлов вам необходимо ввести команду EXIT 1 или Exit(1). Вам не следует использовать пакетный режим, до тех пор пока ваша модель не протестирована. Любые синтаксические ошибки или остановки вызванные ошибками остановят сессию. Вы будете иметь с ними дело, открыв окно и взаимодействуя с сессией.

2.3.3 Искусственные пределы

Пределы запроса памяти

Условный предел памяти используется для того, чтобы защитить вашу имитацию от большого увеличения запроса памяти. Он отслеживает нарушение условия, когда запрос памяти превышает предел. Он применяется относительно таблиц, матриц и графических окон. Вы можете изменить предел посредством изменения предела на странице имитации (Simulate Page) в записной книжке настроек модели (Model Settings Notebook). Выберите **Edit / Settings** и перейдите на Simulate Page введите новый предел в строке **Max Memory Request** затем нажмите ОК.

Размер стека

Имитационный стек используется для размещения процедур вызова. Он может вызвать ошибку, если недостаточно велик, в том случае, когда ваша имитация вызывает процедуру PLUS на глубоком уровне вложенности. Вы можете изменить предел посредством изменения предела на странице имитации (Simulate Page) в записной книжке настроек модели (Model Settings Notebook). Выберите **Edit / Settings** и перейдите на Simulate Page введите новый предел в строке **Stack Size** затем нажмите OK.

Циклические ссылки

Условное ограничение используется, для того чтобы защитить вашу имитацию от циклических ссылок. Если ваша имитация использует высоко вложенный GPSS или вызов PLUS процедуры, то, возможно, что вы захотите увеличить это значение. Вы можете изменить предел посредством изменения предела на странице имитации (Simulate Page) в записной книжке настроек модели (Model Settings Notebook). Выберите **Edit / Settings** и перейдите на Simulate Page введите новый предел в строке **Max Evaluation Depth** затем нажмите OK.

Глава 3 – Выражения модели

Модель GPSS World - это последовательность выражений модели (Model Statements). Выражение модели может быть определением GPSS выражения или процедуры PLUS. Выражение GPSS - это либо команда, либо выражение блока (Block Statement). PLUS - это акроним означающий «Язык программирования симуляции» (Programming Language Under Simulation), язык программирования вложенный в GPSS. PLUS эксперименты - вид процедуры. Эта глава описывает использование выражений модели в среде имитации GPSS World. Так же она описывает общую структуру выражений GPSS. Отдельные команды GPSS рассмотрены в деталях в главе 6, выражения блока в главе 7. В утверждении PLUS только PROCEDURE и EXPERIMENT выражения модели. PLUS описана в деталях в главе 8.

3.1 Использование выражений модели

До того как вы сможете запустить имитацию, вы должны создать имитационный объект. Эта процедура называется трансляция первичной модели (Initial Model Translation). Это просто. Для этого нужно открыть окно модели на объекте модели и выбрать **Command / Create Simulation**. Если необходимо внести некоторые поправки, то только в результате синтаксических ошибок. В том случае, если изменение модели успешно, объект моделирования автоматически скопирует оттранслированную модель и выполнит любую команду, вложенную в файлы модели. Теперь вы можете открывать любое окно поверх имитации. Если вы разместите одну или несколько команд START в одном из файлов модели, то объект моделирования выполнит их по порядку с другими командами, не дожидаясь команды к выполнению. Любое выражение модели может быть частью трансляции первичной модели или может быть послано существующей имитации позже как интерактивная переменная.

3.1.1 Компиляция первичной модели (Initial Model Translation)

Когда модель успешно скомпилирована, то создаётся структура блока, список процедуры регистрации, список команд, и всё это посылаётся объекту моделирования. Когда, модель скомпилирована, последовательность блоков сохраняется, и отдельно сохраняется первоначальная последовательность команд. Когда объект моделирования получает команду, то он или немедленно выполняет её, либо размещает её в очередь чтобы обратиться к ней в порядке очереди.

3.1.2 Интерактивное выражение

После того, как первичная модель скомпилирована, вы можете послать выражения модели существующему объекту моделирования. Эти выражения называются интерактивными выражениями. Они уже скомпилированы и отсылаются объекту моделирования, который принимает и выполняет их. То, что происходит, зависит от того какое выражение модели послано. Выражения PROCEDURE и EXPERIMENT определяют или переопределяют процедуру PLUS в имитации. Команды исполняются немедленно (SHOW или HALT), или размещаются в командной очереди имитации. Интерактивные переменные блока выполняются немедленно в режиме управления методом моделирования. Это означает, что создаётся временный блок и активная транзакция пытается войти в него. Далее блок уничтожается.

3.1.3 Последовательность блоков

Структура блоков имитации устанавливается в момент изменения объекта модели. Он не изменяется до следующего изменения. Во время трансляции модели, преобразователь взаимодействует с каждым выражением блока и прикрепляет ему представление имитационной структуры блока. После того, как имитация началась, то она представляет собой оригинальную последовательность выражений блока, которая определяет последовательность действий в имитации. Пока поток транзакций не изменён, каждая активная транзакция пытается войти в каждый блок по очереди. Выражения блока, принятые имитацией, после того как файл модели изменён, не встраиваются в блоковую структуру имитации. Вместо этого они становятся причиной создания временного блока, используемого и удаляемого в режиме управления методом моделирования.

3.1.4 Последовательность команд

Команды делятся на неотложные и очередные. Только команды HALT и SHOW неотложные, все остальные команды являются очередными. Когда объект моделирования принимает неотложную команду - она сразу выполняется. Очередные команды располагаются в командной очереди моделирования. Объект моделирования же выполняет эти команды по очереди, прерываясь на выполнение неотложных команд. Список команд создаётся во время компиляции первичной модели и отсылается объекту моделирования, так же как и интерактивные команды. Когда компилятор встречает команду во время компиляции первичной модели, он помещает её в список. Когда вся модель скомпилирована, список команд отсылается объекту моделирования после структуры блоков и списка процедур регистрации. Команды START, CONTINUE, и CONDUCT являются очередными. Когда объект моделирования достаёт одну команду из командной очереди, моделирование выполняется до тех пор, пока счётчик окончания не станет отрицательным. Остальные команды ждут своей очереди.

Команда CONDUCT используется для запуска эксперимента, который является типом процедуры PLUS. Эксперимент регистрируется в объекте моделирования, так же как и любая другая PLUS процедура и он может иметь аргументы, переданные ему командой CONDUCT. Команда HALT это особый случай. Она имеет свойство очищать всю командную очередь. Когда вы посылаете команду HALT объекту моделирования, моделирование переводится в состояние остановки, и все типы активности завершаются от лица моделирования. Вы можете управлять моделированием как вам угодно.

3.1.5 Последовательность процедур

Во время трансляции первичной модели, транслятор встречает процедуру или выражение эксперимента и записывает процедуру перевода в регистрационный список процедур. Объект моделирования регистрирует все определённые пользователем процедуры, когда получает оттранслированную модель. Когда объект моделирования принимает выражение интерактивной процедуры, он встраивает процедуру транслирования в библиотеку процедур имитации, размещая любую существующую процедуру под тем же именем. Вызов процедуры происходит, когда встречается выражение с вызовом процедуры или выражение вызова процедуры во время выполнения процедуры PLUS. Вызовы могут быть представлены интерактивно посредством размещения вызова процедуры внутри операнда или команды SHOW. Эксперименты могут быть вызваны только командой CONDUCT. Процедуры, вызванные во время эксперимента, имеют особенное свойство: они могут обращаться к библиотеке процедур DoCommand. Это рассмотрено в главе 8.

3.1.6 Сохранение объектов

Все объекты GPSS могут быть сохранены и открыты в последующих сессиях. Модели, имитации, отчёты и текстовые объекты могут быть изменены и сохранены в любое время. Командное меню **File / Open** позволяет вернуть объект обратно. Имитация может быть сохранена в любой точке своего выполнения с целью отладки, демонстрации, перезапуска, или можно просто сохранить объект моделирования в промежуточном состоянии. Процедура сохранения вызывает команду HALT посылаемую объекту моделирования. В случае эксперимента вам необходимо сохранить состояние завершённой части эксперимента в глобальных переменных и глобальных матрицах, если вы не хотите перезапускать эксперимент с самого начала, когда откроете его.

3.2 Выражения GPSS

Существуют два типа GPSS выражений: блоковые выражения, которые создают сущность блока GPSS и команды, которые не делают ничего. Команды определяют не-блоковые сущности или заставляют выполняться действие. Выражения PLUS здесь не рассматриваются. В отличие от выражений PLUS, выражения GPSS составлены из частей, называемых *полями*. Поля - это переменное количество печатных символов оканчивающихся пробелом или разделителем. Не смотря на то, что точный набор выражений мало изменяется, в общем они состоят из следующего: **Номер строки(опционально)/ Ярлык(либо название)(опционально)/ команда(обязательно)/ Операнды(зависят от команды)/ Комментарий(опционально)**. Произвольные номера строки удерживаются для совместных целей. Если используются, они должны начинаться в колонке 1. Однако, они игнорируются компилятором. Номер строки выдаётся в сообщении об ошибке в форме абсолютного количественного числителя, означающего место строки в файле модели, и не является номером строки выбранной пользователем.

3.2.1 Команды GPSS

Команды описаны детально в главе 6. Вот они:

- **BVARIABLE** – Определяет сущность булевой переменной.
- **CLEAR** – сбрасывает статистику и завершает транзакцию.
- **CONDUCT** – выполняет регистрацию PLUS Experiment.
- **CONTINUE** – продолжает имитацию.
- **EQU** – устанавливает значение пользовательской переменной.
- **EXIT** – Заканчивает сессию GPSS World.
- **FUNCTION** – Определяет сущность функции.
- **FVARIABLE** - определяет сущность Fvariable.
- **HALT** – останавливает имитацию и удаляет все команды из командной очереди.
- **INCLUDE** – Читает и транслирует вторичный файл модели.
- **INITIAL** - Инициализирует или модифицирует Logicswitch, Savevalue, or Сущность матрицы.
- **INTEGRATE** – Автоматически объединяет временной дифференциал в пользовательской переменной.
- **MATRIX** – Определяет сущность матрицы.
- **QTABLE** – Определяет сущность Qtable.
- **REPORT** – Устанавливает имя файла отчёта или запрашивает немедленный отчёт.
- **RESET** – сбрасывает статистику имитации.
- **RMULT** – Устанавливает источник первых 7 генераторов случайных чисел
- **SHOW** – оценивает и отображает выражения.

- **START** – устанавливает счётчик завершения и запускает имитацию.
- **STEP** – ограничивает количество Блок'овых сущностей..
- **STOP** – Устанавливает условие остановки, основанное на попытках входа Блока.
- **STORAGE** – определяет сущность памяти.
- **TABLE** – Определяет сущность таблицы.
- **VARIABLE** – определяет сущность переменной.

3.2.2 Выражения блока GPSS

Выражения блока создают блок GPSS. Когда транзакция вводит сущность блока, происходит специальное действие, которое определяется типом блока. Выражение блока может быть вставлено в модель объекта с помощью полноэкранного текстового редактора или с помощью диалога создания блока. Так же можно использовать текстовый редактор для размещения выражения блока в текстовые объекты. Блоковая структура имитации устанавливается, когда модель объекта скомпилирована. После того как имитация начата, оригинальная последовательность блоковых выражений определяет последовательность действий имитации. Пока поток транзакций не изменён, каждая активная транзакция пытается ввести один блок после другого в оригинальной последовательности. Блоковые выражения принимаются имитацией после того, как оригинальный файл модели скомпилирован и не встраиваются в структуру блоков имитации. Взамен, они являются причиной для создания временного блока, используемого и уничтожаемого в процессе управления методом моделирования, который обсуждался выше.

Выражения блока GPSS описаны в деталях в главе 7. Вот они:

- **ADOPT** – изменяет набор ассемблирования.
- **ADVANCE** - Размещает транзакцию в будущей цепи событий.
- **ALTER** – Тестирует и изменяет транзакции в группе.
- **ASSEMBLE** – ожидает и уничтожает связанные транзакции.
- **ASSIGN** – изменяет параметр транзакции.
- **BUFFER** - Размещает транзакцию в цепи текущих событий
- **CLOSE** – Заканчивает поток данных.
- **COUNT** – размещает счётчик сущностей в параметре транзакций.
- **DEPART** – уменьшает содержание очереди сущности.
- **DISPLACE** – изменяет следующий последовательный блок транзакции.
- **ENTER** - Занимает или ожидает память
- **EXAMINE** – проверяет членство группы.
- **EXECUTE** – представляет действия определённое разными блоками.
- **FAVAIL** - Изменяет статус сущности Средства на «доступен».
- **FUNAVAIL** - Изменяет статус сущности Средства на «не доступен».
- **GATE** – проверяет сущность и изменяет поток транзакций.
- **GATHER** – ожидает связанных транзакций.
- **GENERATE** – создаёт транзакцию и размещает её в цепи будущих событий.
- **INDEX** – изменяет параметр транзакции.
- **INTEGRATION** - переключает объединение пользовательской переменной на «On» или «Off»
- **JOIN** - Размещает члена в Числовой или Транзакционной группе.
- **LEAVE** – освобождает память.
- **LINK** – переносит транзакцию в сущность Userchain.
- **LOGIC** - Изменяет сущность Logicswitch.
- **LOOP** – Уменьшает параметр, переходит к различным блокам, если результат не ноль.

- **MARK** -Размещает значение системных часов в параметр транзакции.
- **MATCH** - Ожидает связанной транзакции для достижения соединённого блока MATCH.
- **MSAVEVALUE** – Назначает значение элементу матрицы.
- **OPEN** – Инициализирует поток данных..
- **PLUS** - Оценивает PLUS Expression и сохраняет результат в параметре.
- **PREEMPT** – вытесняет владельца Facility.
- **PRIORITY** – изменяет приоритет транзакции.
- **QUEUE** – увеличивает содержание очереди.
- **READ** – достаёт следующую линию данных из потока данных.
- **RELEASE** - Освобождает сущность Facility.
- **REMOVE** – удаляет члена из числовой или транзакционной группы.
- **RETURN** - Освобождает сущность Facility.
- **SAVAIL** – изменяет статус памяти на «available»
- **SAVEVALUE** – устанавливает значение сущности Savevalue.
- **SCAN** – проверяет транзакционную группу, устанавливает значение в параметр.
- **SEEK** – изменяет указатель строки в потоке данных.
- **SEIZE** – принимает на себя собственность или ждёт сущность Facility.
- **SELECT** – размещает номер выбранной сущности в параметр транзакции.
- **SPLIT** – создаёт связанную транзакцию.
- **SUNAVAIL** – изменяет статус памяти на "not available".
- **TABULATE** – обновляет таблицу.
- **TERMINATE** – удаляет транзакцию, уменьшает счётчик остановки.
- **TEST** – проверяет арифметическое условие и изменяет поток транзакций.
- **TRACE** – устанавливает отслеживающий индикатор активной транзакции.
- **TRANSFER** – переходит к определённому блоку.
- **UNLINK** – убирает транзакцию из сущности Userchain.
- **UNTRACE** - выключает отслеживающий индикатор активной транзакции.
- **WRITE** – посылает значение в поток данных.

3.3 Поля

Часть Выражений GPSS называется полями. Выражения GPSS являются свободными в том смысле, что вам не нужно будет беспокоиться о построении полей с номерами колонок, только о построении номеров строк. Этот формат почти одинаковый для всех выражений, с некоторыми полями, необязательными для некоторых команд. В общем, поля выражений выстраиваются слева направо в следующей последовательности:

- Номер строки(опционально). Игнорируется.
- Ярлык (переменная, зависящая от поля команды).
- Команда (обязательно).
- Операнды, включая все условные операторы (переменные зависящие от глагола).
- Комментарий (опционально).

3.3.1 Номера строк

Номера строк игнорируются GPSS World. Они разрешены только в выражениях GPSS, и если используются, они должны начинаться с первой колонки. Они сохраняются для целей совместимости.

3.3.2 Ярлыки

Поле ярлыка позволяет вам называть и ссылаться на сущность с помощью имени, которое вы выберете. Ярлык должен подчиняться правилам создания имени, которые описаны позже в этой главе. Вы можете назначать ваши собственные значения именам с помощью команд EQU и операторов присвоения PLUS. Это полезно, когда вы хотите поэкспериментировать с различными значениями переменных. Однако, проще позволить GPSS World использовать его собственные уникальные внутренние номера, когда вы именуєте сущности и параметры транзакций. Если вы решили назначать свои собственные номера именам сущностей, вы должны делать это в выражениях EQU, до того как вы определите или обратитесь к сущности по этому имени. Если вы назначаете свои собственные значения именам, то должны взять на себя ответственность избегать возможных конфликтов. Когда GPSS World сталкивается с новым именем, он назначает ему уникальное целое число. Эти целые числа начинаются с 10000. Вы можете использовать одно и тоже имя для ссылки на разные типы сущностей. Например, вы можете назвать память и таблицу именем Motropool. Однако, если вы используете имя Motorpool в качестве ярлыка во второй команде STORAGE, старый объект Storage будет заменен новым. Постоянные блок-сущности, не заменяются таким образом. Блоки особенный случай. Интерактивное выражение блока выполняется как временный блок ручного метода моделирования. Если вам нужно будет изменить характеристики блока в середине имитации, то вы должны переопределить значения имён используемые в операндах с помощью выражений EQU. Вы также можете использовать исполнительный блок и интерактивно изменять операнд. Блоки не могут быть введены интерактивно. Изменение структуры блоков требует перекомпиляции. Вы не можете использовать номер в качестве имени ярлыка. Это может нарушить правила именования. Если вы хотите точно контролировать количество сущностей, вы можете устанавливать значения имён, используя выражения EQU до того, как использовать имя как ярлык. Вы можете не использовать SNA в качестве ярлыка блока.

3.3.3 Действия (команды)

Поле действия следует за полем ярлыка и требует ключевое слово GPSS. Оно должно быть одним из имён блоков GPSS или одной из команд. Список доступных действий доступен через средства помощи.

3.3.4 Операнды

Большинство действий следуют за одним или более полями операторов и операндов. Эти поля зависят от команды. Некоторые из них обязательны, некоторые нет. Онлайн-овое средство помощи покажет вам, какой синтаксис допустим. Когда вы хотите закончить или опустить операнд, вы должны нажать символ разделителя. Традиционно операторы разделяются запятыми. Если вся информация в утверждении была представлена, вы можете пропустить необязательные поля. Формы, которые вы можете использовать в предложенном поле операнда, описаны в специальном описании выражений в главе 6 и 7. Данная информация доступна в онлайн-овой помощи.

3.3.5 Комментарии

Существуют два типа полей комментариев. Первый начинается с «*» или «;». Второй - необязательное поле комментария следует за полем операнда любого выражения. функции последовательностей и выражений с полями выражений, не имеют полей комментариев. Когда все требуемые операнды выражения завершены вы можете нажать

«;» для начала комментирования. В этот раз вы можете ввести комментарий, как в верхнем, так и в нижнем регистре и с помощью ASCII символов.

3.4 Выражения

3.4.1 Выражения в выражениях GPSS

Выражения в GPSS World определены как часть языка PLUS. Вы можете использовать их в операндах выражений GPSS и процедурах PLUS. Некоторые операнды блока допускают вставление выражений PLUS. Список допустимых синтаксических переменных будет затем включать *Parenthesized Expression*, как единственную или как альтернативную форму операнда. Некоторым командам не нужно использование внешних скобок, но если вы будете всегда заключать в скобки PLUS выражения в выражениях GPSS, вы будете застрахованы от ошибок. Так проще переходить между PLUS и GPSS. Множество операндов выражений GPSS позволяют вам использовать GPSS выражения. Это означает, что вы сможете обеспечить простой фактор, небольшое вычисление или вызов PLUS процедуры прямо в самом операнде. С другой стороны, используя выражения, содержащие SNAs, вы можете получить доступ к переменным GPSS, так же как к глобальным пользовательским переменным, прямо из процедур PLUS. Глава 8.3.2 этого руководства расскажет, как строить выражения PLUS. Все строительные блоки за исключением системы числовых атрибутов рассматриваются там.

3.4.2 Система числовых атрибутов (SNA)

Система числовых атрибутов имитирует переменные состояния, которые доступны для использования во время имитации. Она возвращает числовые или строковые значения и может быть использована выражениями GPSS. Некоторые SNA поддерживаются автоматически, а другие требуют вычислений при обращении к ним. Например, накопленный счётчик сущностей внутри блока доступен как SNA. Существует более 50 классов SNA, которые способствуют мощности вашей имитации. Довольно мудро использовать их все. Многие SNA, такие как PR, оцениваются от лица активной транзакции. Это означает, что некоторые атрибуты транзакций необходимы в обязательном порядке для оценки SNA. Т.е, SNA не могут быть оценены при наличии только лишь активной транзакции. Например, вы не можете ссылаться на параметры транзакции операндов блока GENERATE, потому что когда транзакция входит в блок GENERATE в первый раз, она не имеет параметров. Другие SNA, как например, системные часы C1, могут быть оценены без ссылки на транзакцию.

Признаки сущностей SNA

Многие SNA могут быть определены в одной из множества форм, начиная с классов. Например, SNA W22 возвращает количество транзакций ожидаемых в блоке номер 22. Идентификатор блока в этом случае это номер 22, но так же существует несколько других возможностей. Эта SNA может быть определена с помощью следующих признаков сущностей:

- **Wj** - где *j* положительное целое число, количество блоков в имитации.
- **W\$Name** - где *Name* местоположение желаемого блока.
- **W*j** - где *j* положительное целое число, the количество параметров активной транзакции которая содержит количество желаемых блоков. Это косвенная адресация.
- **W*Name** - где *Name* имя параметра активной транзакции, которая содержит количество желаемых блоков. Это косвенная адресация.

- **W*\$Name** - где *Name* имя параметра активной транзакции, которая содержит количество желаемых блоков. Символ \$ не нужен и используется только как разделитель. По существу, это такой же **W*Name**. Это косвенная адресация.
- **W*Parameter** – указывает что каждая из **W*j**, **W*Name**, или **W*\$Name** может быть использована.

Матрицы SNA класса MX это особый случай. Они могут содержать до трёх косвенных адресаций. Например, **SNA MX*Sales(*Partnumber,*January)** определяет матрицу, чей номер в параметре транзакции называется Sales и затем достаёт значения строк и столбцов, чьи имена Partnumber и January. Обычно, Активная транзакция прошла бы через три блока ASSIGN, инициализируя параметры Sales, Partnumber, и January с номером или именем желаемого значения. Некоторые классы SNA, такие как A1, AC1, C1, M1, MP, PR, и TG1 являются «крошечными» SNA. Эти крохотные SNA являются целостными и не требуют имени или номера для завершения оценки SNA.

Доступность SNA

Следующие SNA доступны для использования в полях операнда и выражениях команд и выражений. Во всех случаях *Entnum* должен быть заменён любым признаком сущности. Признаком сущности может быть имя (со знаком разделения \$), или номер, или косвенная адресация, или маленькой звёздочкой «*», следующей после имени или номера. Формальные правила построения операторов приведены в Приложении.

Это полный список SNA доступных в GPSS World:

- **A1** - Комплект активных транзакций. Целое значение.
- **AC1** - Абсолютное значение системных часов. Время с последней команды CLEAR. Real значение.
- **BVEntnum** – Результат оценки булевой переменной сущности *Entnum*. Real значение.
- **C1** – Значение относительных системных часов. Время с последней команды RESET. Real.
- **CAEntnum** - Среднее содержимое Userchain.
- **CCEntnum** – Общее количество Userchain. Количество всех транзакций связанных с Userchain *Entnum*. Целое значение.
- **CHEntnum** - Текущее содержание Userchain. Текущее количество транзакций связанных с Userchain *Entnum*. Целое значение.
- **CMEntnum** – Максимальное содержание Userchain. Максимальное количество транзакций связанных с Userchain *Entnum*. "Высшая отметка наводнения". Целое значение.
- **CTEntnum** – Среднее время пребывания UserchainСредняя продолжительность транзакций Userchain *Entnum*. Real значение.
- **FEntnum** – Занятое средство. Если средство *Entnum* занято, **FEntnum** возвращает 1. В противном случае **FEntnum** возвращает 0. Целое значение.
- **FCEntnum** – Счётчик захвата средства. Количество раз, которые средство *Entnum* было SEIZED или PREEMPTed транзакцией. Целое значение.
- **FIEntnum** - Прерванное *Entnum* средство. Если средство *Entnum* выгружено, **FIEntnum** возвращает 1. В противном случае **FIEntnum** возвращает 0. Целое число.
- **FNEntnum** - Функция. Результат оценки функции *Entnum*. Real число.
- **FREntnum** – Использование средства. Фрагмент времени которое средство занято *Entnum*. **FREntnum** выражается в частях на тысячу и затем возвращаются значения от 0-1000, включительно. Может быть не целым. Real число.
- **FTEntnum** - Среднее время задержки средства. Среднее время, в течение которого сущность *Entnum* занята транзакцией. Real значение.

- **FVEntnum** - Средство в доступном состоянии. *FVEntnum* возвращает 1 если средство *Entnum* доступно, 0 в ином случае. Целое значение.
- **GNEntnum** – Счётчик числовых групп. *GNEntnum* возвращает счётчик членства числовой группы *Entnum*. Целое значение.
- **GTEntnum** – Счётчик Транзакционных групп. *GTEntnum* возвращает счётчик членства транзакционной группы *Entnum*. Целое число.
- **LSEntnum** – Набор логических выключателей. *LSEntnum* возвращает 1 если значение логического выключателя находится в состоянии «set», 0 в ином случае. Целое значение.
- **MBEntnum** – Соответствие блоков. *MBEntnum* возвращает 1 если есть транзакция в блоке *Entnum* который находится в том же наборе что и активная транзакция. *MBEntnum* возвращает 0 в ином случае. Целое число.
- **MPPParameter** – Время передачи, Параметр. Текущее значение системных часов минус значение параметра. Real значение.
- **MXEntnum(m,n)** – Значение матрицы Savevalue. Возвращается значение строки m, колонки n матрицы *Entnum*. В матрицах имеющих более двух измерений, все остальные значения предполагаются равными 1. В отличии от MX класса SNAs, PLUS Expressions могут ссылаться на каждый элемент многомерной матрицы.
- **M1** – Время передачи. *M1* возвращает абсолютное значение системных часов минус «метка времени» Транзакции. Real число.
- **NEntnum** – Счётчик вхождения блока. Возвращается общее количество транзакций которые входили в блок *Entnum*. Целое значение..
- **PParameter** or ***Parameter** – Значение параметра. *PParameter* или **Parameter* возвращает значение *Parameter* активной транзакции. Целое, Real, строковое значение.
- Косвенная ссылка использует нотацию *SNA*Parameter*.
- **PR** – Приоритет транзакции. Значение приоритета активной транзакции. Целое значение.
- **QEntnum** – Текущее содержимое очереди. Текущее число значений очереди *Entnum*. Целое значение.
- **QAEntnum** – Среднее значение очереди. Среднее по времени число очереди *Entnum*. Real значение.
- **QCEntnum** – Общее поступление в очередь. Сумма всех значений входа в очередь *Entnum*. Целое значение.
- **QMEntnum** – Максимальная величина очереди. Максимальное количество очереди *Entnum*. Целое значение.
- **QTEntnum** - Среднее время задержки очереди. Среднее по времени число задержки очереди. Real значение.
- **QXEntnum** – Среднее время задержки очереди за исключение нулевых входов. Среднее по времени значение для очереди *Entnum* без учёта вхождений имевших нулевую задержку в очереди. Real число.
- **QZEntnum** - Счётчик нулевых вхождений в очередь. Количество входов очереди *Entnum* с нулевым временем задержки. Целое значение.
- **REntnum** – Доступный объём памяти. Содержимое памяти доступное для использования входящими транзакциям. Целое число.
- **RNEntnum** – Случайное число. *RNEntnum* возвращает случайное целое число от 0 до 999 из генератора случайных чисел *Entnum*. Целое значение.
- **SEntnum** – Используемая память. *SEntnum* возвращает количество памяти используемой вошедшей транзакцией. Целое значение.
- **SAEntnum** – Среднее значение используемой памяти. *SAEntnum* возвращает среднее по времени значение используемого объёма памяти. Real значение.

- **SCEntnum** – Счётчик используемой памяти. Общее количество секторов памяти. Целое значение.
- **SEEntnum** – Свободная память. *SEEntnum* возвращает 1 если память полностью доступна, 0 в ином случае. Целое значение.
- **SFEntnum** – Заполнение памяти. *SFEntnum* возвращает 1, если память полностью используется, 0 в ином случае. Целое значение.
- **SREntnum** – Использование памяти. Часть общего использования представленная средним значением используемой памяти. *SREntnum* выражается в частях на тысячу и затем возвращается значениями от 0 до 1000, включительно. Может быть нецелочисленным. Real значение..
- **SMEntnum** – Максимальное значение используемой памяти. Целое значение.
- **STEntnum** - Average holding time per unit at Storage Entity *Entnum*. Real value.
- **SVEntnum** – Доступная память. *SVEntnum* возвращает 1, если память в доступном состоянии, 0 в ином случае. Целое значение.
- **TBEntnum** – Безоценочные входы в таблицу *Entnum*. Real значение.
- **TCEntnum** – Счётчик безоценочных входов в таблицу *Entnum*. Целое значение.
- **TDEntnum** –Стандартное отклонение безоценочных входов в таблицу *Entnum*. Real значение.
- **TG1** – Итог оставшихся окончаний. TG1 возвращает итог который уменьшен посредством TERMINATE Blocks с помощью положительного операнда A. Это значение инициализируется Выражение м START и показывает конец имитации, когда значение равно 0. Целое число.
- **VEntnum** – Результат арифметической оценки или переменной плавающей точки.. Real значение.
- **WEntnum** – Количество текущих блоков. Текущее число транзакций в блоках. Целое значение.
- **XEntnum** – Сохранённое значение. Возвращается значение сохранённого значения. Целое, Real, строковое знаение.
- **XN1** – Номер активной транзакции. Возвращает номер активной транзакции. Целое значение.
- **Z1** – Свободная память. Значение, возвращаемое операционной системой. Целое значение.

Специальные правила возвращаются, когда имитация запущена в режиме совместимости GPSS\PC. Данный аспект будет более полно озвучен ниже.

Операторы

Операторы служат для объединения элементов данных в выражениях. Данные выполняют действия по принуждению оператора. Например, если числовая операция должна выполняться на переменной со значением строки, используется числовой эквивалент, полученный от символов в строке.

Операторы, используемые в GPSS World следующие:

- **^** Возведение в степень. A^B возвращает A в степени B.
- **# (or *)** Умножение. $A \# B$ возвращает результат произведения A на B.
- **/** Деление. A / B возвращает частное от деления A на B.
- **** Целочисленное деление. $A \setminus B$ возвращает результат целочисленного деления A на B.
- **@** Остаток от целочисленного деления. $A @ B$ возвращает модуль от деления A на B.
- **-** Вычитание. $A - B$ возвращает разность между A и B.
- **+** Сумма. $A + B$ возвращает сумму A и B.

- **>= 'GE'** Больше или равно. $A \geq B$ возвращает 1 если A численно больше или равно B, 0 в ином случае.
- **<= 'LE'** Меньше или равно. $A \leq B$ возвращает 1, если A численно меньше или равно B, 0 в обратном случае.
- **'G'** Больше чем. $A > B$ возвращает 1 если A численно больше чем B, 0 в ином случае.
- **< 'L'** Меньше чем. $A < B$ возвращает 1 если A численно меньше чем B, 0 в ином случае.
- **= 'E'** Равно $A = B$ возвращает 1 если A численно равно B, 0 в ином случае.
- **!= 'NE'** Не равно. $A \neq B$ возвращает 1 если A численно отличается от B, 0 в ином случае.
- **& 'AND'** Логическое «И». $A \& B$ возвращает 1 тогда и только тогда, когда A и B не нули, 0 в ином случае.
- **| 'OR'** Логическое или. $A \text{ 'OR' } B$ возвращает 1 если A или B или оба не нули, 0 в ином случае.

Последовательность(приоритет операций)

Когда выражения вычислены, последовательность операторов определяет порядок операций. Оператор с более высоким приоритетом вычисляется раньше, чем оператор с более низким приоритетом. Приоритеты операторов по степени убывания приведены ниже:

- **^** Степень
- **# (or *) / ** Умножение, Деление, Целочисленное деление
- **@** Остаток от деления
- **- +** Сложение, Вычитание
- **>= <= > <** Операторы сравнения
- **= !=** Равно, не равно
- **&** Логическое И
- **|** Логическое ИЛИ

Косвенная адресация

Все объекты имеют положительные номера. Когда вы ссылаетесь на объект по имени, Значение имени сопоставляется с номером объекта. Косвенная адресация означает, что номер объекта это параметр транзакции. Для использования косвенной адресации вы должны использовать значок * (или # -- смотри настройки) в операнде. Например, SNA Q*2 использует значение параметра 2, как номер объекта очереди. SNA возвращает текущее значение этого объекта очереди.

Перед оценкой величины SNA необходимо переместить номер объекта (или значение имени) в параметр транзакции.

Непрямая адресация использует формат записи «SNA**параметр*». Приведем другой пример: для захвата станции, номер которой представляет собой значение, возвращенное функцией, номер которой находится в параметре под именем «Tellersector», необходимо ввести **SEIZE FN*Tellerselector**.

3.4.3 Выражения в режиме совместимости GPSS/PC

В GPSS World промежуточные результаты и значения SNA никогда не округляются, если пользователь явно не указывает этого. В то же время, результаты округления промежуточных результатов в целях совместимости сохраняются в режиме совместимости GPSS/PC. Когда симуляция производится таким способом, SNA всегда возвращает и округляет целые числа (если они не используются в качестве

модификаторов функций), а промежуточные результаты, полученные при оценке В-переменных и переменных единиц, округляются.

Использование режима совместимости GPSS/PC описывается в главе 1.

Если в режиме совместимости выражение используется для определения SNA, результат всегда округляется до целого числа. Необходимо выбирать единицы времени для симуляции таким образом, чтобы эти округления не были значимыми.

В режиме совместимости на оценку выражения влияет команда, в которой она содержится.

- **SHOW** выполняет все операции в числах с плавающей точкой двойной точности и отображает результаты в окне данных как числа с плавающей точкой двойной точности.
- **VARIABLE** создаёт арифметическую переменную GPSS, которая при оценке запросом SNA выполняет все операции в числах с плавающей точкой двойной точности, а затем округляет все промежуточные результаты перед переходом к следующей операции. Общий результат также округляется и возвращается.
- **FVARIABLE** создаёт GPSS-переменную "с плавающей точкой", которая при оценке запросом SNA выполняет все операции в числах с плавающей точкой двойной точности, а затем округляет все промежуточные результаты. Дробные значения могут отображаться при правильном выборе единиц, например, «тысячных». Значение SNA, равное 500, интерпретируется как число 1/2.
- **BVARIABLE** создает булеву переменную GPSS, которая при оценке запросом SNA выполняет все операции в числах с плавающей точкой двойной точности, а затем возвращает 1 при ненулевом результате и 0 во всех остальных случаях.

Команды **FUNCTION**, отличные от типа C, оценивают аргумент и выбирают член списка на основе результата для определения итогового значения, которое округляется.

Оценка **FUNCTION** типа C начинается с оценки аргумента. Результатом всегда является целое число, используемое для идентификации сегмента функции. Аргумент далее используется в линейной интерполяции двойной точности для получения результата функции двойной точности. Если аргумент является SNA RN-класса, то используется случайная дробь между 0 и .999999 включительно. Если функция не используется как модификатор функции, итоговый результат округляется.

Все SNA, за исключением продолжительных функций, используемых в качестве модификаторов функций, возвращают целое число.

3.5 Имена

Вы можете создавать имена блоков, пользовательских переменных и других объектов GPSS. Имя должно начинаться с буквы, может содержать до 200 буквенных и цифровых знаков и подчёркиваний (_).

При выборе имени необходимо избегать использования ключевых слов, системных числовых атрибутов или классов системных числовых атрибутов, а применять легко запоминающееся имя.

Имя не может быть глаголом или частью глагола, ключевым словом, а также действующим SNA. Если вы не уверены в том, какие имена допустимы, включите в создаваемые имена подчёркивание (_). Разрешено также использовать имена, начинающиеся с как минимум 3 букв и затем цифры. В любом случае, GPSS World не допустит создания недействительного имени.

3.5.1 Ярлыки

Имена, используемые в поле «Ярлык» выражения GPSS используют для ссылки на объекты GPSS и называются ярлыками объектов или просто ярлыками. Ярлыки блоков также называются местоположениями.

Местоположения являются назначенными значениями, соответствующими номеру блочного объекта. Имя местоположения блока не следует использовать для других целей. Попытка такого использования вызовет в GPSS World ошибку «Stop». Это позволяет избежать возможности случайного изменения соответствия блока с ярлыком.

Когда объект симуляции получает имя от сеанса, его имени присваивается уникальное числовое значение. Эти системные значения начинаются со значения 10000. Вы можете назначить своё значение любого типа данных позже.

Если необходимо принудительно назначить именованному объекту выбранный номер, используйте это имя в утверждении EQU перед определением (или ссылкой) объекта по имени. Пользователь должен самостоятельно избегать случая, когда различные имена ссылаются на один объект. Для ссылки на объект каждого типа можно использовать одно имя. Например, SNA Q\$BARBER и F\$BARBER ссылаются на разные объекты. Первый ссылается на объект очереди, а второй – на объект станции. Когда одно имя ссылается на различные типы объектов, путаницы не происходит.

При создании единицы с ярлыком, текущее значение ярлыка принимается как постоянный номер объекта. Даже если именное значение в дальнейшем изменяется, номер созданного объекта остаётся неизменным.

При возможности объект симуляции создаст объект GPSS при первой встрече именной ссылки на этот тип объекта. Тем не менее, объекты, которые не могут быть созданы без дополнительной информации, например, объекты хранения, должны быть определены командами GPSS перед ссылкой. Единицы, которые требуют определения перед упоминанием, описаны в главе 4.

3.5.2 Пользовательские переменные

Помимо использования в ярлыках, имена могут представлять собственные значения пользователя. Такие переменные называются пользовательскими переменными и создаются по их появлению в команде EQU или в выражении PLUS. Они выполняют обычную роль программных переменных. Пользовательское имя, которое присваивается значению через команду EQU, но затем появляется в качестве ярлыка объект, называется ярлыком.

Пользовательские переменные могут содержать цифровые или строковые значения во время моделирования. На них можно ссылаться в выражениях PLUS и их можно изменять командами EQU или выражениями назначения в процедурах PLUS. Кроме того, их можно автоматически обновить путем интеграции. Это обсуждается в главе 4, в разделе «Продолжительная симуляция».

3.5.3 Область имён

Помимо временных имен все пользовательские имена известны по всей модели. Исключение составляют имена созданные в выражениях TEMPORARY или TEMPORARY MATRIX процедуры PLUS. Новые экземпляры временных именованных значений или матриц создаются при запуске процедуры PLUS и удаляются при выходе процедуры PLUS. Эти имена переопределяют любые другие похожие элементы в модели во время выполнения процедуры PLUS. Другие имена, использованные в процедуре PLUS, относятся к объектам, известным по всей модели.

В одном эксперименте необходимо размещать результаты симуляции в глобальные переменные и глобальные матрицы для их сохранения между симуляциями. Например, процедура библиотеки ANOVA требует размещения объекта глобальной матрицы в качестве аргумента. Кроме того, во время эксперимента доступна процедура библиотеки DoCommand. Любые строки, переданные DoCommand, не могут включать переменные

TEMPORARY и TEMPORARY MATRIX или процедур аргументов, так как строка переводится в глобальную область, в которой не распознаются временные переменные.

3.6 Числа

Числа могут быть сохранены внутри трёх типов данных: целое число, натуральное и строковое. Большинство переменных могут принимать любой из этих типов данных. Так как числовые операторы преобразуют их операнды в подходящую форму, все три типа данных могут использоваться в выражениях.

Целые числа хранятся в виде 32-битных соответствующих номеров. Если целое число превышает значение, оно преобразуется в натуральное.

Натуральные значения хранятся в виде 64-битный чисел с плавающей точкой двойной точности. Экспонента может принимать значения от - 308 до 308, при чём точность ограничена 15 десятичными единицами.

Строки хранятся как массив символов ASCII любой длины, ограниченные параметром **«Max Memory Request»** на странице симуляции настроек объекта симуляции. Настройки можно просматривать с помощью меню **Edit / Settings**.

Системные числовые атрибуты теперь возвращают значения, которые могут иметь значения целые, натуральные или строковые.

Если во время оценки выражения происходит переполнение, возникает ошибка «Stop». Арифметические исключения могут возникать при оценке деления, стандартных отклонений или логарифмов. Неверный аргумент функций библиотеки также вызывает ошибку «Stop».

При печати статистики отчета, когда число слишком велико для места, отведенного в отчёте, нарушается формат отчета, но значение печатается, а отчёт продолжается.

3.7 Использование строк

Строки – это массивы ASCII-символов. Практически любая переменная может принимать строковое значение. Арифметические операторы будут преобразовывать строковые значения к числовым эквивалентам перед выполнением операций.

Строковые константы заключены в двойные кавычки. При вводе строки, например спецификации файла в команде INCLUDE или REPORT необходимо заключать последовательность ASCII-символов в двойные кавычки. Строковые значения, написанные в стандартных отчетах и в ответ на команды SHOW, показываются в двойных кавычках.

Чтобы вставить «строку в строке», используйте 4 дополнительные двойные кавычки вокруг внутренней строки, перемежаемой двумя парами двойных кавычек. Когда транслятор встречает две двойных кавычки, в выходной строке печатается одинарная кавычка.

Строки используются для вывода результатов симуляции в файл результата. Строки также можно использовать для форматирования собственных отчетов о симуляции. Блоки потоков данных OPEN, CLOSE, READ WRITE, SEEK доступны для этих целей так же, как и процедуры библиотеки PLUS под теми же именами. Строки также используются процедурой библиотеки DoCommand при экспериментах. Другим использованием строк является создание следа для ввода в анимационный пост-процессор.

Строками можно манипулировать с помощью процедур библиотеки процедур PLUS. Для комбинации 2 строк используйте процедуру Catenate или PolyCatenate для объединения нескольких строк. Эти и другие процедуры описаны в главе 8.

Глава 4 – Объекты GPSS

GPSS построена на нескольких основных единицах, называемых объектами. Для создания сложных симуляций необходимо понять объекты и правила обращения с ними. Объекты GPSS – это абстрактные объекты, существующие в симуляции. Для более конкретного представления представьте объекты GPSS в виде набора чисел в памяти вашего компьютера. Коллекция объектов называется симуляцией. Наиболее значимыми типами объектов являются транзакции и блоки, так как симуляции, в большинстве своём, состоят из различных транзакций, которые передвигаются от одного блока к другому. Транзакции – это единственные типы объектов, которые можно удалить из симуляции. Всего существует больше десяти типов объектов. Симуляция может содержать множество экземпляров любого типа объектов. Для эффективного создания симуляций необходимо понять свойства каждого объекта GPSS и правила использования GPSS блоков для создания взаимодействия между объектами.

Все объекты GPSS имеют свой номер. Когда происходит обращение объекту по имени, используется связанное с ним целочисленное значение, чтобы найти этот объект. Не обязательно связывать номер с именем объекта, но EQU выражение позволяет Вам это сделать. Обычно GPSS World назначает уникальное значение превышающее или равное 10000.

Большинство GPSS объектов создаются автоматически при необходимости. Например, если имеется ссылка на объект «Facility» с именем Barber, то при отсутствии такого объекта он будет создан. Однако, в этом случае, процесс моделирования может использовать много виртуальной памяти из-за наличия ошибок в GPSS-модели. Модели могут быть очень большими. Использование виртуальной памяти позволяет использовать модели до 512 Мегабайт.

Некоторые объекты должны быть специально объявлены до их использования. Вообще, они должны иметь такой атрибут как, размер. Имя в поле название, называемое «Название объекта» используется для доступа к объекту.

Следующие объекты должны быть объявлены перед использованием:

Объекты памяти должны быть объявлены в области STORAGE.

Арифметические переменные должны быть объявлены в области VARIABLE.

Переменные «с плавающей точкой» должны быть объявлены в области FVARIABLE.

Логические переменные должны быть объявлены в области BVARIABLE.

Матрицы должны быть объявлены в области MATRIX или в области Temporary Matrix PLUS Declarations.

Таблицы должны быть объявлены в области TABLE

Таблицы QTables должны быть объявлены в области QTABLE.

Функции должны быть объявлены в области FUNCTION или Function Follower.

Параметры заявки должны быть объявлены в блоках ASSIGN, MARK, READ, SELECT, SPLIT, COUNT or TRANSFER SUB до того, как на них будут ссылаться.

Некоторые типы объектов имеют одну или более связей с заявками. Объекты Storage и Facility могут частично или полностью быть заполнены заявками. Другие заявки могут перейти в режим ожидания в модели пока дожидаются обработки. Вам решать, что использовать для входа в блоки – RELEASE, RETURN, или LEAVE. Иначе, моделирование может завершиться неуспешно. Некоторые блоки, такие как PREEMPT и FUNAVAIL имеют параметры, которые изменяют маршрут собственной заявки. Однако, все еще необходимо предусмотреть выпуск собственного объекта.

4.1 Объекты заявок

Заявки перемещаются от блока к блоку в модели также как и в реальной системе, которая представлена данной моделью. Как только заявка начала перемещаться в модели, она будет входить в блоки, пока это возможно. В процессе моделирования, заявка, которая

стремится перейти от блока к блоку, называется активной заявкой. Если заявка не может найти подходящего состояния при входе в блок, она становится в режим ожидания. Затем, другая заявка, выбранная для движения через модель, пока она работает, становится в режим ожидания.

Заявки пронумерованы последовательно во всей сессии, начиная с 1. Пустое выражение начинает нумерацию заявок снова с 1.

Поведение заявки определяется несколькими переменными состояниями, называемыми атрибутами заявки. Важные атрибуты заявки:

Параметры - набор значений, связанных с заявкой. У заявки может быть любое количество параметров. Номер параметра – это положительное число. Значение любого параметра активной заявки может быть получено с помощью команды *SNA PParameter*, где *Parameter* – имя или номер параметра. Если операнд блока определяет имя параметра или его номер как искомое значение, *P\$Parameter* or *PParameter* не используются, используется имя или номер параметра отдельно.

Чтобы повысить эффективность, можно разрешить GPSS World прямой доступ к параметрам, используя блоки параметров. Блок Параметра – это массив смежных параметров, которые сгруппированы и расположены, как один сегмент памяти. Когда создается первый параметр, номер которого будет лежать в определенном диапазоне, все другие параметры в сегменте будут располагаться в этом же диапазоне. Когда происходит обращение к параметру в блоке параметров, объект получает доступ к нему быстрее, чем, если бы ему надо было проверять каждый параметр отдельно. Это может сэкономить много времени. Все, что нужно сделать для использования блоков параметров, это объявить их в области Simulation Page в настройках модели (Model Setting Notebook). Об этом говорилось в главе 2.

Параметры используются при косвенной адресации в GPSS. Когда номер желаемого GPSS объекта принимает значения параметра активной транзакции, то имя или номер идентификатора искомого GPSS объекта может быть замещен ссылкой на параметр транзакции. Более подробное описание этого способа приведено в разделе 3.4.

Прежде чем использовать параметр транзакции, следует задать его имя и значение. Это можно сделать с помощью блоков: ASSIGN, MARK, и TRANSFER SUB, SELECT, SPLIT и COUNT.

Приоритет – приоритет транзакции определяет преимущество транзакции на получение, когда оно в состоянии ожидания вместе с другими транзакциями. Транзакции с более высоким приоритетом получают преимущество. Приоритеты важны при возникновении очередей следующих типов: цепь текущих событий, цепь задержки в устройстве, цепь задержки на хранении. Цепь Будущих событий не нуждается в приоритетах. Об этих структурах будет рассказано в главе 9. Эффект приоритета заключается в том, что из трех транзакций, новой, в устройстве и транзакции в очереди будет выбрана заявка с большим приоритетом. Заявки внутри приоритета приходят первыми и обслуживаются первыми.

Время отсчета – абсолютное время на часах, когда заявка только пришла или же попала в определенный блок без операнда A.

Assembly Set – положительное целое число, содержащееся в каждой заявке. Используется для синхронизации заявки в блоках ASSEMBLE, GATHER, и MATCH. Когда заявка создается в блоке GENERATE, Assembly SET устанавливается равным номеру заявки. Если заявка создается с помощью блока SPLIT, Assembly Set принимает значение родительской заявки. Значение Assembly SET может быть изменено в блоке ADOPT.

Индикатор задержки – флаг, содержащийся в каждой заявке, устанавливается при отказе заявки в вхождении в блок и сбрасывается при вхождении в блок TRANSFER SIM. Используется блоком TRANSFER SIM для перенаправления заявок.

Индикатор маршрута – флаг, содержащийся в каждой заявке. Генерирует трассировочное сообщение каждый раз, когда заявка входит в блок. Индикатор маршрута устанавливается блоком TRACE и сбрасывается блоком UNTRACE.

Текущий блок – номер объекта блока, содержащего заявку.

Следующий блок – номер объекта блока, в который должна перейти заявка.

Последовательности – состояние заявки определяется в некоторой степени последовательностям, к которым она принадлежит. Глава 9 содержит более подробную информацию о последовательностях заявок. Заявка может находиться только в одном из следующих состояний:

- Активном – заявка с самым высоким приоритетом в последовательности текущих событий.
- Приостановленном – заявка, ожидающая в последовательности будущих или текущих событий, чтобы перейти в активное состояние.
- Пассивном – заявка перешедшая в режим ожидания в модели на последовательности задержек, пользовательской или последовательности задержек.
- Удаленном – заявка, удаленная из процесса моделирования.
- В добавлении ко всему, есть еще одно состояние, которое не исключается другими:
- В обработке – заявка, находящаяся на устройстве и входящая в одну или более последовательностей прерываний.

Активная заявка

В любой момент в процессе отдельной фазы моделирования, определенная заявка пытается попасть в новый GPSS блок. Такая заявка называется активной. Вообще, активная заявка перемещается настолько далеко, насколько это возможно в модели. Когда она уже не может двигаться дальше, выбирается следующая заявка, которая становится активной. Может быть только одна активная заявка. Иначе говоря, активная заявка – заявка с самым высоким приоритетом на последовательности текущих событий, когда работа последнего блока завершена. Более подробно это объясняется в главе 9.

Связанные SNAs

SNAs, связанные с заявками:

- A1 – Assembly Set. A1 возвращает значение Assembly Set активной заявки.
- MBEntnum – Проверить блок. MBEntnum возвращает 1, если существует заявка в блоке Entnum, у которой значение Assembly Set совпадает с активной заявкой. В противном случае возвращает 0. Этот SNA класс не стоит использовать в режиме отказа блоков GATE или TEST в тестовом состоянии. Вместо этого надо пользоваться блоками MATCH.
- MPPParameter – транзитное время параметра. Текущее абсолютное значение системного времени минус значение параметра заявки Parameter.
- M1 – транзитное время. M1 возвращает разность абсолютного значения системного времени и времени отсчета заявки.
- PParameter или *Parameter – значение параметра. Возвращает значение параметра Parameter активной заявки.
- PR – Приоритет заявки. Значение приоритета активной заявки.
- XN1 – Активная заявка. Номер активной заявки.

4.2 Блок объектов

Блок объектов GPSS – это базовый структурный элемент моделирования. Полезно представить GPSS модель с помощью диаграммы блоков. Это соединенные сети символов блоков, которые соотносятся с расположениями блока объектов в модели. Представление диаграммы можно увидеть в окне «Block Input» или «Blocks Window». Каждая заявка в модели содержит точно один блок, но большинство блоков могут содержать много заявок. Последовательность блоков, с которыми сталкиваются заявки, определяет природу и большую часть результата любого моделирования. Каждая заявка проходит сначала один блок, потом другой блок, пока не будет уничтожена или не закончится процесс

моделирования. Иногда заявке приходится задержаться в блоке, пока не появится возможность перейти в следующий блок. Это может произойти несколькими путями, детали, каким лучше, описаны в главе 9.

Каждый тип блока связан с действием, которое изменяет другие объекты моделирования. Эти действия описаны в главе 7, а контролируемые планировщиком заявок, описаны в главе 9. Сначала, блок определяет, может ли активная заявка войти. Если да, обновляется широкая статистика блоков, заявок и системы вообще. Затем блок определяет возникающее действие и выбирается следующий блок для заявки. Обычно, следующий последовательный блок (NSB) планируется. Постоянные блоки моделирования создаются в процессе начальной трансляции модели из выражений блоков в модель. Порядок блоков в транслированной модели такой же, как и в выражении блоков, определенных транслятором. Номера строк игнорируются. Выражения блоков GPSS можно создать в модели объекта, используя полноэкранный текстовый редактор через один или более диалогов создания блока, которые находятся в меню: **Edit / Insert Block**.

Выражения блока, посланные соответствующей модели, создают одноразовый временный блок в режиме, называемом «Ручное Моделирование». Такие интерактивные выражения дают команду активной заявке зайти в блок, а затем уничтожить его. Таким образом, Выражения блоков могут использоваться интерактивно для контроля моделирования. После того, как произошло действие блока, активная заявка продолжает свой старый путь в модели, пока не будет введено выражение TRANSFER или какой-нибудь другой блок не изменит путь заявки.

Это обсуждалось в главе 2.

Связанные SNAs:

SNAs связанные с блоками:

- NEntnum – количество записей в блоке. Общее количество заявок.
- WEntnum – Текущее количество заявок.

4.3 Объект «Устройство»

Устройство – объект с несколькими атрибутами, наиболее важным из которых является использование (ownership). В устройстве может находиться заявка, в этом случае оно говорит, что занято, или же в нем может не содержаться заявок вовсе, тогда оно сообщает, что свободно. В отличие от объекта памяти, заявка не может пройти мимо, не попав в устройство. Заявка попадает в устройство, успешно войдя в блоки SEIZE или PREEMPT. Блок PREEMPT может менять заявки в устройстве. Если заявка не может попасть в устройство, она переходит в режим ожидания в последовательности заявок устройств.

Устройство имеет несколько мест ожидания для заявок, которые ожидают нужного действия, чтобы попасть в устройство. Каждое устройство имеет последовательность задержек для заявок, последовательность ожидания решений для режима приоритетного прерывания и последовательность прерываний для предыдущих прерванных заявок. Заявки ожидают на последовательностях задержек, ожидания решений и прерываний сообщения «утверждено» от устройства. Каждая заявка, находящаяся в устройстве, должна, в конечном счете, получить новое состояние, войдя в блоки RELEASE или RETURN. После этого устройство освобождается.

Те заявки, которым не удалось попасть в блок SEIZE, переходят в режим ожидания и в порядке приоритета попадают на последовательность задержек устройства. После того, как устройство освобождается, выбирается следующая заявка в порядке приоритета.

Состояния объекта

Устройство, в котором находится заявка, пребывает в состоянии «занято». Если устройство не обрабатывает заявки, оно находится в состоянии «свободно».

Устройство может быть доступно и недоступно. Когда оно доступно, заявки принимаются на обработку нормально. Когда устройство недоступно, новая заявка не может быть принята на обработку. Блоки FAVAIL используются, чтобы перевести устройство в

состояние доступности, а блоки FUNAVAIL, напротив, в состоянии недоступности. Любые заявки, находящиеся в устройстве, когда оно переводится в состояние недоступности, следуют согласно командам блока FUNAVAIL.

Связанные блоки

Несколько GPSS блоков, которые могут быть использованы с устройствами:

- SEIZE – добавляет заявку в устройство.
- RELEASE – освобождает устройство.
- PREEMPT – замещение заявки в устройстве.
- RETURN – освобождает устройство.
- FAVAIL – переводит устройство в состоянии доступности
- UNAVAIL – переводит устройство в состояние недоступности

Связанные SNAs

SNAs, связанные с устройствами:

- *FEntnum* – устройство занято. Если устройство *Entnum* занято, возвращает 1, иначе 0.
- *FCEntnum* – количество обработанных заявок.
- *FIEntnum* – работа устройства *Entnum* прервана. Если обработка заявки прервана приоритетом другой заявки, возвращает 1, иначе 0.
- *FREntnum* – использование устройства. Доля времени, которое устройство *Entnum* было занято. Выражено в частях в тыс., поэтому возвращает реальное значение между 0 и 1000.
- *FTEntnum* – среднее обработки заявки. Среднее время, которое заявка находилась в устройстве *Entnum*.
- *FVEntnum* – устройство в режиме доступности. Возвращает 1, если устройство доступно, иначе 0.

4.4 Функции

Объекты функции GPSS используются для получения значения на основе какого-либо аргумента, например случайного числа. Вообще, любая SNA может быть использована в качестве аргумента. Функция определяется с помощью команды FUNCTION следующей за одним или более выражением Function Follower. Операнд А функции определяет аргумент, а операнд В определяет тип функции и число пар данных, фигурирующих в выражении Function Follower Statements. Это числа, имена и/или SNA, которые завершают определение объекта функции.

В большинстве случаев, удобнее использовать PLUS процедуры. Язык PLUS описан в главе 1. Тем не менее, GPSS функции приспособлены для использования, в качестве списка, основанного на функциях и эмпирических распределений вероятности. Также, GPSS функция может быть намного эффективнее в терминах компьютерного времени, чем эквивалентная ей процедура.

Существует 5 различных типов функций:

Тип С – функции с непрерывным значением. Представляют собой линейную интерполяцию. Принимают случайный аргумент.

Тип D – функция с дискретным значением. Каждое значение аргумента или массива должно быть определено и отделено.

Тип E – дискретная функция атрибута. Каждое значение аргумента или массива определено и должно быть вычислено.

Тип L – функция списка. Значение аргумента используется для определения позиции в списке, где находится возвращаемое значение.

Тип M – функция списка. Аргумент определяет позицию в списке, где находится SNA. Это SNA вычисляется и возвращается как результат работы функции.

Функция, используемая в операнде В блоков ADVANCE или GENERATE, называется «Модификатором Функции». Результат двойной точности с плавающей точкой функции умножается операндом А. Затем результат используется для приращения времени в блоке. Значения, указанные в объявлении функции, т.е. в выражении Function Follower, хранятся как значения двойной точности с плавающей точкой. Пределы этих значений могут быть от 15 десятичных чисел до 306. Линейная интерполяция, оценивая случайную функцию, выбирает случайное число от 0-999999, которое выдается генератором случайных чисел. Это случайное число умножается фактором интерполяции и добавляется к базовому интервалу.

Команда ФУНКЦИЯ обсуждается в главе 6.

Связанные SNA

SNA, применяемые к функциям:

FNEntnum – функция. Результат вычисления функции *Entnum*.

4.5 Объект «Логический переключатель» (Logicswitch)

Объект логический переключатель является самым простым из всех и имеет только два состояния: «Установлен» или «Сброшен». Существуют блоки, которые изменяют состояния переключателей и SNA, которые возвращают значение текущего состояния переключателя.

Связанные блоки

Блоки LOGIC устанавливают, сбрасывают или инвертируют состояния переключателя.

Связанные SNA

- LSEntnum – состояние переключателя. Возвращает 1, если переключатель в состоянии «Установлен», иначе возвращает 0.

4.6 Объект «Матрица»

Объект Матрица – это массив элементов, каждый из которых может принимать определенное значение. Размер матрицы ограничивается параметром Maximum Memory Request в настройках модели. Увеличив этот параметр, можно увеличить размер матрицы. Матрицы могут иметь максимальную размерность 6 и быть глобальными, постоянными, локальными или временными. Постоянная матрица определяется командой MATRIX в модели. Такие матрицы имеют глобальную область видимости и доступны из любого места модели. При создании матрицы, всем ее элементам присваивается значение 0. Также элементу обнуляются при команде CLEAR (без опции OFF). Элемент матрицы или вся матрица может быть переведена в состояние «Не определено» командой INITIAL. Временные матрицы определяются в области Temporary Matrix Declarations PLUS процедур. Они создаются при вызове процедуры, а при окончании ее работы уничтожаются. Они имеют локальную область видимости и могут быть использованы только в пределах процедуры, в которой объявлены. Элементы временной матрицы не инициализированы, так что нельзя использовать матрицу в выражениях, пока не будут заданы значения элементов.

Связанные блоки

- MSAVEVALUE – изменяет или устанавливает значение элемента матрицы

Связанные команды

- INITIAL – устанавливает значение элемента матрицы без активной заявки.

Связанные выражения PLUS

- Temporary Matrix Declarations - используется для создания временной матрицы в PLUS процедурах.
- Assignment Statements – может задавать значение любого элемента матрицы, определенной глобально или локально в процедуре.

Связанные SNA

SNA, связанные с матрицами:

- $MXEntnum(m, n)$ – значение матрицы. Возвращает значение элемента в строке m и столбце n матрицы $Entnum$. В матрицах размерностью больше 2 все другие индексы предполагаются равными 1. В отличие от класса MX SNA, Выражение PLUS может получить доступ к любому элементу матрицы более высокой размерности.

4.7 Объект «Очередь»

Не следует путать объект очередь с блоками QUEUE, которые являются экземплярами объектов блока. Объекты очереди используются главным образом для сбора статистики, например: текущий счет, количество записей, количество нулевых записей, максимальное количество, и счет времени. Блоки QUEUE и DEPART используются для обновления статистики, связанной с объектом очереди. Стандартная процедура «положить» блоки SEIZE, PREEMPT или ENTER между блоками QUEUE и DEPART. Тогда статистика очереди для устройств и памяти будет считаться автоматически. Некоторая базовая статистика объекта очередь может быть получена вызовом соответствующего SNA. Частота распределения может быть получена с помощью команды QTABLE.

Состояния объекта

Самым важным атрибутом очереди является количество заявок. Оно изменяется при входе в блоки QUEUE и DEPART. Это можно представить, как элементы, выстроенные в линию. Некоторые пункты статистики, связанные с содержанием, подсчитываются автоматически. Они доступны с помощью соответствующих SNA.

Связанные блоки

- QUEUE – увеличивает количество заявок в очереди.
- DEPART – уменьшает количество заявок в очереди.

Связанные SNAs

- $QEntnum$ – Текущее состояние. Количество заявок в очереди $Entnum$.
- $QAEntnum$ – Среднее количество заявок. Среднее по времени количество заявок в очереди $Entnum$.
- $QCEntnum$ – Общее количество заявок в очереди $Entnum$.
- $QMEntnum$ – Максимальное значение очереди. Максимальное количество заявок, которое может находиться в очереди $Entnum$. Верхний предел.
- $QTEntnum$ – Среднее время пребывания заявки в очереди $Entnum$.
- $QXEntnum$ – Среднее по времени количество заявок, исключаящее заявки с нулевым временем.
- $QZEntnum$ – Количество заявок в очереди $Entnum$ с нулевым временем.

4.8 Объекты памяти

Объект памяти содержит ячейки, в которых могут содержаться заявки. Объекты памяти могут использоваться как «накопители захвата» для контроля потока заявок в модели.

Когда заявка поступает в объект памяти, она занимает одну или более ячеек объекта памяти. Заявке может быть отказано, если блок памяти уже переполнен. Тогда заявка переходит в режим ожидания в последовательности задержек объекта памяти. И будет находиться в этом состоянии, пока другие заявки не освободят достаточно места в памяти, пройдя через блок LEAVE.

Количество свободного места в памяти может быть увеличено любой заявкой, даже той, что не поступала ранее в объект памяти. Тем не менее, если емкость объекта памяти будет больше, чем это заявлено по команде STORAGE, возникнет ошибка.

Когда заявка попадает в блок LEAVE и освобождает одну или более ячеек памяти, выбирается следующая заявка. Планирование ожидающих заявок осуществляется согласно дисциплине «первый пригодный с пропуском». Это означает, что каждая заявка на последовательности задержек проверяется на доступность в объекте памяти, начиная с

самого высокого приоритета. Как только подбирается соответствующая заявка, она изымается из последовательности и попадает в блок ENTER и помещается в СЕС согласно приоритету. Затем проверяется следующая заявка на последовательности задержек. Блоки ENTER и LEAVE используются для обновления статистики, связанной с объектом памяти, а также доступны некоторые SNA, с помощью которых можно получить значение статистики.

Объекты памяти должны быть определены командой STORAGE.

Состояния объекта

Объект памяти считается пустым, когда все ячейки свободны и полным, когда нет свободных ячеек. Объект может быть заполнен не полностью. Также, он может находиться в состоянии доступности или недоступности. Запросы к памяти возможны, только, если объект памяти находится в доступном состоянии.

Связанные блоки

- ENTER – запрос на заполнение памяти новой информацией.
- LEAVE – освобождает ячейки памяти.
- SAVAIL – переводит объект памяти в состояние доступности.
- SUNAVAIL – переводит объект памяти в состояние недоступности.

Связанные SNA

- *REntnum* – неиспользованные ячейки.
- *SEntnum* – использованное памяти. Возвращает количество использованных ячеек объекта памяти *Entnum*.
- *SAEntnum* – среднее использование памяти. Возвращает среднее по времени использование объекта памяти *Entnum*.
- *SCEntnum* – количество использованной памяти. Общее количество ячеек памяти, которые были использованы *Entnum*.
- *SEEntnum* – памяти пуста. Возвращает 1, если объект памяти *Entnum* пуст, иначе 0.
- *SFEntnum* – память заполнена. Возвращает 1, если объект памяти *Entnum* заполнен, иначе 0.
- *SREntnum* – использование памяти. Часть полного использования, представленного в среднем значении использования объекта памяти *Entnum*. Выражена в частях/тысячу и возвращает значение между 0 и 1000.
- *SMEntnum* – Максимальное значение использования объекта памяти *Entnum*. Верхний предел.
- *STEntnum* – Среднее время задержки в ячейке объекта памяти *Entnum*.
- *SVEntnum* – состояние объекта памяти. Возвращает 1, если объект памяти *Entnum* в состоянии доступности, иначе 0.

4.9 Объекты хранения

Объект хранения ассоциируется с переменной, которая может принимать любое значение. Значение может быть установлено или изменено блоками и может быть получено с помощью SNA класса X.

При создании хранилища, оно принимает значение 0, а также это же значение устанавливается командой CLEAR (без опции OFF). Хранилище может быть установлено в неопределенное состояние командой INITIAL.

Связанные блоки

- SAVEVALUE – устанавливает или изменяет значение объекта хранения.

Связанные команды

- INITIAL – устанавливает значение объекта хранения без активной заявки.
- Связанные SNA
- *XEntnum* – возвращает значение объекта хранения.

4.10 Объект «Таблица»

Таблица – это набор целых чисел, используемый для сбора данных для гистограммы. Каждое число представляет определенную частоту в гистограмме.

Таблица определяется командой TABLE.

Связанные блоки

- TABULATE – обновление данных гистограммы, собранных в таблице.

Связанные SNA

- TBEntnum – среднее количество записей в таблице Entnum.
- TCEntnum – количество записей в таблице Entnum.
- TDEntnum – стандартное отклонение в таблице Entnum.

4.11 Пользовательские потоки

Пользовательские потоки содержат специальные последовательности заявок, называемых пользовательской последовательностью, манипулировать которой можно с помощью блоков LINK и UNLINK. Пользовательские потоки применяются для моделирования сложных алгоритмов планирования и распределения очереди. Они обеспечивают более закрытое управление очередями заявок, чем просто доступ к последовательности задержек.

Флаг, называемый «индикатор соединения» есть в каждом пользовательском потоке. Он полезен при использовании пользовательской последовательности для распределения очереди заявок на ресурс. Более детально с объяснением блоков LINK и UNLINK это описано в главе 7.

Связанные блоки

- LINK – условно помещает заявку на пользовательскую последовательность.
- UNLINK – удаляет заявку с пользовательской последовательности.

Связанные SNA

- CAEntnum – среднее значение пользовательской последовательности. Среднее по времени количество потоков заявок для пользовательской последовательности Entnum.
- CCEntnum – Общее количество записей. Общее количество заявок, прошедших через пользовательскую последовательность Entnum.
- CHEntnum – текущее значение последовательности. Текущее количество заявок на пользовательской последовательности.
- CMEntnum – максимальное количество заявок, которое может находиться на пользовательской последовательности Entnum.
- CTEntnum – среднее время задержки заявки в пользовательской последовательности Entnum.

4.12 Переменные

Переменные – это комплексные выражения, которые вычисляются мгновенно по запросу. Все переменные могут быть определены выражениями, включающие константы, с помощью SNA, функциями арифметических библиотек, арифметическими или логическими операторами. Эти выражения описаны в главе 8.

Переменная определяется командами VARIABLE, FVARIABLE или BVARIABLE.

Команда VARIABLE создает арифметическую переменную GPSS, которая, когда вызывается SNA, вычисляется выражение и возвращает результат. В режиме совместимости GPSS/PC промежуточные результаты отбрасываются.

Команда FVARIABLE создает переменную с «плавающей точкой».

Команда BVARIABLE создает логическую переменную, которая, вычисляя выражение, принимает значение 1, если результат ненулевой, в противном случае принимает значение 0.

Связанные SNA

BVEntnum – результат вычисления логической переменной *Entnum*.

VEntnum – результат вычислений арифметической переменной или переменной с плавающей точкой.

4.13 Группы чисел

Группа чисел – это набор числовых значений. Группа чисел используется для записи событий или отображения состояния процесса, который моделируется.

Операции группы чисел выполняются быстрее с целыми числами, чем с вещественными.

Связанные блоки

- JOIN – помещает значение в группу чисел.
- REMOVE – Удаляет значение из групп чисел.
- EXAMINE – проверяет корректность значений группы чисел.

Связанные SNA

- *GNEntnum* – возвращает количество значений группы чисел *Entnum*.

4.14 Группа заявок

Группа заявок – это набор заявок. Можно использовать неограниченное число групп заявок, и можно создавать неограниченное число групп для одной заявки. Группы заявок полезно использовать для классификации заявок и организации доступа к ним. Активная заявка может проверить параметры заявки всех членов группы заявок.

Связанные блоки

- JOIN – помещает заявку в групп заявок.
- REMOVE – удаляет сообщество членов группы заявок.
- EXAMINE – проверяет члены группы заявок.
- SCAN – тестирует и/или изменяет члены группы заявок.
- ALTER - тестирует и/или изменяет члены группы заявок.

Связанные SNA

- *GTEntnum* – Возвращает количество членов группы заявок *Entnum*.

4.15 Генераторы случайных чисел

Потоки случайных чисел в GPSS World генерируются мультипликативным когерентным алгоритмом с максимальным периодом 32. Период равен 2³¹-2 и не включает 0. Можно использовать любое количество генераторов случайных чисел, не объявляя их. Начальное число генератора совпадает с номером объекта генератора случайных чисел. Но, с помощью выражения *RMULTT* можно контролировать только генераторы, пронумерованные от 1 до 7.

Алгоритм генерации псевдослучайных чисел в GPSS World основан на когерентном мультипликативном алгоритме Лехмера с максимальным периодом. Алгоритм выдает числа в интервале от 0, до 2,147,483,647 и генерируется 2,147,483,646 уникальных неповторяющихся случайных чисел. Существует также дополнительный способ перестановок. Класс SNA RN возвращает 0-999 включительно и вычисления функций использующих числа от 0-999999 включительно.

Важные атрибуты генераторов

Начальное число. Если не было изменено выражением *RMULT*, то начальное число равно номеру объекта генератора. Например, генератор RN2 стартует с начальным значением 2.

Использование системы. GPSS World использует генераторы чисел в определении интервалов времени в фрагментном режиме блоков TRANSFER, а также для выбора случайных чисел для блоков GENERATE и ADVANCE. Можно указать, какой генератора будет использоваться для генерации конкретного случайного числа, на странице «Random» в настройках модели.

Значения SNA. При использовании SNA возвращается значение в пределах 0-999. Можно получить и большие значения, используя выражения типа 1000#RN2+RN2 при определении переменной. Новые значения могут быть получены с помощью класса V SNA. Более удобно использовать их в процедурах. Более подробно это описано в главе 8. Значения интерполяции. Дробные значения от 0 до 999999 выдаваемые генератором при использовании интерполяции при непрерывной функции.

Связанные SNA

- *RNEntnum* – случайное число. Возвращает случайное число 0-999, полученное от генератора *Entnum*.

4.16 Потоки данных

Поток данных – это последовательность текстовых строк. Каждый поток определяется своим уникальным номером, несколько потоков могут быть обработаны одновременно в процессе моделирования. Номера потоков – это произвольные положительные целые числа, определяемые пользователем.

Потоки данных могут быть использованы для чтения или записи в файл или для поддержания прямого доступа к памяти компьютера. Операции открытия, закрытия, чтения, записи и поиска связаны между собой и находятся в процедурах PLUS библиотеки. Блоки обозначаются только заглавными буквами, а процедуры преобразовывают только первую букву. Можно выполнять сложные операции ввода-вывода с помощью PLUS процедур или, используя объекты GPSS более высокого уровня, а также, можно совместить эти два режима. Блоки описаны в главе 7, а процедуры PLUS в главе 8. Для простоты, мы рассмотрим только блоки.

Базовый элемент потока данных – это текстовая строка, состоящая из печатных символов, включая пробелы. Для работы со строками можно использовать встроенную библиотеку процедур для строк. Если при операциях READ или WRITE попадают непечатные символы, вся строка обрезается.

Элементарные операции, применяемые к потокам данных, такие как READ, WRITE и SEEK, применяются к каждой отдельной строке. При чтении, данные из файла очищаются от специальных символов CR или LF, перед тем как поступить на обработку в модель. Последовательности символов CR и LF добавляются автоматически к любой текстовой строке, написанной в редакторе моделирования.

Существует два типа потока потоков данных:

1. Ввод/вывод (I/O) для доступа к файлам,
2. В потоках памяти для проверки и для прямого доступа в внутренним данным.

У каждого потока данных есть свойство – «Текущая строка». Это относительный индекс, указатель на следующую строку, с которой будут производиться операции. Например, чтение происходит, когда значение текущей позиции равно 1, то будет получено значение первой строки потока данных. Операция SEEK используется для смены текущей позиции в потоках I/O и потоках памяти.

Запись в файл или в память происходит в двух режимах: Добавление, замещение. При этом, свойство текущей позиции воспринимается немного по-разному.

Для обработки потока данных предусмотрены 5 блоков: OPEN и CLOSE, которые начинают или завершают процесс обработки. READ и WRITE – добавляют или получают текстовые строки из потока данных. SEEK – используется для установки текущей позиции.

4.16.1 Операции потока данных

Каждая из следующих операций может быть представлена как GPSS блок или как процедура PLUS библиотеки. Эти два метода эквивалентны и взаимозаменяемы,

исключая режим перезаписи, который доступен в блоке WRITE, но не содержится в процедуре Write() библиотеке. Здесь описано преимущественно работа с блоками.

OPEN

Блок инициализирует работу с потоком данных и устанавливает текущую позицию в 1. Необходимо определить тип потока, указав параметр A. Этот параметр является строковой константой, указывающий на тип потока данных. Необходимо помнить, что строковые константы – это PLUS выражения. Каждое PLUS выражение, должно быть взято в скобки. Потоку ввода вывода определяется указанием на файл, а поток памяти – пустой строкой.

Например, в блоке OPEN можно указать:

OPEN ("MYFILE.TXT") – для I/O потока

OPEN (« ») – для потока памяти

Если при создании файлового потока данных указан файл без пути к нему, будет выбрана предполагаемая директория расположения файла. Если указанный файл найден, он загружается в виртуальную память в процессе обработки блока OPEN. В противном случае система создает файл с указанным именем и обработка продолжается.

Можно вызвать операция открытия с помощью процедуры PLUS Open(). После завершения операции, вся информация хранится в объекте моделирования, пока поток не будет закрыт. Все изменения вступят в силу, только после вызова блока CLOSE или процедуры Close().

CLOSE

Блок CLOSE освобождает ресурсы, используемые потоком данных, а также возвращает код ошибки. Для IO потоков, блок записывает данные из виртуальной памяти на диск.

Можно вызвать операцию закрытия с помощью PLUS процедуры Close().

READ

Блок READ получает текстовую строку текущей позиции и увеличивает значение текущей позиции на 1. Если следующая строка доступна, активная заявка идет по альтернативному направлению, но не запоминает внутренний код ошибки.

Можно вызвать операцию READ с помощью PLUS процедуры Read().

WRITE

Блок WRITE передает текстовую строку потоку данных.

Операции зависит от того, в каком режиме будет производиться действие. Если параметр D операции WRITE включен, используется режим вставки. Если параметр D отключен, используется режим замены. По умолчанию D включен.

Можно вызывать эту операцию с помощью PLUS процедуры Write(), однако процедура не поддерживает режим замены.

Режим вставки

Режим по умолчанию для операции WRITE.

Действие:

1. Перемещает все текстовые строки вниз на одну позицию.
2. Если текущая позиция имеет значение последней строки, устанавливает его в значение после последней строки в потоке данных.
3. Помещает новую строку в текущую позицию.
4. Увеличивает значение текущей позиции.

Режим замены (только для блока WRITE)

Действие:

1. Если значение текущей позиции находится сразу за последней строкой, стирает все промежуточные строки.
2. Удаляет строку в текущей позиции.
3. Помещает строку в текущую позицию.
4. Увеличивает текущую позицию на 1.

SEEK

Блок SEEK устанавливает текущую позицию. Текущая позиция не может быть меньше 1. При попытке установить значение 0 будет установлена 1.

Можно вызывать операцию seek с помощью PLUS процедуры Seek(), которая возвращает предыдущее значение позиции, которое было до вызова процедуры. Это значение недоступно, когда используется блок SEEK.

4.16.2 Использование потоков данных

Тип потока данных выбирается указанием параметра A при открытии.

I/O потоки

Создается при указании файла в параметре A. Если не указывается полный путь к файлу, используется директория текущей модели. При операции открытия, весь файл считывается в память компьютера. При закрытии потока, файл записывается на диск. Запись на диск происходит только по операции CLOSE. Если при открытии, указанный файл не найдет, ошибка не возникает. В этом случае, GPSS World считает, что файл должен быть создан. Если необходимо проверить, существует ли файл, следует использовать READ, перед продолжением работы.

Ниже представлен простой пример использования I/O потока. Это маленькая часть модели, которая открывает файл, считывает первую строку, вызывает SEEK, переходит на 20 позицию и записывает текстовую строку, затем закрывает файл.

*

* Read and Modify MYFILE.TXT

*

```
GENERATE 1,,1
OPEN   ("MYFILE.TXT"),1,Done   ;Copy the file to memory
READ   Text_Parm,1,Done       ;Place text line in parm
SAVEVALUE Opening_Line,P$Text_Parm ;Text line to safeplace
SEEK    20,1                  ;Access text line 20
WRITE  ("New Line 20"),1,Done  ;Replace the line
Done   CLOSE  Error_Parm,1      ;Copy the file to disk
SAVEVALUE File_Error,P$Error_Parm ;Put in Standard Report
TERMINATE 1
```

Потоки памяти

Если в параметре A операции OPEN используется пустая строка, создается поток памяти. В этом случае, все строки сохраняются в памяти. При закрытии потока, все строки удаляются.

Все операции потока памяти работают также, как и в потоках I/O с той лишь разницей, что при открытии он не содержит данных, а при закрытии не сохраняет их. Поток должен быть сначала загружен строками, прежде чем с ним можно будет работать.

Потоки памяти предоставляют прямой доступ к данным с помощью SEEK и позволяют проверить данные без доступа к файлу.

4.16.3 Проверка на ошибки

Можно обрабатывать ошибки потоков самостоятельно или же останавливать модель, когда возникает хоть одна ошибка. По умолчанию, при возникновении ошибки, остановка не происходит. Это устанавливается в настройках модели. Чтобы изменить эту опцию, выберите в меню **Edit / Settings** и установите/снимите галочку **I/O Stream Error Stops**.

При возникновении ошибок и других неожиданных случаев запоминается код ошибки потока данных и, если необходимо, направляет активную заявку по альтернативному пути. С помощью блока CLOSE можно получить код ошибки потока. Запоминается только первый ненулевой код ошибки. Остальные отбрасываются.

Альтернативный путь может быть указан параметром С в блоках OPEN, CLOSE, READ и WRITE. Активная заявка пойдет по альтернативному пути, вместо того, чтобы попасть с следующий последовательный блок, если возникла ошибка. Если эти параметры не указаны, по существу происходит игнорирование ошибок. Вообще, необходимо использовать альтернативные пути, чтобы направить активную заявку в блок CLOSE, которые возвратит код ошибки. Далее можно будет запомнить этот код и прекратить моделирование.

Когда активная заявка входит в блок CLOSE, код ошибки помещается в параметр заявки. И, если код ошибки ненулевой и указан альтернативный путь в параметре С, активная заявка начинает двигаться по нему.

Конец данных

Когда заявка входит в блок READ, но в текущей позиции отсутствует строка, возникает особый случай. В общем случае, это означает, что весь файл прочитан. При этом заявка отправляется по альтернативному пути, но код ошибки не сохраняется.

Коды ошибок

0 – Нет ошибок.

10 – Ошибка открытия. Слишком длинное имя файла. Поток данных не создан.

11 – Ошибка открытия. Ошибка чтения файла. Поток не создан.

12 – Ошибка открытия. Отказано в доступе к памяти при чтении файла. Поток не создан.

21 – Ошибка чтения. Отказано в доступе к памяти при чтении.

22 – Ошибка чтения. Поток не открыт.

31 – Ошибка записи. Отказано в доступе к памяти при записи.

32 – Ошибка записи. Поток не открыт.

41 – Ошибка закрытия. Ошибка записи на диск.

43 – Ошибка закрытия. Поток не открыт.

51 – Ошибка перемещения. Поток не открыт.

4.17 Непрерывное моделирование

GPSS World может автоматически интегрировать системы обычных дифференциальных уравнений. Интегрирование пользовательских переменных происходит чрезвычайно быстро. Необходимо всего лишь определить переменную и записать одно или более INTEGRATE выражений. Интегрирование производится автоматически модифицированием переменной 5 шагами по методу Рунга-Кутты-Фелберга, RK4(5).

Можно моделировать любой набор простых систем дифференциальных уравнений. Можно строить графики состояний переменных.

4.17.1 Как начать интегрирование

Необходимо сделать две вещи для установки автоматического интегрирования пользовательских переменных. Необходимо задать команду INTEGRATE и определить начальное значение переменной.

Допустим, что имеет простую систему дифференциальных уравнений вида:

$$y' = f(-)$$

где $f(-)$ – выражение представляющее собой системное время, т.е. Систему атрибутов AC1, и другие пользовательские переменные. Производная по времени.

Сначала, заключим производную в круглые скобки и допишем команду INTEGRATE:

$$Y_INTEGRATE (f(-))$$

Далее, убедимся, то переменная Y имеет начальное значение, например:

Y_ EQU 100.3

Теперь, когда процесс моделирования перейдет к следующему моменту времени, мы увидим, что переменная Y изменить свое значение.

Начиная с некоторой последовательности букв, имена пересекаются с классами SNA, здесь мы будем подчеркивать их, чтобы быть уверенными в их уникальности. Все остальные пользовательские переменные, включенные в производную, должны быть обязательно инициализированы. Можно использовать PLUS выражения для задания значений переменным.

Интегрирование автоматически начинается в активном или «включенном» состоянии. Однако, можно включать или выключать интегрирование в процессе моделирования блоками INTEGRATION. Этот способ описан в главе 7.

4.17.2 Базовые концепции

Производная показывает, как быстро изменяется значение переменной. Например, если запасы пополняются на 2 единицы в час, а моделирование происходит единица в секунду, все что нужно сделать, это добавить:

Inventory INTEGRATE (2.0 / 3600)

и

Inventory EQU 100

Степень изменения (производная) позволяет GPSS World автоматически увеличивать значение запаса в процессе моделирования. Когда происходит операция продажи, можно просто уменьшить значение запаса используя процедуру PLUS.

Интегрирование занимает больше компьютерного времени, чем более простые выражения. Например, быстрее будет вычислено значение как функции от времени при передаче ей текущего времени, чем интегрирование пользовательской переменной.

Интегрирование применяется тогда, когда нет другого решения дифференциального уравнения.

4.17.3 Пределы

Интегрированные пользовательские переменные создают заявки, когда пересекают пределы. Это упрощает использование непрерывно моделируемых значений для условий возникновения событий.

Каждая команда INTEGRATE может иметь нулевые, одну или две числовых предела. Параметры В и С используются для определения 1 предела и/или параметры D и E используются для определения 2 предела. Первый параметр пары определяет значение предела, второй показывает блок, который получит сгенерированная заявка.

В процессе интегрирования, если значение интегрированной переменной пересекает значение предела такого же направления, создается новая заявка. Ей выдается нулевой приоритет, и она должна войти в блок, связанный с пределом в команде INTEGRATE. Время пребывания заявки в модели определяется линейной интерполяцией. С целью повышения точности, шаг интеграции уменьшается, когда приближается предел.

Пределы могут быть константами, заключенными в скобки выражениями, или даже содержать вызовы процедур. К тому же, заявка, сгенерированная пересечением предела, может передвигать предел.

Оба предела ведут себя одинаково. Нет необходимости определять один как верхний, другой как нижний. Пересечение предела (в любом направлении), вызывает событие прибытия заявки. Если направление пересечения значимо для модели, необходимо отслеживать состояние интегрированной переменной или проверять направление пересечения, когда происходит соответствующее событие.

4.17.4 Уравнения более высокого порядка

Необходимо уменьшать порядок дифференциальных уравнений высокого порядка. Перевод вводим дифференциальных уравнений высокого порядка в GPSS World, необходимо переписать их как систему уравнений первого порядка. Это относительно просто, и требует того, чтобы для каждой промежуточной производной была представлена своя переменная.

Например, если у нас есть:

$$25 y''' - 6 y' + y = 0.$$

Допустим $u = y'$

$$v = u' = y''$$

Для более высоких порядков продолжим подставлять переменные.

Затем, подставляя в оригинальное уравнение и используя только первый порядок производной, имеем:

$$25 v' - 6 u + y = 0.$$

Теперь, у нас есть система уравнений:

$$y' = u$$

$$u' = v$$

$$v' = 6/25 u - 1/25 y$$

В GPSS World можно использовать следующие выражения:

Y_ INTEGRATE U_

U_ INTEGRATE V_

V_ INTEGRATE ((6/25) # U_ - (1/25) # Y_)

Для этого необходимо добавить EQU выражения, которые инициализируют переменные U_, V_, Y_.

4.17.5 Непрерывное моделирование

Если вы хотите проводить исключительно непрерывное моделирование, необходимо создать условия остановки вхождением в блок TERMINATE. Поэтому включают такой сегмент GPSS как:

GENERATE End_Time

TERMINATE 1

В моделях, которые начинаются с

START 1

4.17.6 Фазы

Моделирование проходит поочередно в непрерывной и дискретной фазах. В любой момент времени, где события запланированы, моделирование проходит в дискретной фазе. Часы моделирования не изменяются без дискретной фазы. Между моментами времени моделирование проходит в непрерывной фазе, в течении которой интегрирование происходит маленьким приращением времени, называемых минишагом. График интегрированной переменной отображает промежуточные значения в конце каждого минишага.

Когда пересечение предела генерирует заявку, моделирование переходит в дискретную фазу. Таким образом, непрерывная и дискретная фазы тесно связаны между собой. Значения пользовательских переменных могут быть установлены в дискретной фазе, даже если они были интегрированы. Это можно сделать, используя команду EQU или PLUS выражения. Если вы хотите, чтобы такие изменения произошли в процессе

моделирования, то необходимо объявить PLUS процедуру, которая сделает такие изменения. Например:

PROCEDURE SetPop(Pop_Level)Foxes = Pop_Level ;

Можно переопределить переменную FOXES, вызвав PLUS блок, например:

PLUS (SetPop(200))

или используя выражение в круглых скобках, которое вызывает SetPop() другого типа блока.

4.17.7 Ошибки интегрирования

Настройка модели, называемая «устойчивость интегрирования» используется для ограничения локальное отсечение ошибок интегрирования. Эта настройка применяется для всех интегрирований, выполняемых в процессе моделирования. Если сделать устойчивость меньше, интегрирование будет происходить дольше, но будет более точной. Оно устанавливается в настройках модели.

Выберите **Edit / Settings**

Затем, выберите страницу **Simulate**. И укажите желаемое значение в поле «Устойчивость интегрирования». Значение по умолчанию 10^{-6} .

4.17.8 Связанные команды

- INTEGRATE – устанавливает автоматическое интегрирование.

Связанные блоки

- INTEGRATION - устанавливает интегрирование переменных: ON или OFF.

Глава 5 – Окна GPSS World

Этот раздел рассматривает все главные окна GPSS World. Главные операции GPSS World, включая меню и диалоги, описывается в главе 2.



Рисунок 5.1 – Главное окно

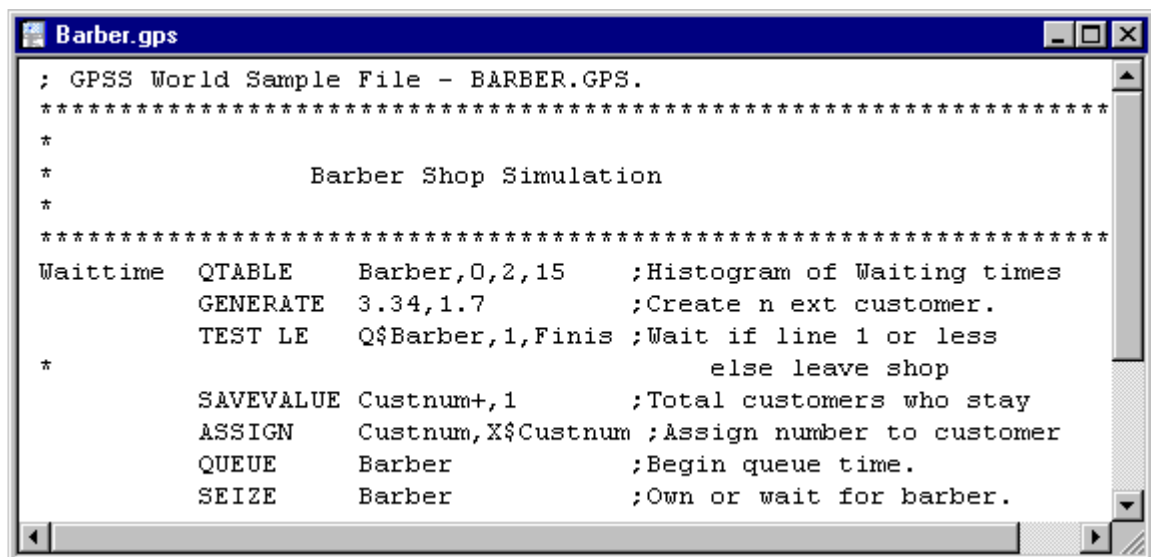


Рисунок 5.2 –Текстовое окно

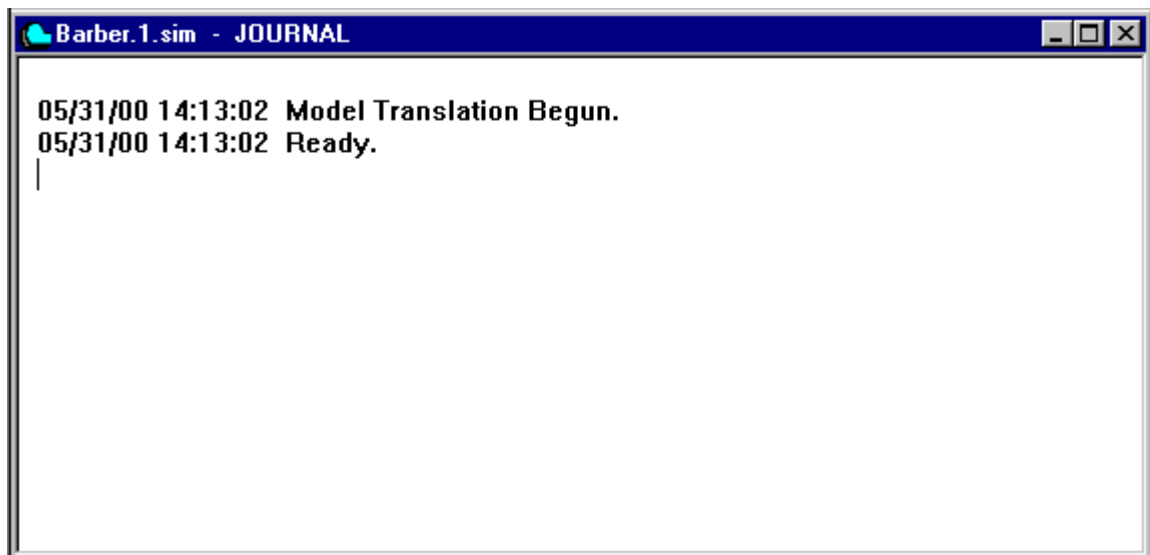


Рисунок 5.3 – Журнал моделирования

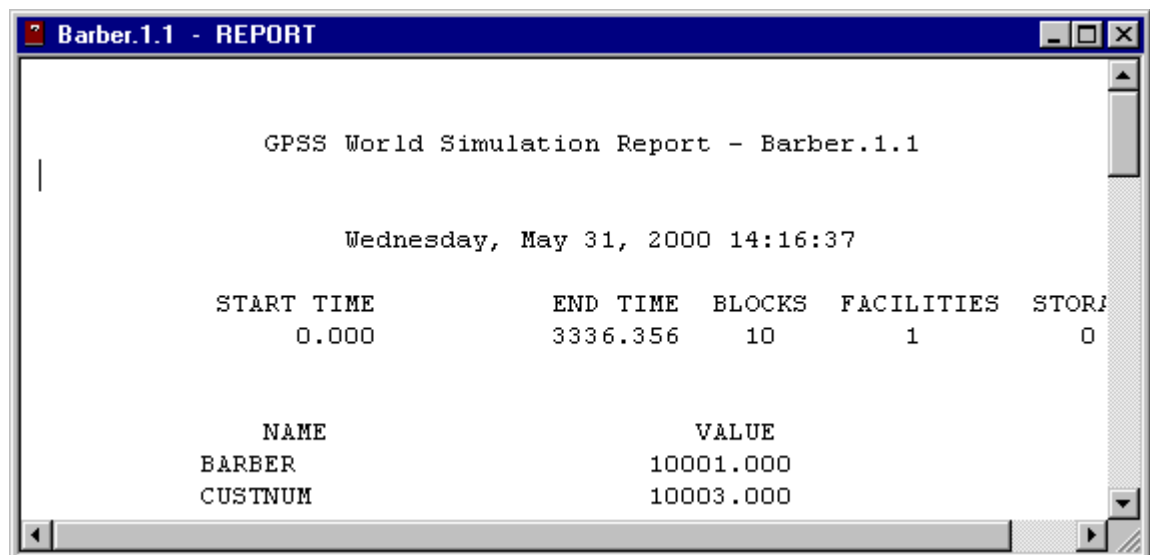


Рисунок 5.4 – Окно отчета

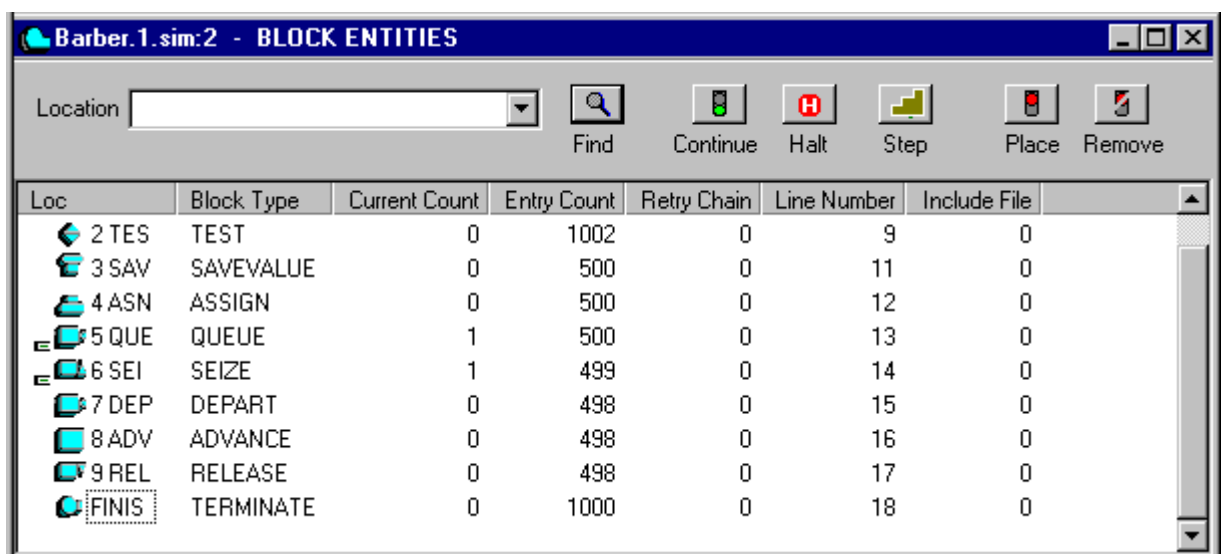


Рисунок 5.5 – Подробный вид блоков

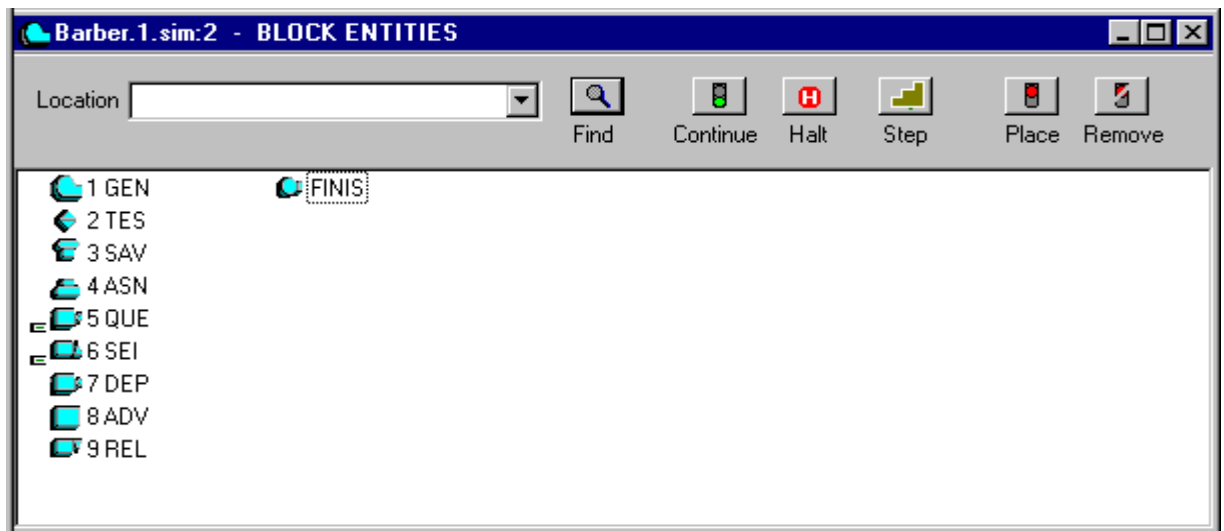


Рисунок 5.6 – Просто обзор блоков

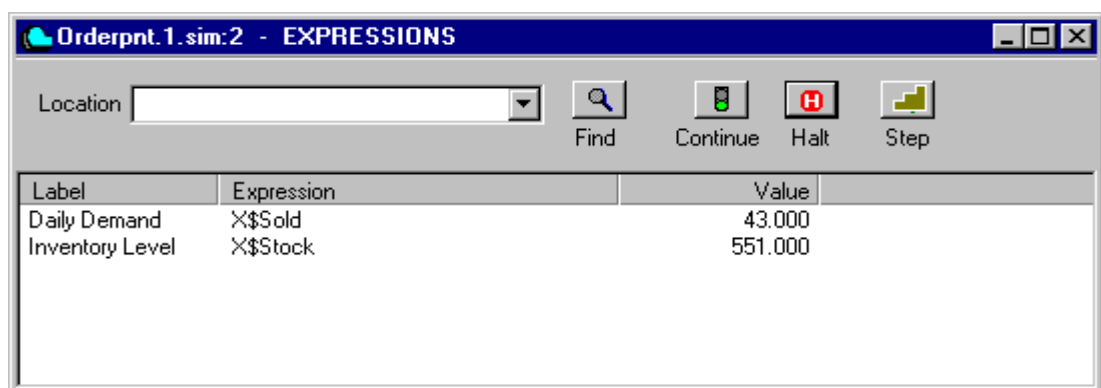


Рисунок 5.7 – Окно выражений

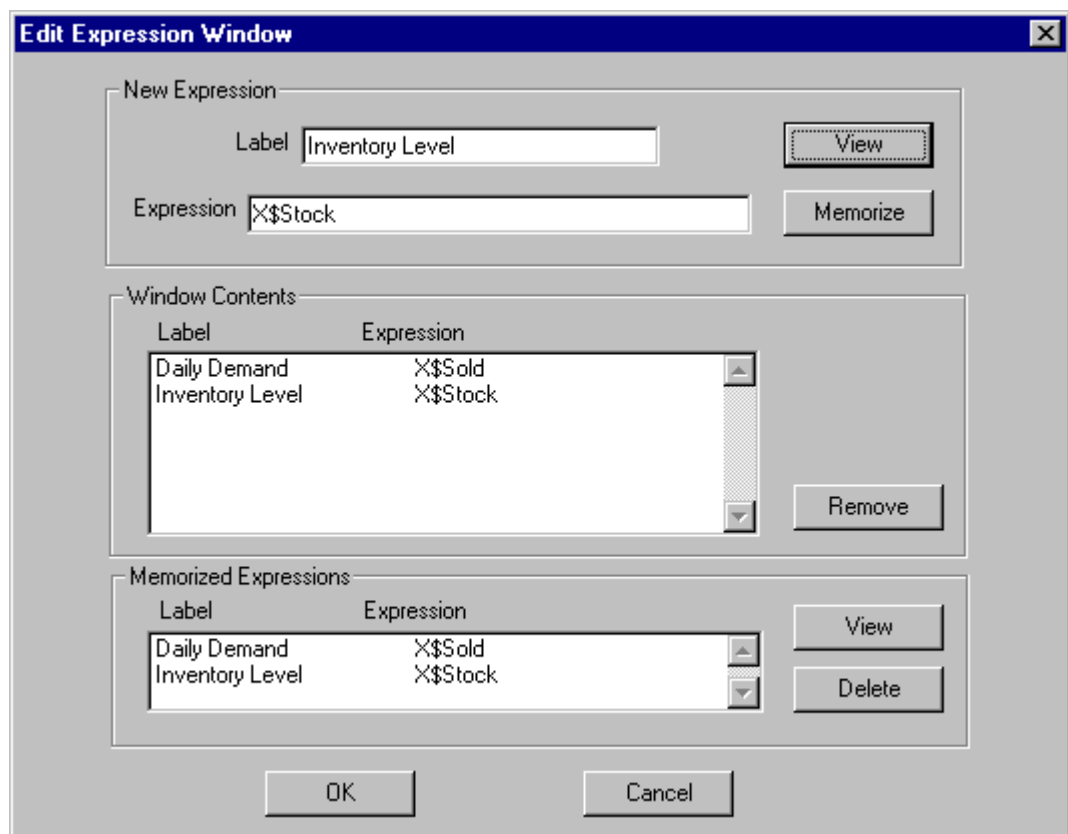


Рисунок 5.8 – Диалог редактирования выражений

Location Find Continue Halt Step

Facility	Utilization	Delay Chain	Acquisitions	Available	Ave. Time	Owner XN	Retry ...	Pending...
1	0.028	0	9265	+	20.000	0	0	0
2	0.018	0	5997	+	20.000	0	0	0
3	0.010	0	3268	+	20.000	0	0	0
4	0.028	0	9264	+	20.000	0	0	0
5	0.015	0	5106	+	20.000	0	0	0
6	0.012	0	4158	+	20.000	0	0	0
7	0.019	0	6457	+	20.000	0	0	0

Рисунок 5.9 – Детальная информация об устройствах

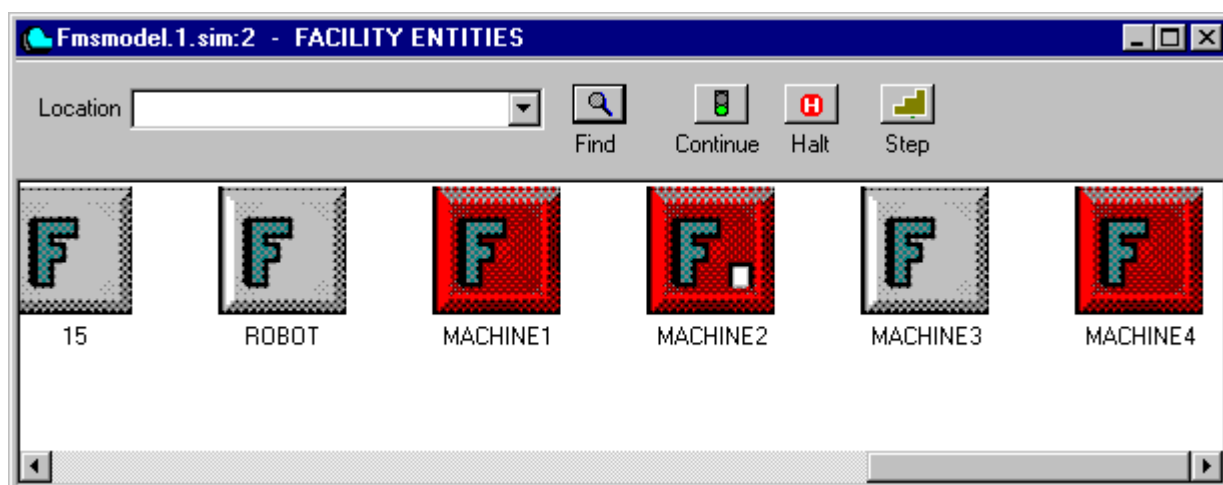


Рисунок 5.10 – Краткая информация об устройствах

Location Find Continue Halt Step

Logicswitch	Value	Retry Chain
MAINSWITCH	1	0

Рисунок 5.11 – Детальный обзор логических переключателей

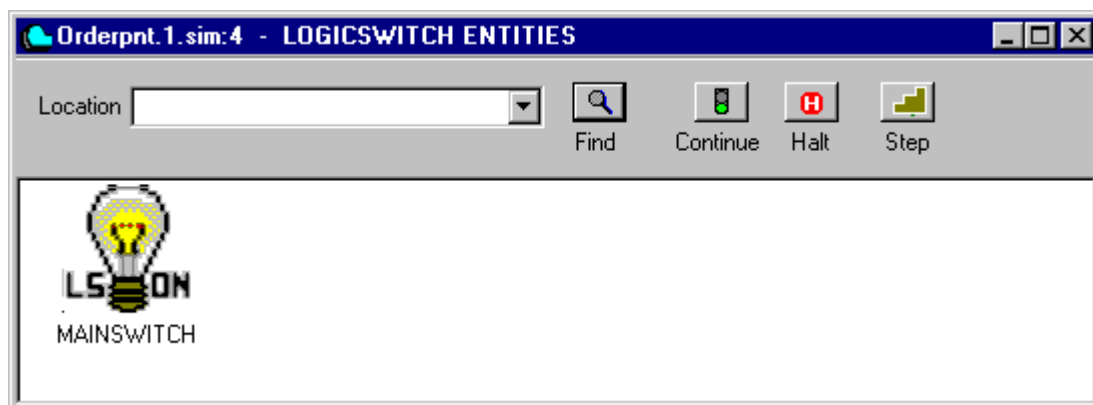


Рисунок 5.12 – Краткий обзор логических переключателей

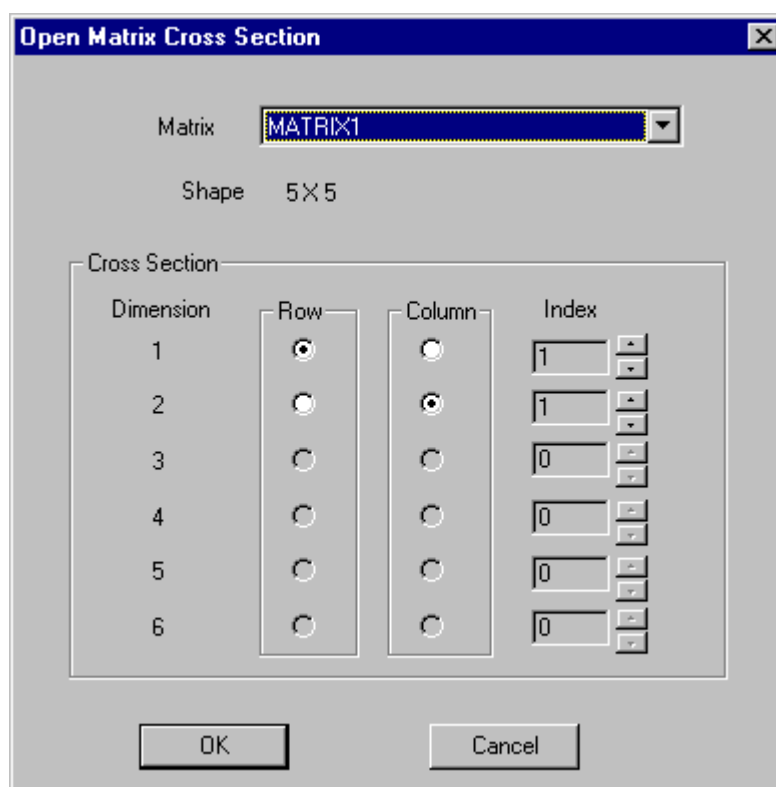


Рисунок 5.13 – Диалог матриц

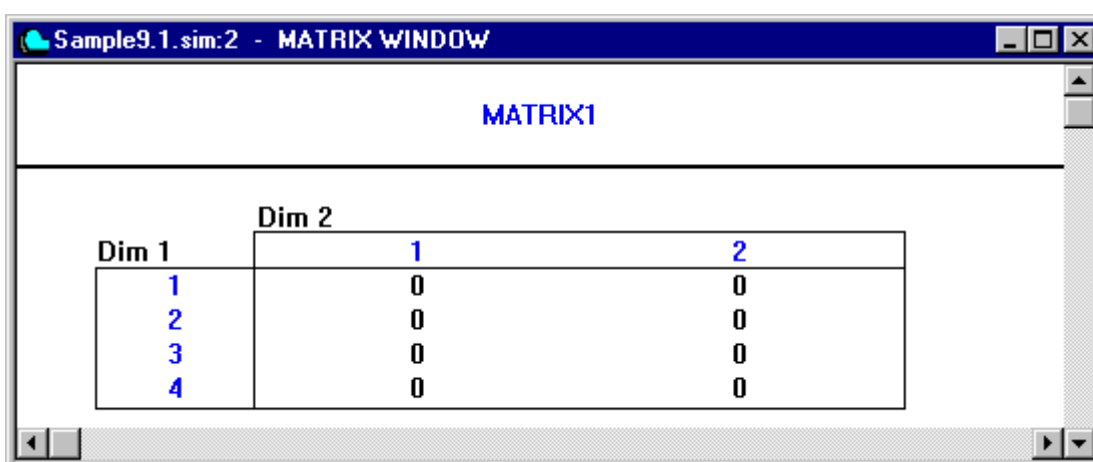


Рисунок 5.14 – Матрицы

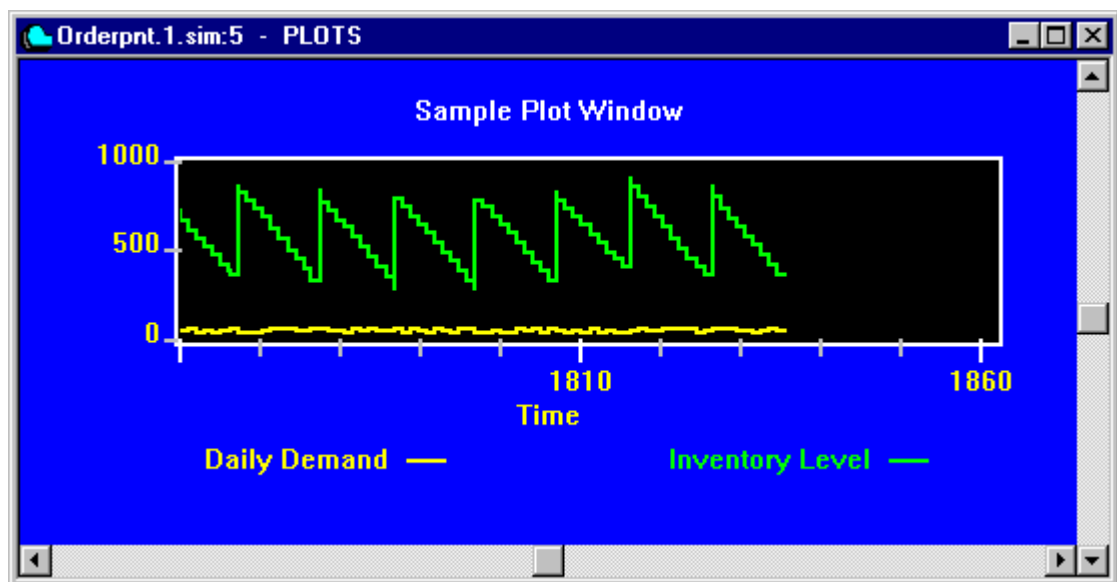


Рисунок 5.15 – Окно графиков

The figure shows the "Edit Plot Window" dialog box. It is organized into three main sections:

- New Expression:** Contains a "Label" field with the text "Daily Demand" and an "Expression" field with the text "X\$Sold". There are "Plot" and "Memorize" buttons to the right.
- Window Contents:** Contains a "Title" field, a "Time Range" field with the value "100.000000", a "Min Value" field with "0.000000", and a "Max Value" field with "100.000000". Below these is a table:

Label	Expression
Daily Demand	X\$Sold
Inventory Level	X\$Stock

 A "Remove" button is located to the right of this table.
- Memorized Expressions:** Contains a table:

Label	Expression
Daily Demand	X\$Sold
Inventory Level	X\$Stock

 There are "Plot" and "Delete" buttons to the right of this table.

At the bottom of the dialog are "OK" and "Cancel" buttons.

Рисунок 5.16 – Диалог редактирования графиков

Fmsmodel.1.sim:4 - QUEUE ENTITIES

Location Find Continue Halt Step

Queue Entity	Current Cont...	Entry Count	Zero Entry Count	Maximum Content	Average Content	Average Time (+0)	Average Time (-0)	Retry Chair
ARRIVAL	0	9265	0	1	0.090	64.542	64.542	C
ONE	0	3268	1409	4	0.338	690.541	1213.926	C
WIPONE	0	3267	0	1	0.032	64.600	64.600	C
TWO	2	4158	909	9	1.188	1906.920	2440.434	C
WIPTWO	1	4156	0	1	0.041	65.336	65.336	C
THREE	0	2804	1381	6	0.292	694.031	1367.578	C

Рисунок 5.17 – Детальный обзор очередей

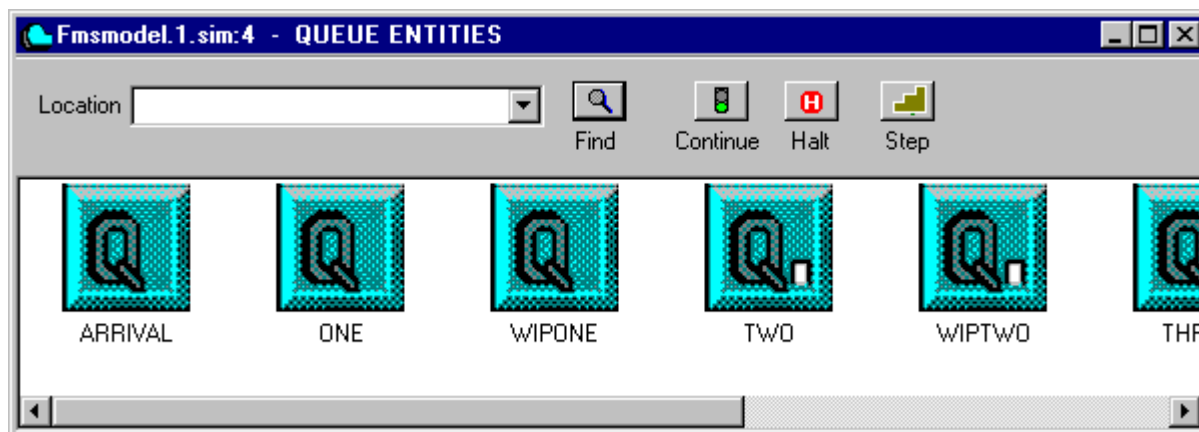


Рисунок 5.18 – Краткий обзор очередей

Fmsmodel.1.sim:5 - SAVEVALUE ENTITIES

Location Find Continue Halt Step

Savevalue	Value	Retry Chain
1	520.000	0
2	633.000	0
3	548.000	0
4	621.000	0
5	634.000	0
6	574.000	0
7	570.000	0

Рисунок 5.19 – Детальный обзор хранилищ

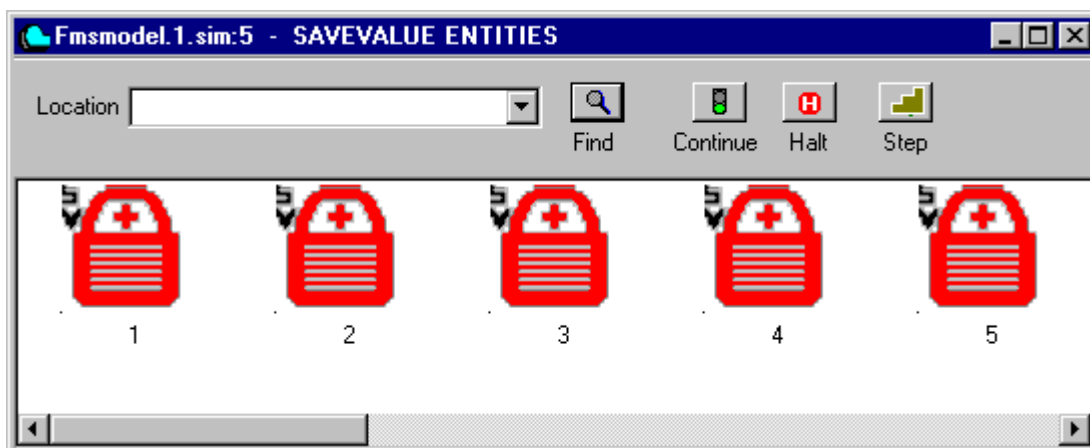


Рисунок 5.20 – Краткий обзор хранилищ

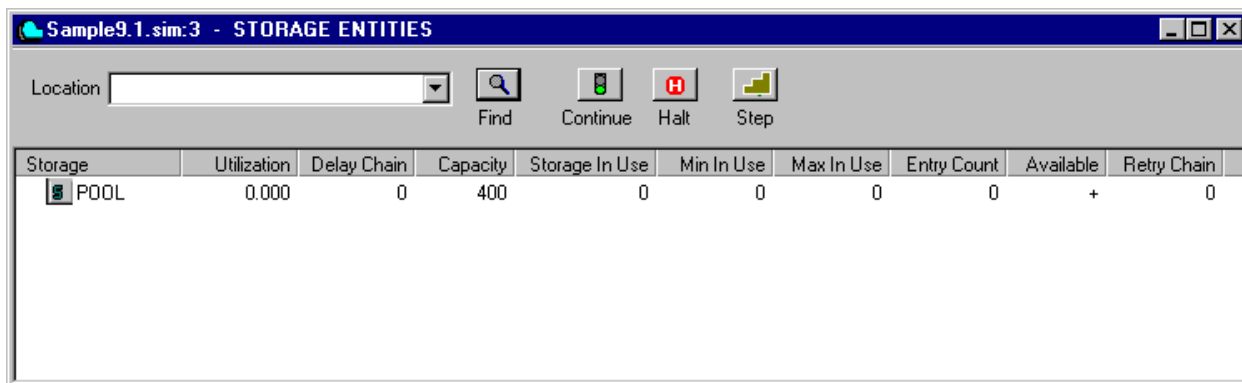


Рисунок 5.21 – Детальный обзор памяти

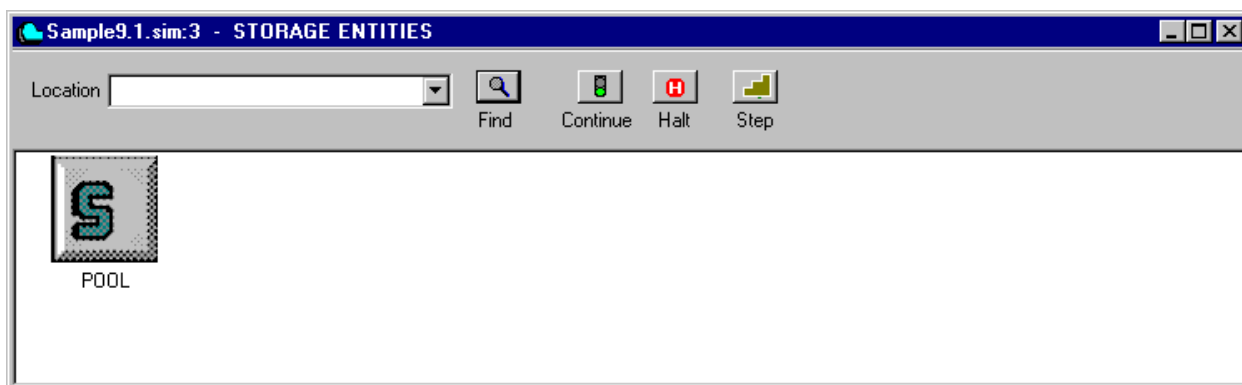


Рисунок 5.22 – Краткий обзор памяти

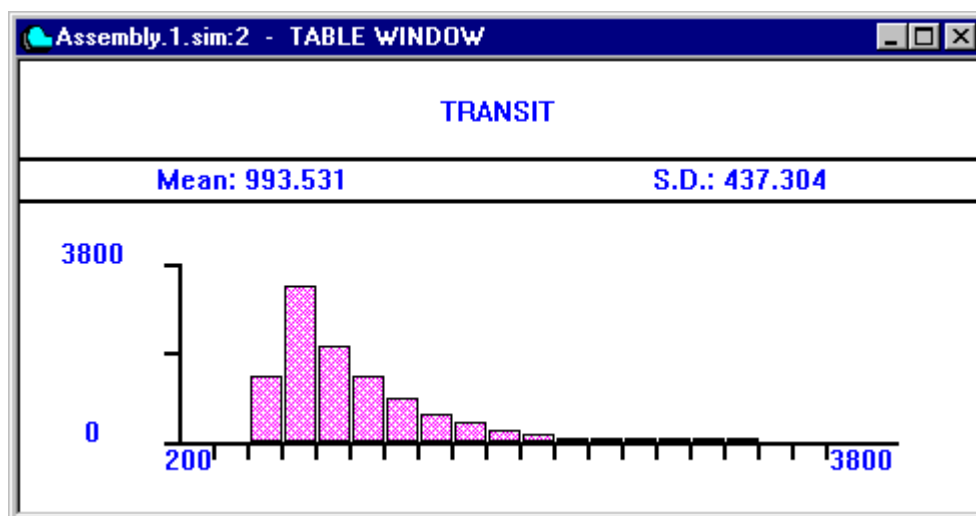


Рисунок 5.23 – Окно таблиц

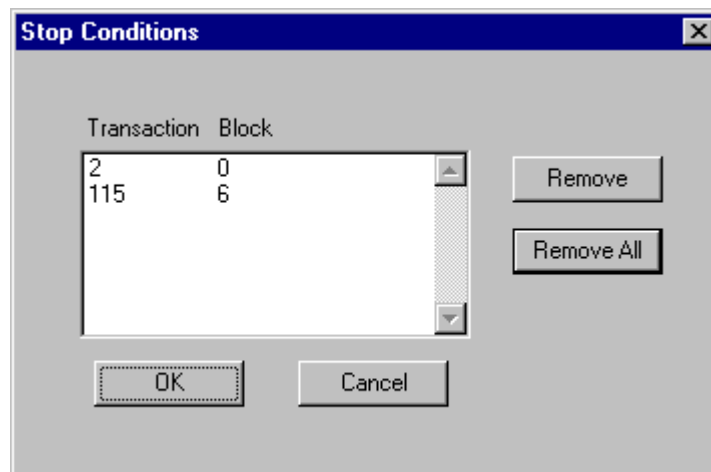


Рисунок 524 – Окно остановок пользователей

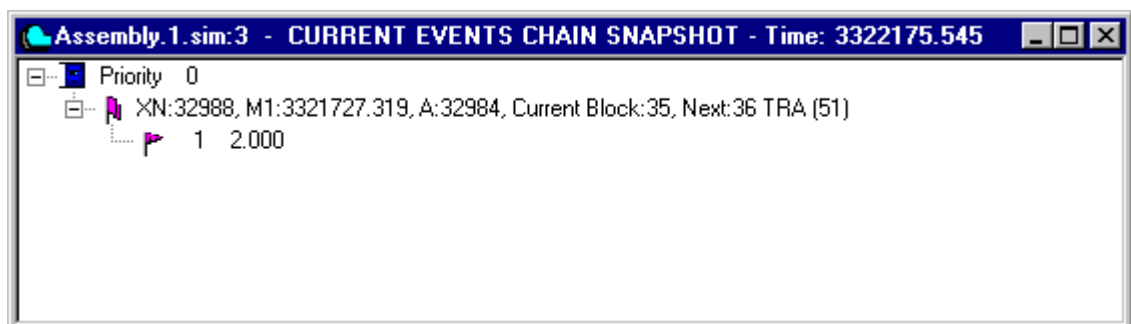


Рисунок 5.25 – Снимок текущего события

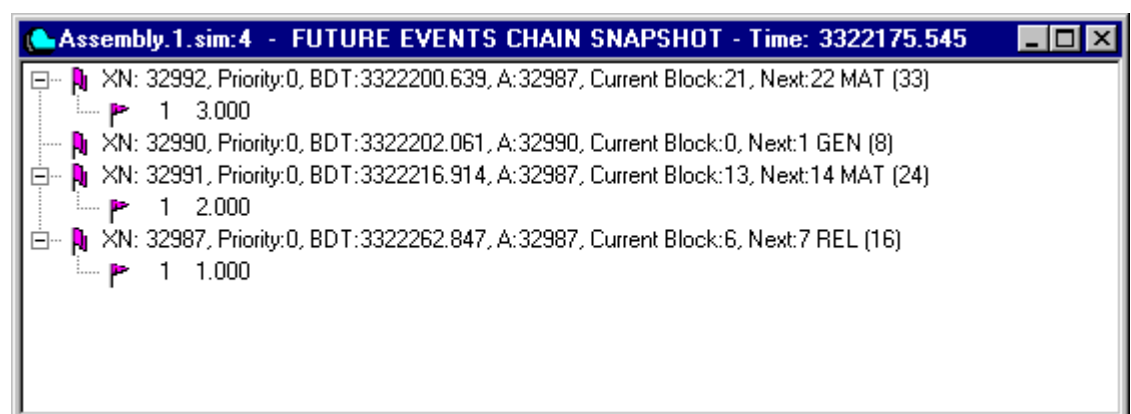


Рисунок 5.26 – Снимок последовательности будущих событий

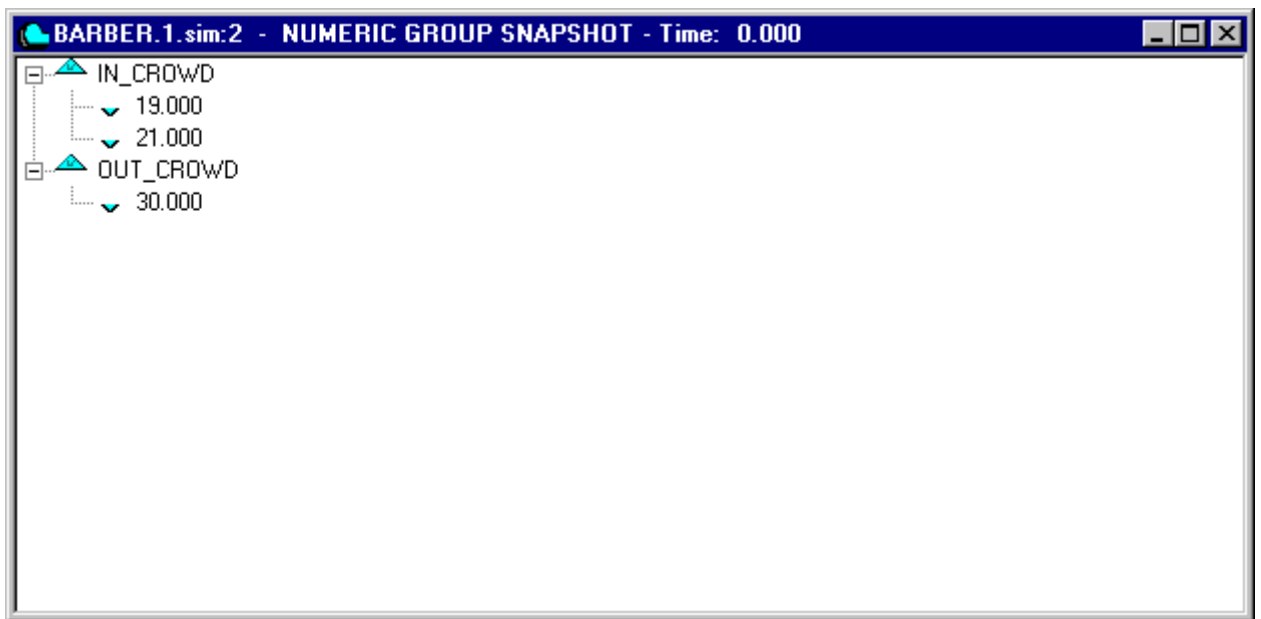


Рисунок 5.27 – Снимок групп чисел

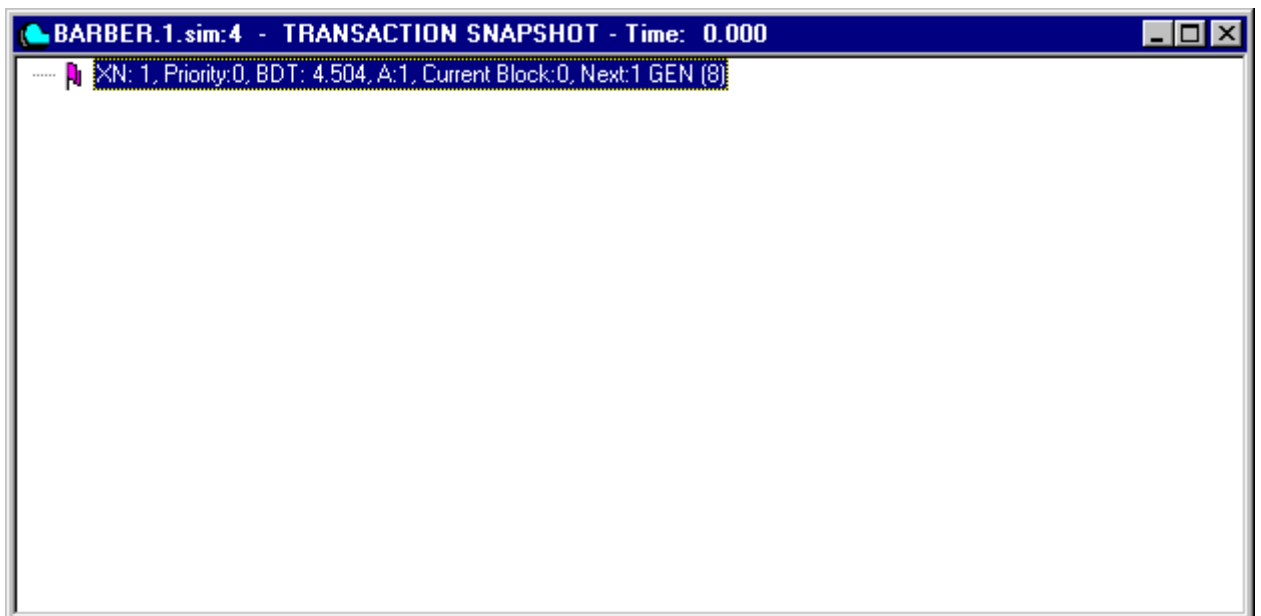


Рисунок 5.28 – Снимок заявок

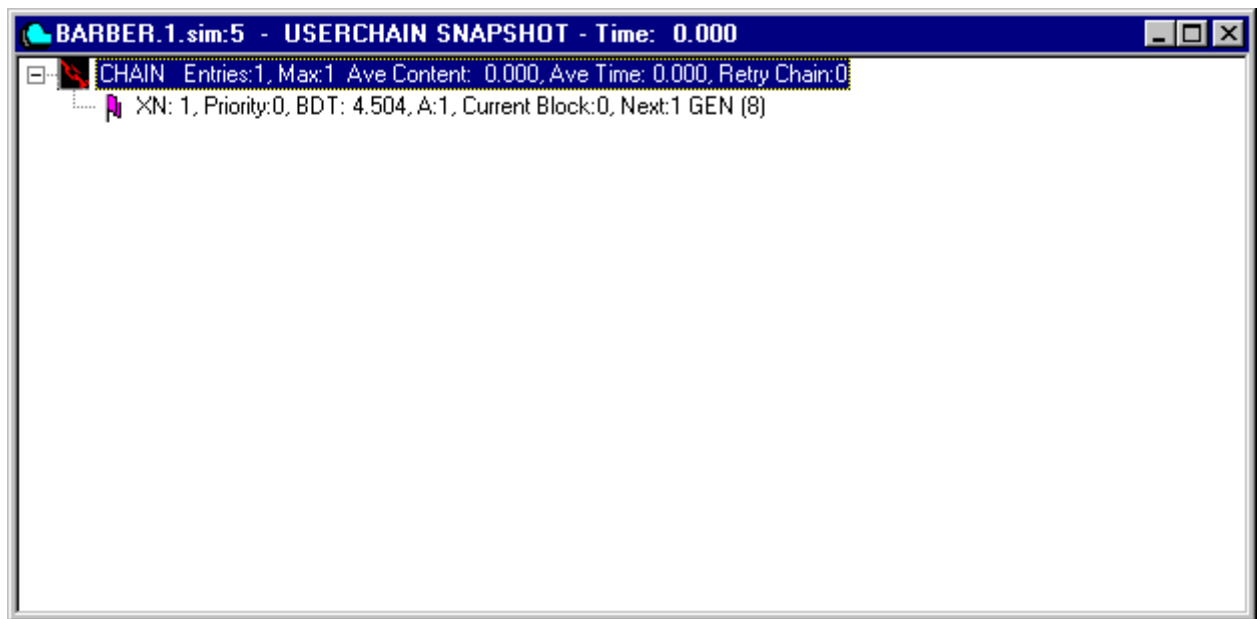


Рисунок 5.29 – Снимок пользовательской последовательности

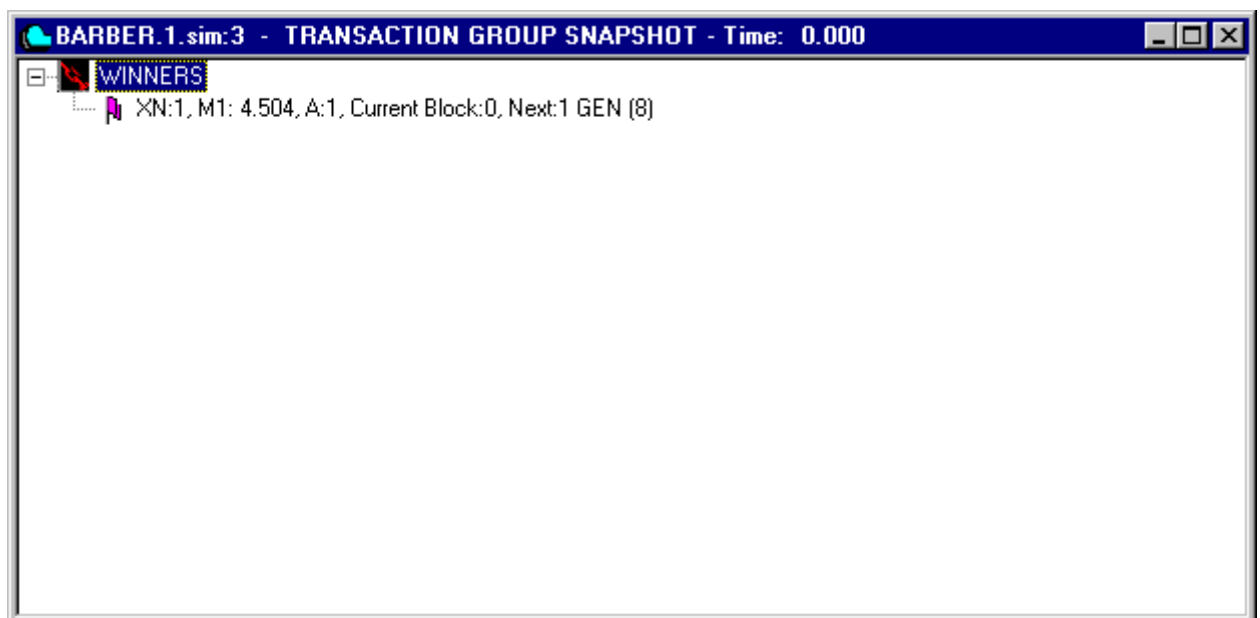


Рисунок 5.30 – Снимок группы заявок

Insert GPSS Block into Model Object		
ADOPT	ASSEMBLE	ALTER
ADVANCE	CLOSE	COUNT
ASSIGN	GATE	DISPLACE
BUFFER	JOIN	EXAMINE
DEPART	LINK	EXECUTE
ENTER	LOGIC	FAVAIL
GENERATE	LOOP	FUNAVAIL
LEAVE	MATCH	GATHER
MARK	OPEN	INDEX
MSAVEVALUE	PREEMPT	INTEGRATION
PLUS	PRIORITY	SAVAIL
QUEUE	READ	SCAN
RELEASE	REMOVE	SELECT
SAVEVALUE	RETURN	SUNAVAIL
SEIZE	SEEK	TABULATE
SPLIT	TEST	TRACE
TERMINATE	UNLINK	UNTRACE
TRANSFER	WRITE	

Рисунок 5.31 – Меню блоков ввода

Enter Block Information


ADOPT

ADOPT - Change Assembly Set.

A:

B:

C:

D:

E:

F:

G:

H:

New Assembly Set. Required.

Label:

Comment:

OK

Cancel

Help

Рисунок 5.32 – Диалог создания ADOPT блока

Screening Experiment Generator

Experiment Name:

'Run Procedure' Name:

Factors

	Name (User Variable)	Value 1	Value 2
A	<input type="text" value="Node_Count"/>	<input type="text" value="100"/>	<input type="text" value="300"/>
B	<input type="text" value="Min_Msg"/>	<input type="text" value="256"/>	<input type="text" value="512"/>
C	<input type="text" value="Max_Msg"/>	<input type="text" value="6072"/>	<input type="text" value="24288"/>
D	<input type="text" value="Fraction_Short_Msgs"/>	<input type="text" value="600"/>	<input type="text" value="300"/>
E	<input type="text" value="Intermessage_Time"/>	<input type="text" value="1.0"/>	<input type="text" value="0.5"/>
F	<input type="text"/>	<input type="text"/>	<input type="text"/>

Fraction: ☐ Full ☐ Half ☒ Quarter ☐ Eighth ☐ Sixteenth Run Count:

Result: Expression

☒ Generate Run Procedure ☒ Load F11 with CONDUCT Command

Рисунок 5.33 – Диалог создания области

Optimizing Experiment Generator

Experiment Name:

'Run Procedure' Name:

Initial Local Experimental Region

	Factor Name (User Variable)	Value 1	Value 2
A	<input type="text" value="Node_Count"/>	<input type="text" value="100"/>	<input type="text" value="300"/>
B	<input type="text" value="Min_Msg"/>	<input type="text" value="256"/>	<input type="text" value="512"/>
C	<input type="text" value="Max_Msg"/>	<input type="text" value="6072"/>	<input type="text" value="24288"/>
D	<input type="text" value="Fraction_Short_Msgs"/>	<input type="text" value="600"/>	<input type="text" value="300"/>
E	<input type="text" value="Intermessage_Time"/>	<input type="text" value="1.0"/>	<input type="text" value="0.5"/>

Movement Limits

	Low Limit	High Limit
A	<input type="text" value="0"/>	<input type="text" value="100"/>
B	<input type="text" value="0"/>	<input type="text" value="100"/>
C	<input type="text" value="0"/>	<input type="text" value="100"/>
D	<input type="text" value="0"/>	<input type="text" value="100"/>
E	<input type="text" value="0"/>	<input type="text" value="100"/>

Result: Expression

☒ Maximize ☐ Minimize

Redirection Limit:

☒ Generate Run Procedure ☒ Load F12 with CONDUCT Command

Рисунок 5.34 – Диалог оптимизации экспериментов

Глава 6 – Команды GPSS

Команды используются для определения объектов и контроля процессом моделирования. Команды могут быть частью начальной трансляции модели, или они могут исполняться как интерактивные выражения. Как часть трансляции модели, команды посылаются объекту моделирования как группа, после всех блоков. Или же, они отправляются сразу после набора. Такие команды называются интерактивными.

Чтобы послать команду моделированию, используется меню Command окна Model. Команда будет преобразована и послана объекту моделирования на исполнение.

Каждый процесс моделирования имеет очередь команд. Объект моделирования выполняет каждую команду в очереди одну за одной, пока этот процесс не будет остановлен, или до тех пор, пока команда не сможет знать, что делать. Даже команды меню File, HALT и SHOW помещаются в очередь команд, до выполнения.

Команды могут исполняться немедленно или помещаться в очередь. Команды, исполняющиеся немедленно, такие как HALT и SHOW, выполняются практически сразу, как только получаются объектом моделирования, а другие команды становятся в очередь. Они добавляются в конец списка команд, которые еще не были исполнены. Когда у объекта моделирования нет больше немедленных команд, она начинает исполнять следующую команду в очереди команд. Если при получении немедленно команды запущен процесс моделирования, он приостанавливается до завершения выполнения команды.

Помимо того, что команда HALT исполняется немедленно, она также удаляет любые команды в очереди. После выполнения этой команды, объекту моделирования нечего больше делать.

Их удобно добавлять в список часто используемых команд в маленький текстовый файл. А затем с помощью команды INCLUDE послать всю последовательность команд объекту моделирования. Для простоты, можно присвоить команде INCLUDE функциональную клавишу и вызывать в последствии сочетанием клавиш. В главе 2 показано, как это делается.

Команды:

BVARIABLE – определяет логическую переменную.

CLEAR – очищает статистику и удаляет заявку.

CONTINUE – продолжает моделирование.

EQU – присваивает значения переменной пользователя.

EXIT – заканчивает сессию GPSS World.

FUNCTION – определяет функцию.

FVARIABLE – определяет объект Fvariable.

HALT – останавливает моделирование и очищает очередь команд.

INCLUDE – читает и транслирует внешний файл модели.

INITIAL – Инициализирует или изменяет логический переключатель, хранилище, или матрицу.

INTEGRATE – Интегрирует переменную пользователя.

MATRIX – определяет матрицу.

QTABLE – определяет объект Qtable.

REPORT – Устанавливает имя файла отчета или запрашивает мгновенный отчет.

RESET – сбрасывает статистику моделирования.

RMULT – Устанавливает начальное число первых 7 генераторов случайных чисел.

SHOW – вычисляет и отображает выражение.

START – Начинает процесс моделирования.

STEP – Запрашивает блок записей.

STOP – устанавливает условие остановки.

STORAGE – определяет объект памяти.

TABLE – определяет таблицу.

VARIABLE – определяет переменную.

Операнды

Для выражения обычно необходимо задать один или более операндов. Большинство операндов имеют несколько правильных форм. Далее правильный класс операндов будет выделяться курсивом. Необходимо выбрать член класса и ввести его в поле операнда. Например, если один из правильных форм операнда принимает значения типа *PosInteger*, следует ввести:

21

Слова, выделенные курсивом, обычно объясняют все, что нужно, однако, иногда стоит обратиться к приложению, чтобы посмотреть формальное определение.

Окна

Доступен широкий выбор окон для отображения эффектов команд. Все окна называются в зависимости от типа объекта.

- Model Window – Текстовый вид — полноэкранный текстовый редактор.
- Journal Window – записывает события сессии.
- Blocks Window – показывает динамику изменения состояний блоков.
- Expressions Window – просмотр в реальном времени изменений значений выражений.
- Facilities Window – показывает динамику работы устройств.
- Logicswitches Window – показывает динамику работу логических переключателей.
- Matrix Window – показывает динамику изменений матрицы.
- Plot Window – отображает график из 8 последних значений.
- Queues Window – отображает динамику очереди.
- Savevalues Window – отображает динамику хранилища.
- Storages Window – отображает динамику объектов памяти.
- Table Window – отображает динамику изменения таблиц.
- Transaction Snapshot – снимок состояния заявки.
- CEC Snapshot – снимок текущего состояния последовательности событий.
- FEC Snapshot - снимок текущего состояния последовательности будущих событий.
- Numeric Groups Snapshot – снимок состояния динамики групп чисел.
- Userchains Snapshot – снимок состояния пользовательской последовательности.
- Transaction Groups Snapshot – снимок групп заявок.

BVARIABLE

Команда **BVARIABLE** объявляет объект **Bvariable**

NAME BVARIABLE X

Наименование/Операнд

NAME – является обязательным. Имя объекта.

X – выражение. Является обязательным. Выражения рассматривались в разделе 3.4.

Пример

LINE11 BVARIABLE (BV\$CLK'AND'BV\$PHASE2)

В этом примере определяется объект **Bvariable**, значение которого будет рассчитано при вызове **BV\$LINE11 SNA**. Результат будет равен 1 (**TRUE**). Если **CLK** и **PHASE2** равны **TRUE**, значение будет вычислено. В противном случае, **LINE11 Bvariable** вернет 0 (**FALSE**).

Действие

Команда **BVARIABLE** – команда очереди. Когда объект моделирования получает ее, то помещает в очередь команд.

Когда команда **BVARIABLE** выполнена, объект моделирования создает или переопределяет объект **Bvariable**. Объект **Bvariable** создан и определен тогда, когда

Системный числовой атрибут класса BV, ссылающийся на этот объект, доступен объекту моделирования.

Выражение, содержащееся в команде BVARIABLE вычисляется согласно правилам, описанным в главе 3, и может включать вызовы PLUS процедур или библиотечных процедур. Результат преобразуется в 0, если он 0, и в 1, если результат содержит ненулевое значение. Процесс вычисления в режиме совместимости GPSS/PC проходит по-другому.

Выражения должны быть составлены согласно правилам элементарной алгебры. Формальные определения находятся в приложении. Можно использовать любые арифметические и логические операторы, приведенные в разделе 3.4. Если в поле выражения используются SNA, то они вычисляются точно для активной заявки. Значение имени, которое не может быть однозначно установлено, не может быть использовано в качестве элемента выражения. В этом случае, необходимо задать значение до использования его в выражении. Установка значений пользовательских переменных производится командами EQU или PLUS процедурами.

Выражения в команде BVARIABLE не ограничены логическими операторами. Они могут включать арифметические операторы, а также вызовы библиотечных процедур. Значения логических переменных TRUE и FALSE интерпретируются как 1 и 0.

Созданный объект Bvariable, никогда не будет уничтожен. Однако, он может быть переопределен командой BVARIABLE.

Совместимость GPSS/PC

- Все SNA в этом режиме отсекаются.
- Все промежуточные результаты вычислений объекта Bvariable отбрасываются.

CLEAR

Команда CLEAR переводит процесс моделирования в свободное состояние (unused).
CLEAR A

Операнд

A – ON или OFF. Если операнд опущен, считается ON. Необязательный.

Действие

Команда CLEAR – команда очереди. При получении она помещается в очередь команд.

Команда CLEAR очищает всю накопленную статистику, удаляет все заявки из моделирования, затем начинает каждый блок GENERATE с его первой заявкой.

Состояние всех устройств и объектов памяти сбрасывается и переводиться в незанятое состояние. Содержимое всех блоков становится равным 0.

Когда исполняется команда CLEAR или CLEAR ON:

Все заявки удаляются из процесса моделирования.

Текущее количество устанавливается в 0.

Системные часы устанавливаются в 0.

Устройство переводятся в состояние «свободен и доступен».

Значения таблица устанавливаются в 0.

Объекты памяти устанавливаются в состояние полной доступности.

Время работы устройств, памяти, очереди и пользовательской последовательности устанавливается в 0.

Все счетчики устанавливаются в 0.

Минимальные и максимальные значение устанавливаются согласно содержимому, очереди, пользовательской последовательности и памяти.

Генераторы случайных чисел не сбрасываются.

Количество сгенерированных заявок в блоках GENERATE устанавливается в 0.

Удаляются все члены числовых групп.

Хранилище устанавливается в 0.

Логические переключатели сбрасываются.

Элементы матриц устанавливаются в 0.

Если используется опция CLEAR OFF, все выше перечисленные действия, кроме трех последних, не производятся. Когда параметр A – off, состояние хранилищ, логических переключателей, элементов матриц не изменяется.

Специальные ограничения

Отсутствуют

Связанные NSA

Отсутствуют

CONDUCT

Команда CONDUCT начинает эксперимент

CONDUCT A

Операнды

A – Вызов PLUS процедуры эксперимента. Должен быть вызовом процедуры *ProcedureCall*.

Действие

Команда CONDUCT – команда немедленного исполнения, которая может быть послана только неработающему объекту моделирования.

Команда CONDUCT формирует и передает аргументы предварительно зарегистрированному PLUS эксперименту объекта моделирования. Если у объекта моделирования может быть только один эксперимент без аргументов, параметр A не обязателен.

Пример

CONDUCT MyExperiment(NumberOfTellers, StartingReplicateNumber)

В этом примере, PLUS эксперимент MyExperiment начинается как обычная процедура. Переменные NumberOfTellers и StartingReplicateNumber используются для указания эксперименту, где начинать, а где продолжать моделирование. Аргументы вычисляются глобально и передаются эксперименту.

В процессе работы эксперимента командой CONDUCT, управление им ограничено. Всегда можно отобразить системное время моделирования (**View / Clock**), но необходимо остановить эксперимент, чтобы изменить какие либо свойства объекта моделирования.

Специальные ограничения

Команда CONDUCT не может быть запущена вызовом DoCommand.

В ходе эксперимента доступна только команда HALT.

Библиотечная процедура DoCommand может быть вызвана только в течение эксперимента.

Связанные SNA

Отсутствуют

CONTINUE

Команда CONTINUE продолжает остановленный процесс моделирования

CONTINUE

Операнды

Отсутствуют

Пример

CONTINUE

Эта команда используется для продолжения моделирования.

Действие

Команда CONTINUE – команда очереди. При получении она помещается в конец очереди команд. Она продолжает процесс моделирования после остановки. Процесс моделирования останавливается, когда выполняется условия остановки, поступает

команда HALT, возникает ошибка или же завершается согласно модели. Условия остановки могут быть заданы с помощью команд STOP и STEP.

Если в ходе моделирования выполняется условие остановки, команда CONTINUE заставляет процесс пропустить это условие, но не удаляет его. Если тоже условие выполнится снова, процесс моделирования снова остановится. Условие остановки должно быть удалено опцией OFF в команде STOP или в окне блоков. Когда модель транслирована, все условия остановки удаляются. Подробнее об этом написано в главе 2, в секции установка условий остановки.

CONTINUE может быть использована, когда процесс моделирования прерван командой HALT. Команда HALT удаляет все команды из очереди, однако незавершенные команды будут выполнены.

Когда объект моделирования обрабатывает команду CONTINUE, в первую очередь определяется наличие кода остановки. Это означает, что предыдущая команда START не была закончена. Если код остановки положителен, CONTINUE формирует стандартный отчет, но не планирует никаких заявок. Но, объект моделирования вызывает планировщик заявок для дальнейшей их обработки.

Горячие клавиши

Команда CONTINUE может быть вызвана сочетанием клавиш [Ctrl] + [Alt] + [C]. Окно GPSS World должно быть активным в этот момент.

EQU

Команда EQU вычисляет значение выражения и помещает результат в именованное значение.

NAME EQU X

Имя/Операнд

Name – имя переменной, которая примет значение. Является обязательным.

X – выражение. Обязательно. Выражения описаны в разделе 3.4.

Примеры

Price EQU 19.95

Эта команда определяет имя Price и присваивает ему значение 19.95. Следующее обращение к Price будет использовать числовое значение 19.95.

Действие

EQU – команда очереди. При получении она помещается в очередь команд.

Когда объект моделирования получает EQU команду, она создает или переопределяется именованное значение и вычисляет выражение. Созданное имя связывается с вычисленным значением выражения. Это значение заменяется, если будет вычислено еще раз в процессе моделирования.

Именованные значения могут использоваться как пользовательские переменные или как объекты, определяющие имена других объектов. Не следует задавать значение имени, используемому как название объекта. Имена определяются системой, т. е. имена, не присутствующие в команде EQU не могут быть использованы в выражениях или операндах. Однако, они могут быть использованы как указатели объектов в SNA. Объект моделирования автоматически присвоит значение соответствующему имени.

Выражение, содержащееся в команде EQU, вычисляется согласно правилам главы 3 и может включать вызовы PLUS процедур или библиотечных процедур. Выражения должны быть составлены согласно правилам элементарной алгебры. Формальное определение содержится в приложении. Можно использовать любые арифметические и логические операторы, приведенные в разделе 3.4. Именованные значения, которые неопределенны однозначно, не могут быть использованы в выражениях. Необходимо определить значение до вычисления выражения. Определение значений переменных производится командами EQU или PLUS процедурами.

Однажды созданное именованное значение, никогда не уничтожается. Однако переменные могут изменять значения, как результат последующих EQU команд, PLUS процедур или интегрирования. Числовое интегрирование переменных обсуждалось в главе 1 и производится командой INTEGRATE.

Специальные ограничения

- Значения имен блоков не могут быть изменены командой EQU.
- SNA точно вычисляются для активной заявки.

Если имя используемый как указатель объекта, изменилось после определения объекта, невозможно будет обратиться к старому объекту, используя это имя.

FVARIABLE и BVARIABLE используют одно пространство имен.

Если вы хотите назначить числовое значение имени объекта для использования блока SELECT, убедитесь, что имя/номер, назначения в EQU командах предшествует определению объекта. Например:

100 Stor1 EQU 1
200 Stor1 STORAGE 10000

В SNA и операндах, этот объект памяти может быть использован по номеру 1 или имени Stor1.

Связанные SNA

Отсутствуют

EXIT

Команда EXIT завершает сессию GPSS World.

EXIT

Операнды

A – Код выхода. Необязательный. Операнд должен быть -1, 0, 1 или Null.

Действие

Команда EXIT незамедлительно завершает сессию.

Операнд A может быть использован для записи объектов модели и моделирования в файлы. Если операнд имеет значение 0 или не определен, измененные файлы запросят сообщение о принятии изменений. Если значение операнда 1, будут сохранены все объекты. Если значение операнда -1, объекты не будут сохраняться.

Команда EXIT может быть использована в пакетном режиме, когда работает «невидимая сессия» без оконных операций. Пакетный режим описан в разделе 2.3.2.

Операции выхода также доступны в виде библиотечной процедуры, а потому может быть инициализирована в PLUS процедуре.

Специальные ограничения

Нет

Связанные SNA

Отсутствуют

FUNCTION

Команда FUNCTION определяет правила просмотра таблицы.

Существует несколько типов функций. Каждая имеет свои собственные правила просмотра таблицы. Для каждой, просмотр таблицы определяется один или несколькими выражениями Function Follower. Тип C просматривает таблицу с помощью линейной интерполяции.

Суть команды FUNCTION выделить определенное распределение из встроенных распределений библиотечной процедуры. Это описано в главе 8. GPSS World также поддерживает старые типы функции.

NAME FUNCTION A,B

Имя/Операнды

NAME – название объекта. Является обязательным.

A – аргумент функции. Обязателен. Может быть именем, PosInteger, String, выражением, SNA или *SNA*Parameter*.

B – Тип функции (одна буква) определяемый в выражении Function Follower. Обязателен.

Действие

Команда FUNCTION – команда очереди. При получении она помещается в очередь команд.

Команда FUNCTION вместе с одни или более выражениями Function Follower определяет функцию GPSS. Последующий вызов SNA класса FN возвращает результат выполнения функции. Операнд A рассчитывается численно. Есть несколько типов функций, который необходимо рассмотреть отдельно. Тип определяется операндом B команды FUNCTION.

В каждой команде FUNCTION должны содержаться данные, разделенные слешами, которые определяют таблицу. Текстовые строки, которые содержат список, называются выражениями Function Follower. Каждая пара данных имеет значение X и значение Y (или SNA) разделенные запятой. Выражения Function Follower создают таблицы, в которых содержатся математические функции для вычисления переменной. Когда объект моделирования натывается на ссылку *FNEntnum SNA*, он вычисляет значение функции и возвращает результат. Вид таблицы зависит от типа функции.

Модификаторы функции

Класс FN SNA, используемых в операнде B выражений GENERATE или ADVANCE называются модификаторами функции. Когда заявка попадает в блок GENERATE или ADVANCE с модификатором функции, результат выполнения функции умножается операндом A блока, и используется для приращения времени. Операнд C выражения ASSIGN также называется модификатором функции, хотя определяется по-другому. Только указатель объекта, не класс FN SNA, используется в выражении ASSIGN при поступлении заявки в блок ASSIGN с модификатором, операнд C используется для определения номера объекта функции. Затем, результат вычислений функции умножается операндом B блока ASSIGN и конечный результат используется как ASSIGN значение.

Типы функций

Существует пять различных типов функций:

Тип C

Это непрерывные функции. Получая значение X, в результате линейной интерполяции возвращает значение Y.

В выражении Function Follower C функции значения X и Y должны быть типа Integer или Real. Функции типа C без случайного аргумента, пары данных выражения Function Follower определяют кусочногладкую линейную функцию одного аргумента. Первая пара данных определяет левую точку, а другая пара определяет правую конечную точку. Значения X и Y хранятся как числа двойной точности с плавающей точкой.

Вычисление функции начинается с вычисления аргумента. Результат используется для определения линейного сегмента функции. Аргумент приводится к виду двойной точности линейной интерполяции и возвращается как результат с двойной точностью функции. Если аргумент выходит за границы конечных точек, возвращается значение ближайшей конечной точки.

Когда операнд A представляет собой класс RN SNA, говорят о функции со случайным аргументом. Тип C функции со случайным аргументом используется для определения непрерывного распределения вероятности. Такая функция представляет собой функцию накопления распределения (CDF) с 0 значением в левой точке и значением 1 в правой. CDF представляет собой часть линейной функции. Случайное число между 0 и 999999 включительно, используется при линейной интерполяции и возвращается как значение функции с двойной точностью.

Пример

Output FUNCTION V\$Input,C3 1.1,10.1/20.5,98.7/33.3,889.2

В этом примере задается кусок линейной функции с двумя сегментами. Когда для FN\$Output SNA получено значение функции, значит, вычислен первый аргумент V\$Input. Если значение результата выходит за пределы 1.1 и 33.3, возвращается значение ближайшей конечной точки. Например, если V\$Input возвращает 1 или меньше, FN\$Output возвращает 10.1

Если результат имеет значение ниже, чем определенное внутри линейного сегмента, выполняется линейная интерполяция. Например, если V\$Input возвращает 25, тогда FN\$Output возвращает результат следующих вычислений:

$$98.7 + (889.2 - 98.7) \# (25 - 20.5) / (33.3 - 20.5)$$

В конечном итоге, 376.6101563..

Пример

Xpdis FUNCTION RN200,C24 0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/.75,1.38 .8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2 .97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8

В этом примере мы определили функцию с именем Xpdis. Команда FUNCTION определяет функцию типа C 24 парами данных, использующую генератор случайных чисел 200.

Это пример аппроксимации к отрицательному экспоненциальному распределению со значением 1. Можно использовать встроенные распределения вероятности из библиотечных процедур, которые будут более точными, но функции являются более эффективными.

Тип D

Это дискретные функции. Каждое значение аргументы или массива представляет собой числовое значение.

В выражениях Function Follower функций D, значения X – это выражения, а Y – значения типа *Integer*, *Real* или *Name*.

Если не используется случайный аргумент, то в выражения Function Follower определяет набор входных, значений, которым соответствует набор выходных значений функции. Значения должны быть неубывающими. Они представляют собой значения двойной точности. При вычислении значения функции выбирается X между минимальным и максимальным. Когда выбирается X, значение, которое больше или равно значению аргумента, возвращается соответствующее ему значение Y. Если не удастся подобрать X в качестве результата, возвращается максимальное значение X.

Пример

DIs1 FUNCTION X\$A2,D5 1.1,6.9/2.1,7/6.33,9.4/7,10/9.9,12.01

Функции типа D со случайным аргументом используются для определения дискретного распределения вероятности. Функции представляет собой функцию накопления распределения (CDF) с 0 значением левой точки и 1 значением правой. При вычислении такой функции выбирается случайное число от 0 до 999999. При это подбирается самое маленькое значение X, которое больше или равно выбранному случайному числу.

Пример

Ran1 FUNCTION RN1,D5 0,0/.2,7.2/.4,6.667/.8,9.92/1.0,10

Тип E

Это дискретные функции с атрибутами. Значение каждого аргумента вычисляется с помощью SNA.

Значения X – выражения, а значения Y имеет тип *Integer*, *Real*, *Name*, *SNA* или *выражения*.

Функции E рассчитываются также как и функции D, с добавлением еще одного шага. После подбора X, соответствующая SNA (значение Y) вычисляет и возвращает результат.

Пример

**Edisc FUNCTION X\$QRA,E5
1,S\$Stor1/3,S\$Stor2/5,S\$Stor3/9,S\$Stor5/10,S\$Stor6**

Тип L

Это функции списка. Значение аргумента используется для определения номера элемента списка, значение которого нужно получить.

X значение – имеет тип *Integer*, Y – *Integer*, *Real* или *Name*.

Выражение Function Follower определяет список значений, из которого будет выбран возвращаемый результат. Когда вычисляется значение функции, значение аргумента интерпретируется как порядковый номер элемента списка. Возвращается значение указанного элемента списка. Если значение аргумента слишком велико или меньше 1, возникает ошибка. Значение X должно начинаться с 1 и увеличиваться на 1 для каждого последующего элемента.

Пример

**Listtype FUNCTION Q\$Barber,L5
1,PAR1/2,PAR2/3,PAR3/4,PAR4/5,PAR5**

Тип M

Это функции списка с атрибутами. Значение аргумента используется для определения номера элемента списка, значение которого нужно получить.

X значение – имеет тип *Integer*, Y – *Integer*, *Real*, *Name*, *SNA*, или *ParenthesizedExpression*.

Значение M функций вычисляется также как и L функций с добавлением одного шага. После определения позиции в списке, соответствующая SNA вычисляет и возвращает результат.

Пример

**Mlist FUNCTION X\$Name1,M5
1,Q\$Nnam1/2,Q\$NamX/3,Q\$Nam4/4,Q\$Nam6/5,F\$Tan1**

Правила для функций

При создании функций необходимо придерживаться нескольких правил. Они одинаково применимы как к командам FUNCTION, так и к выражениям Function Follower.

Значения X должны быть неубывающими.

Строки выражения Function Follower никогда не нумеруются.

Функция со случайным аргументом должно описывать подходящее распределение вероятности в выражении Function Follower.

Необходимо заполнять все поля команды FUNCTION.

Необходимо указывать оба значения X и Y в выражении Function Follower.

Число пар данных, указанных в операнде B команды FUNCTION должно соответствовать парам, разделенным слешами [/] в выражении Follower Statements.

X1 – первое возможное значение, определяющее непрерывную случайную функцию, должно быть 0. Значения CDF должны быть неотрицательными, неубывающими, и не должны включать 1.

Выражение Function Follower не предусматривает полей комментариев.

В выражении Function Follower значение X сопровождается [,], [,] сопровождается значением Y, значение Y сопровождается [/] или [CR], а [/] сопровождается значением X. Значения CDF должны быть неотрицательными и лежать в пределах между 0 и 1 включительно. Любая пропущенная вероятность функций C считается для правого интервала и выдает ошибку.

У Функций типа C, D и L отсутствуют SNA для получения значений Y.

У функций типа E и M есть SNA и выражения для получения значений Y.

Функции типа L и M не имеют случайных аргументов.

Выражения Function Follower функций L и M должны содержать последовательности значений X, начинающихся с 1.

Специальные правила, применяемые при режиме совместимости GPSS/PC, описаны в главе 3.

Связанные SNA

- *FNEntnum* – функция. Результат вычисления функции *Entnum*.

FVARIABLE

Команда FVARIABLE определяет переменную с плавающей точкой.

NAME FVARIABLE X

Название/Операнд

Name – название переменной. Обязательно.

X – выражение. Обязательно. Выражения описаны в разделе 3.4.

Пример

VAR1 FVARIABLE 5#LOG(Q\$WAITINGLINE)

Эта команда определяет переменную с плавающей точкой, выражение которой будет вычислено при ссылке на V\$VAR1. Доступ к переменным, созданным командами VARIABLE и FVARIABLE можно получить с помощью класса V SNA. Когда вызывает V\$VAR1 SNA, вычисляется выражение, указанное в переменной, начиная с Q\$WAITINGLINE SNA. Вычисляется логарифм от результата и умножается на 5. Результат отсекается и возвращается как значение SNA.

Действие

Команда FVARIABLE – команда очереди. При получении она помещается в очередь команд.

При исполнении команды FVARIABLE объект моделирования создает или переопределяет переменную GPSS. Значение созданной переменной вычисляется при ссылке на этот объект переменной.

Выражение, содержащееся в команде FVARIABLE, вычисляется согласно правилам главы 3, и может включать вызовы PLUS процедур или библиотечных процедур. В режиме совместимости GPSS/PC вычисление происходит по-другому.

Выражения должны быть составлены согласно правилам элементарной алгебры. Формальное определение приведено в приложении. Можно использовать арифметические и логические операторы, приведенные в разделе 3.4. Если в поле выражения используются SNA, то они вычисляются для активной заявки. Именное значение, неопределенное однозначно, не может использоваться в выражении.

Созданная переменная, никогда не удаляется из моделирования. Однако она может быть переопределена командой FVARIABLE.

Если моделирование выполняется не в режиме совместимости GPSS/PC, команды VARIABLE и FVARIABLE обрабатываются одновременно.

Совместимость GPSS/PC

- Все SNA отбрасываются в этом режиме.

Связанные SNA

- *VEntnum* – результат вычисления выражения переменной *Entnum*.

HALT

Команда HALT прерывает процесс моделирования и очищает очередь команд.

HALT

Операнды

Отсутствуют

Действие

Команда HALT – команда немедленного исполнения, она не помещается в очередь команд. Она исполняется немедленно, переводя процесс моделирования в состояние

остановки и удаляя любые команды из очереди команд. Моделирование может быть продолжено командой CONTINUE.

Пример

HALT

Это единственный способ использовать команду GROUPS.

Горячие клавиши

Эта команда может быть вызвана сочетанием клавиш [Ctrl] + [Alt] + [H]. Окно GPSS World должно быть активно в этот момент.

INCLUDE

Команда INCLUDE транслирует команду внешнего файла модели.

INCLUDE A

Операнд

A – файл. Строка, ссылающаяся на файл, содержащий команды. Является обязательным. Должен быть типа *String*.

Пример

INCLUDE "SAMPLE1.TXT"

В этом примере, когда транслятор обрабатывает команду INCLUDE, он обрабатывает команды из файла SAMPLE1.TXT в текущем моделировании. Если не указан путь к файлу, GPSS World использует текущий каталог модели.

Действие

Команда INCLUDE – немедленная команда. Она считывает команды и файла, которые потом будут обработаны. Файлы имеют формат обычных текстовых файлов. При использовании выражения INCLUDE, подключаемые файлы называются вторичными файлами модели.

Операнд A используется для указания файла, содержащего список команд. Если не указан путь к файлу, будет использован каталог файла модели, в котором указано выражение INCLUDE. При обработке выражения INCLUDE, транслятор GPSS World заменяет его текстовым объектом, указанным в выражении. Транслятор открывает вторичный файл модели и транслирует каждое выражение в нем.

Выражения файла команд или вторичного файла модели обрабатываются так, как они записаны в строке. Допускается вложенность глубиной в 5 единиц. Когда все выражения транслированы, они направляются на обработку объекту моделирования.

Если обнаруживаются ошибки, GPSS World подает звуковой сигнал, генерирует сообщение об ошибке синтаксиса, также ищет другие ошибки. Сообщение об ошибке содержит номер строки вторичного файла модели, где она произошла. Обработка ошибок описана в главе 2. Звук может быть отключен в настройках модели выбором опции Silence.

Иногда удобно сделать проверку трансляции команд файла, до использования его в INCLUDE. Это делается следующим образом:

1. Открыть вторичный файл модели в окне модели.
2. Транслировать модель, представленную в окне модели.
3. Исправить синтаксические ошибки.
4. Сохранить файл

Команда INCLUDE может быть введена интерактивно, также ей можно присвоить функциональные клавиши, как и любому другому выражению GPSS. Это описано в главе 2.

Номера файла модели

Номера файла модели используются в окне блоков и в сообщениях об ошибках для определения блока выражения, которое создано данным блоком.

Как только транслятор находит файл модели, он присваивает ему числовое значение, которое впоследствии будет использовано для идентификации файла. Объекту модели

назначается номер 0, а обрабатываемым файлам назначаются числа в топ порядке, в котором они обрабатываются транслятором. После этого, объекты могут быть связаны с соответствующей строкой в файле модели.

При каждом вхождении файла, присваивается соответствующий номер. При многократном вхождении одного файла, присваивается разное число. При многократном вхождении файла с разным набором блоков, каждый раз присваивается уникальный идентификатор.

Специальные ограничения

Можно использовать команды INCLUDE к вложенным файлам глубиной до 5 единиц.

Нельзя помещать команду INCLUDE внутри PLUS процедуры.

INITIAL

Команда INITIAL инициализирует объекты матрицы, логические переключатели, хранилища или элементы матриц.

INITIAL A,B

Операнды

A – Один из вышеперечисленных объектов, определяемы как SNA, или имя матрицы.

Операнд A должен иметь форму LS, X или MX классов SNA или имя матрицы. Обязателен. Операнд должен иметь тип *Name*, *LSPosInteger*, *LS\$Name*, *XPosInteger*, *X\$Name*, *MXPosInteger(m,n)* или *MX\$Name(m,n)*. Координаты (m,n) должны иметь тип *Name* или *PosInteger*.

B – назначаемое значение или «не указано». По умолчанию 0. Не обязателен. Операнд должен быть *Null*, *Number*, *String*, *Name*, или не указан.

Действие

Команда INITIAL – команда очереди. При получении она помещается в очередь команд.

Команда присваивает значение, указанное операндом B, логическому переключателю, хранилищу, матрице, указанных операндом A. Если операнд B не указан, значение 1 назначается объекту или элементу.

Если операнд B имеет значение «не указано», объекты переводятся в неопределенное состояние. Это можно использовать для отображения потерянной информации в результатах матриц, чтобы проанализировать библиотекой ANOVA.

Если операнд A указывает на логический переключатель, могут быть использованы только значения 0 или 1. Если значение операнда B 0, будет присвоено 0, если 1, то будет присвоено 1. Опция НЕОПРЕДЕЛЕННО не может быть использована с классом LS SNA, потому что логический переключатель не имеет неопределенного состояния.

Если операнд A определяет имя матрицы, все ее элементы будут переведены в состояние, указанное операндом B. Значение по умолчанию – 0. Можно использовать MX SNA в операнде A для определения элементов квадратной матрицы размерностью 2. Чтобы перевести такие элементы в НЕОПРЕДЕЛЕННОЕ состояние, можно инициализировать хранилища в НЕОПРЕДЕЛЕННОЕ состояние и использовать PLUS выражения.

Команду INITIAL можно использовать, когда нет активной заявки, или не определен параметр заявки. Использование операндов позволяет работать с объектами интерактивно.

Примеры

INITIAL X\$Quote,"Now is the time "

Эта команда назначает строковую константу хранилищу, QUOTE.

INITIAL MX\$Inventory(Part_39,Stocklevel),200

Эта команда присваивает значения 200 элементам матрицы Inventory с количеством строк Part_39 и количеством столбцов Stocklevel. Переменные Part_39 и Stocklevel должны быть определены заранее командами EQU.

INITIAL MainResult,UNSPECIFIED

Эта команда инициализирует определенную заранее матрицу MainResult для использования в эксперименте, в ходе которого могут быть утеряны данные.

Специальные ограничения

Операнд А должен иметь форму LS, X, или MX классов SNA, или имени матрицы.

Нельзя использовать параметры заявок в операнде.

Нельзя использовать состояние НЕОПРЕДЕЛЕНО при инициализации логических переключателей.

Связанные SNA

- *LSEntnum* – логический переключатель. Возвращает значение логического переключателя *Entnum*.
- *MXEntnum(m,n)* – элемент матрицы. Возвращает значение строки m столбца n матрицы *Entnum*.
- *XEntnum* – хранилище. Возвращает значение хранилища *Entnum*.

Связанные блоки

- LOGIC – назначает значение логическому переключателю.
- NSAVEVALUE – назначает значение или инкрементирует элемент матрицы.
- SAVEVALUE – назначает значение или инкрементирует хранилище.

INTEGRATE

Команда INTEGRATE устанавливает пределы интегрирования непрерывных переменных.

NAME INTEGRATE A,B,C,D,E

Операнды

NAME – переменная. Обязателен.

A – производная. Обязателен. Должен иметь тип *Name*, *Number*, *String*, *ParenthesizedExpression*, или *SNA*.

B – граница 1. Необязателен. Должен иметь тип *Null*, *Name*, *Number*, *String*, *ParenthesizedExpression*, или *SNA*.

C – принимающий блок 1. Не обязателен. Должен иметь тип *Null*, *Name*, *Number*, *String*, *ParenthesizedExpression*, или *SNA*.

D – граница 2. Не обязателен. Должен иметь тип *Null*, *Name*, *Number*, *String*, *ParenthesizedExpression*, или *SNA*.

E – принимающий блок 2. Не обязателен. Должен иметь тип *Null*, *Name*, *String*, *ParenthesizedExpression*, или *SNA*.

Examples

Rabbits INTEGRATE (a_ # Rabbits) - (b_ # Rabbits # Foxes)

Foxes INTEGRATE (- c_ # Foxes) + (d_ # Rabbits # Foxes)

Rabbits EQU 10000

Foxes EQU 1500

a_ EQU 0.9

b_ EQU 0.4

c_ EQU 0.2

d_ EQU 0.1

Этот пример определяет модель «хищник-жертва», на основе популяций кроликов и лис. Выражения в командах INTEGRATE представляют собой производные по времени. Значения констант и выражений, а также начальные значения популяций устанавливаются с помощью команд EQU. В процессе моделирования интегрирование выполняется автоматически при изменении времени.

X_ INTEGRATE (Y_),0.707,Wake_Up

Y_ INTEGRATE (-X_)

X_ EQU 1.0

Y_ EQU 0.0

В этом примере представлена двойная система простых дифференциальных уравнений (ODEs), которые решаются: $X_ = \cos(Y_VAR)$ $Y_ = -\sin(X_VAR)$. Выражения в командах

INTEGRATE есть производные по времени. Начальные значения переменных устанавливаются командами EQU. В процессе моделирования интегрирования производится автоматически между интервалами времени.

Границы используются для переменной X_. Когда значение переменной достигает 0,707 в любом направлении, генерируется новая заявка для блока WAKE_UP.

X – класс SNA, поэтому не может использоваться в качестве именных значений. Здесь мы используем X_. Всегда более безопасно давать имена с символом подчеркивания.

Действие

Команда INTEGRATE – команда очереди. При получении она помещается в очередь команд.

При выполнении команды INTEGRATE структуры данных устанавливаются так, что переменные будут интегрироваться автоматически, когда изменяется системное время. Интегрирование производится при помощи модифицированного пяти ступенчатого метода Рунга-Кутты-Фелберга (RK4(5)) с переменным шагом.

Всем интегрируемым переменным должно быть заранее присвоено значение перед процессом моделирования. Это можно делать с помощью команд EQU или PLUS процедур.

Операнд А команды INTEGRATE задает производные по времени для интегрирования. Они могут быть простыми или сложными. Впоследствии можно определять PLUS процедуры и помещать вызовы процедур в скобки выражений.

Каждая команда INTEGRATE может иметь нулевую, одну или две числовых границы. Операнды В и С используются для определения границы 1, и/или операнды D и E используются для определения границы 2. Первый операнд пары определяет значение границы, второй указывает блок, который получит сгенерированную заявку.

В процессе интегрирования, если значение интегрированной переменной пересекает предел в любом направлении, генерируется новая заявка. Ей задается 0 приоритет и направляется в блок, связанный с этой границей в команде INTEGRATE. Время прихода заявки рассчитывается линейной интерполяцией. Для повышения точности, шаг уменьшается при приближении к пределу.

Пределы могут содержать константы, выражения и даже вызовы процедур. Заявка, сгенерированная пересечением предела, может быть использована для смещения предела.

Интегрирование производится автоматически в активном состоянии. Но, можно менять состояние интегрирования с помощью блоков INTEGRATION. Это описано в главе 7.

Фазы

Моделирование проходит в двух фазах дискретной и непрерывной. В момент времени, когда события запланированы, моделирование проходит в дискретной фазе. Системное время не может измениться, не пройдя дискретную фазу. Между интервалами времени моделирования протекает в непрерывной фазе, в течение которой интегрирование протекает маленькими скачками времени, называемыми минишагами. График интегрирования отображает значения в конце каждого минишага.

Когда пересечение предела генерирует заявку, процесс моделирования переходит в дискретную фазу. Поэтому дискретная и непрерывная фазы связаны между собой. Значения пользовательских переменных могут быть установлены в дискретной фазе, даже если они были интегрированы. Это можно сделать, используя команду EQU или PLUS выражения. Если вы хотите, чтобы такие изменения произошли в процессе моделирования, то необходимо объявить PLUS процедуру, которая сделает такие изменения. Например:

```
PROCEDURE SetPop(PopLevel) BEGIN  
    Foxes = PopLevel ;  
    END ;
```

Можно переопределить переменную Foxes с помощью PLUS блока:

```
PLUS (SetPop(200))
```

или используя выражение SetPop() других блоков.

Ошибка интегрирования

Настройка модели, называемая «устойчивость интегрирования» используется для ограничения локального отсечения ошибок интегрирования. Эта настройка применяется для всех интегрирований, выполняемых в процессе моделирования. Если сделать устойчивость меньше, интегрирование будет происходить дольше, но будет более точной. Оно устанавливается в настройках модели.

Выберите **Edit / Settings**

Затем, выберите страницу **Simulate**. И укажите желаемое значение в поле «Устойчивость интегрирования». Значение по умолчанию 10^{-6} .

Моделирование в непрерывном режиме описано в главе 2.

Связанные блоки

- INTEGRATION – включает или выключает интегрирование переменных.

Связанные окна

- Окно выражений – просмотр динамических изменений выражений.
- Окно графиков – график динамического изменения значений выражений. Показывает до 8 выражений.

Ограничения

- Операнды В и С должны использоваться вместе.
- Операнды D и E должны использоваться вместе.

MATRIX

Команда MATRIX определяет GPSS матрицу.

NAME MATRIX A,B,C,D,E,F,G

Название/Операнды

NAME – название объекта. Обязательно.

A – Неиспользуемое поле (для совместимости со старым синтаксисом GPSS).

B – Необходим. Максимальное количество элементов первой размерности. Число строк. Должен иметь тип *PosInteger*.

C – Необходим. Максимальное количество элементов второй размерности. Число столбцов. Должен иметь тип *PosInteger*.

D – Необязателен. Максимальное количество элементов третьей размерности. Должен иметь тип *PosInteger*.

E – Необязателен. Максимальное количество элементов четвертой размерности. Должен иметь тип *PosInteger*.

F – Необязателен. Максимальное количество элементов пятой размерности. Должен иметь тип *PosInteger*.

G – Необязателен. Максимальное количество элементов шестой размерности. Должен иметь тип *PosInteger*.

Пример

Inventory MATRIX ,1000,5

Эта команда задает матрицу с именем Inventory с 1000 строк и 5 столбцами.

Действие

Команда MATRIX – команда очереди. При получении она помещается в очередь команд. Команда создает объект матрицы. Все матрицы должны быть заданы этой командой до использования.

Операнд A не используется в GPSS World, потому что нет необходимости задавать точность элементов матрицы. Он применяется только для совместимости со старыми версиями GPSS.

Объект матрицы может иметь максимальную размерность 6. Однако в блоке MSAVEVALUE доступны только первые две. Все пропущенные координаты будут предположительно равны 1.

PLUS процедуры могут манипулировать всеми элементами матриц. Если необходимо использовать матрицы более высокой размерности, большей 2, необходимо создать одну или более PLUS процедур для управления ими. Матрицы, заданные командой MATRIX имеют глобальную видимость и доступны любой PLUS процедуре. А вот временные матрицы с локальной областью видимости могут быть созданы только на время работы PLUS процедуры. Это описано в главе 8.

Объекты матриц никогда не удаляются из моделирования. Однако, они могут быть переопределены командой MATRIX.

Когда матрица создается или когда выполняется команда CLEAR ON, всем элементам матрицы присваивается значение 0. Но, можно использовать команду INITIAL, чтобы задать элементам НЕОПРЕДЕЛЕННОЕ состояние. Это полезно, когда матрица используется для хранения результатов экспериментов. Когда значения матрицы передаются библиотечной процедуре ANOVA, неопределенные элементы обрабатываются как потерянные данные, в отличие от значений 0.

Ограничения памяти

Память, выделяемая под матрицы, ограничена. Это описано в главе 2.

Связанные SNA

- *MXEntnum(m, n)* – элемент матрицы. Возвращает значение строки *m* столбца *n* матрицы *Entnum*. Для значений строк и столбцов могут использоваться только следующие типы: *names*, *integers* или Р класс SNA.

Связанные блоки

- *MSAVEVALUE* – задает элемент матрицы.

Связанные команды

- *INITIAL* – инициализирует матрицу 0 или НЕОПРЕДЕЛЕННЫМ состоянием.
- *CLEAR* – устанавливает все элементы в 0..

Связанные окна

- Динамическое состояние матрицы отображается в окне матриц. Это окно показывает ячейки любой 2 размерной матрицы.
- В окне графиков или выражений, отображаются любые SNA.

QTABLE

Команда QTABLE инициализирует таблицу частоты распределения.

NAME QTABLE A,B,C,D

Название/Операнды

NAME – Название объекта. Обязателен.

A – имя очереди. Обязателен. Должен иметь тип *PosInteger* или *Name*

B – Верхний предел первого класса частоты. Аргумент максимума, который обновляет верхний предел частоты. Обязателен. Должен иметь тип *PosInteger* или *Name*

C – Размер класса частоты. Разность между верхним и нижним пределами для каждой частоты. Обязателен. Должен иметь тип *PosInteger* или *Name*.

D – Число классов частоты. Должен иметь тип *PosInteger*.

Действие

Команда QTABLE – команда очереди. При получении она помещается в очередь команд.

Используя команду QTABLE можно автоматически накапливать статистику заявок, поступивших в блоки QUEUE и DEPART. Объект очереди определяется операндом A в определении команды QTABLE.

Когда заявка попадает в блок QUEUE, который ссылается на объект очереди, содержащий одну или более таблиц Qtable, к заявке прикрепляется значение времени. Чтобы собрать все данные, заявка должна пройти блок DEPART, который ссылается на тот же самый объект очереди. Когда это происходит, аргумент таблицы вычисляется разностью сохраненного времени в заявке и текущего системного времени. Если посчитанное значение меньше или равно значению операнда B команды QTABLE, выбирается первый

класс частоты. Если аргумент таблицы не подходит в первый класс точности, класс определяется делением аргумента на значения операнда *C*. Нижний предел класса частоты включается в предыдущий класс. Если емкости таблицы не достаточно для хранения значений, выбирается последний класс частоты.

Затем, число в выбранном классе частоты и общее количество классов увеличиваются операндом *B* блока DEPART. И, наконец, суммируются для расчета среднего и стандартного отклонений аргумента таблицы.

Объект *Qtable* может быть переопределен второй командой QTABLE с таким же именем, как и первая.

Пример

WaitTimes QTABLE WaitingLine,100,100,10

В этом примере распределение интервалов прихода-ухода записывается в таблицу с именем *WaitTimes*. Команда QTABLE создает таблицу 10 классами частоты.

Все временные интервалы до 100 составляют первый класс частоты. Это означает, что целое число первого класса частоты увеличено на 1, однако фактор оценки определяется операндом *B* блока DEPART. Добавления фактора оценки к классу частоты накладывает свои изменения. Фактор оценки применяется к среднему и стандартному отклонению и имеет такое же значение, как сложные записи в блоке DEPART.

Если временной интервал превышает 900, он будет помещен в 10 и последний класс частоты. Если временной интервал не попадает ни в первый, ни в последний класс частоты, будет выбран класс со 2 по 9.

Например, если временной интервал равен 290, будет обновлен 3 класс частоты.

Статистика, накопленная объектом *Qtable*, выводится в стандартной форме отчета.

Стандартный отчет рассматривается в главе 11.

Связанные SNA

- *QEntnum* – Текущее содержимое очереди. Текущее количество значение очереди *Entnum*.
- *QAEntnum* – Среднее по времени количество элементов в очереди *Entnum*.
- *QCEntnum* – Общее количество записей в очереди *Entnum*.
- *QMEntnum* – Максимальное значение очереди. Максимальное количество элементов в очереди *Entnum*.
- *QTEntnum* – среднее время задержки в очереди *Entnum*.
- *QXEntnum* – среднее время задержки в очереди исключая нулевые интервалы времени.
- *QZEntnum* – количество нулевых записей. Число записей в очереди *Entnum* с нулевым временем.

Ограничения памяти

Объем памяти, выделяемый для объекта *Qtable*, ограничен. Это описано в главе 2.

Связанные блоки

- QUEUE – отмечает статистику о времени начала и ожидания.
- DEPART – отмечает статистику времени конца ожидания, используя фактор оценки.

Связанные окна

- В окне таблиц отображаются объекты *Qtable*.
- Любые SNA отображаются в окне графиков или выражений.

REPORT

Команда REPORT немедленно создает отчет

REPORT A,B

Операнды

A – должен быть *Null*.

B – NOW, для записи стандартного отчета немедленно. Необязательный. Может быть NOW или *Null*.

Пример

REPORT

Эта команда говорит GPSS World о том, что нужно немедленно создать стандартный отчет.

REPORT ,NOW

Эта команда существует в целях совместимости. Как и предыдущая, она генерирует стандартный отчет немедленно.

Действие

Команда REPORT – команда очереди. При получении она помещается в очередь команд. Как только объект моделирования принимает команду REPORT, он немедленно создает стандартный отчет согласно настройкам «In Windows» параметров объекта моделирования. Эта опция находится на странице отчетов настроек модели.

Если опция «In Windows» установлена, объект моделирования создает объект отчета, содержащий новый стандартный отчет. Он может быть сохранен в файл или сброшен. Если опция «In Window» не установлена, создается стандартный отчет, и ему присваивается серийный номер и данные записываются в файл.

REPORT больше не использует операнд A и предполагает, что значения операнда B по умолчанию NOW. Он игнорирует опцию «Создать Стандартный Отчет», которая используется для автоматического создания отчета, а также игнорирует операнд B команды START данного процесса моделирования.

Обычно нет необходимости использовать команду REPORT. Все отчеты управляются автоматически, согласно конфигурации. Это описано в главе 11.

Связанные настройки

Стандартный отчет обычно отображается в окнах отчетов, но может быть направлен непосредственно в файл. Это контролируется опцией «In Windows» в настройках модели.

RESET

Команда RESET сбрасывает счетчики измерений.

RESET

Операнды

Отсутствуют

Действие

Команда RESET – команда очереди. При получении она помещается в очередь команд.

Команда RESET сбрасывает счетчики статистики, не удаляя заявки из процесса моделирования. Это полезно использовать для повторения процесса моделирования и отбрасывания данных, представляющих переходный период.

RESET устанавливает время начальных измерений равным текущему значению системных часов и инициализирует счетчики статистики для устройств, очередей и объектов памяти. Это позволяет запустить новое окно времени, используемое для сбора статистики.

Команда RESET не удаляет заявки из системы, как это делает CLEAR.

RESET не изменяет состояний генераторов случайных чисел, системного времени и количества заявок.

После выполнения команды RESET:

Время работы устройств, объектов памяти, очередей и пользовательских последовательностей устанавливается в 0.

Общие количества становятся равными текущим количествам.

Максимальные и минимальные значения устанавливаются согласно содержимому очередей, пользовательских последовательностей и объектов памяти.

Генераторы случайных чисел не сбрасываются.

Относительные часы (время, прошедшее с последней команды RESET) устанавливаются в 0.

Сбрасывается статистика в таблицах.

RMULT

Команда RMULT устанавливает начальное число генераторов случайных чисел.

RMULT A,B,C,D,E,F,G

Операнды

A – Начальное число для генератор случайных чисел 1. Необязателен. Должен быть *Null* или иметь тип *PosInteger*.

B – Начальное число для генератор случайных чисел 2. Необязателен. Должен быть *Null* или иметь тип *PosInteger*.

C – Начальное число для генератор случайных чисел 3. Необязателен. Должен быть *Null* или иметь тип *PosInteger*.

D – Начальное число для генератор случайных чисел 4. Необязателен. Должен быть *Null* или иметь тип *PosInteger*.

E – Начальное число для генератор случайных чисел 5. Необязателен. Должен быть *Null* или иметь тип *PosInteger*.

F – Начальное число для генератор случайных чисел 6. Необязателен. Должен быть *Null* или иметь тип *PosInteger*.

G – Начальное число для генератор случайных чисел 7. Необязателен. Должен быть *Null* или иметь тип *PosInteger*.

Пример

RMULT „111

В этом примере, генератор случайных чисел 3 инициализируется начальным числом 111.

Действие

Команда RMULT – команда очереди. При получении она помещается в очередь команд.

RMULT инициализирует до 7 генераторов случайных чисел. Только генераторы с номером 7 и ниже контролируются командой RMULT. Однако, по умолчанию, начальное значение генератора совпадает с номером объекта этого генератора, таким образом можно создавать любое количество генераторов случайных чисел, и выбирать начальное значение для каждого. После первого запуска процесса моделирования, числа управляются только с помощью команды RMULT на генераторах 1-7.

Можно выбирать, какой поток случайных чисел GPSS World будет использовать для расчета времени в блоках ADVANCE и GENERATE в блоках частичного режима TRANSFER и TRANSFER PICK и управлять планированием времени. Это управляется опциями на вкладке «Random» в настройках модели.

выберите **View / Settings / Model**

Затем выберите вкладку Random. Затем введите необходимое значение в соответствующее поле. По умолчанию Поток случайных чисел равен 1.

Как только GPSS World стал использовать когерентные мультипликативные генераторы, стало возможным управлять парам потоков случайных чисел для различных целей. Это производится сравнением начальных значений генераторов. Допустим, s – начальное значение первого генератора, тогда если указать второму генератору начальное значение 2147483647-s, он будет генерировать поток случайных чисел с противоположными свойствами.

Специальные ограничения

- Начальные значения генераторов должны быть положительными числами.

Связанные SNA

- *RNEntnum* – случайное число. *RNEntnum*. Возвращает случайное целое число в пределах 0-999 генератора *Entnum*.

Связанные окна

Окно ANOVA может выполнять анализ расхождений и вычислять доверительные интервалы.

SHOW

Команда **SHOW** вычисляет выражение и записывает результат в строку статуса.
SHOW X

Операнд

X – Выражение.

Пример

SHOW 2#LOG(Q\$Barber)

Это команда находит натуральный логарифм SNA Q\$Barber, возводит результат в квадрат и записывает его в строку статуса окна модели.

Действие

Команда **SHOW** – немедленная команда. При получении она сразу же выполняется.

Команда вычисляет выражение в контексте моделирования и записывает результат в строку статуса окна модели. А также посылает сообщения в журнал. Правила вычисления выражений можно найти в разделе 3.4. Выражения должны быть составлены согласно правилам элементарной алгебры. Можно использовать арифметические и логические операторы, приведенные в разделе 3.4.

Если в выражении используются SNA, они вычисляются для активной заявки. Если активная заявка отсутствует, так как не запущен процесс моделирования, будет выдано сообщение об ошибке. Имена, которым неопределенно значение нельзя использовать в выражении. Для определения значений следует использовать команду **EQU**, которая должна следовать до выражения.

Примеры

SHOW C1

Показывает системные часы в строке статуса.

SHOW 4#(SQR(2)+SIN(C1))

Показывает результат 4 сложений квадратного корня из 2 и синуса от системного времени.

SHOW N1^W\$Chair

Показывает результат увеличения числа заявок, вошедших в блок 1, до числа заявок ожидающих в блоке Chair.

Специальные ограничения

- Некоторые SNA не могут быть использованы без активной заявки.
- **SHOW** отображает до 38 символов и цифр.

Связанные окна

- Любое правильно выражение отображается в окне графиков или выражений.

START

Команда **START** начинает процесс моделирования.
START A,B,C,D

Операнды

A – Количество проходов. Обязателен. Должен иметь тип *PosInteger*.

B – Операнд вывода. Значение NP - «не печатать». По умолчанию, печатать стандартный отчет. Необязателен. Может иметь значения NP или *Null*.

C – Не используется. Существует для совместимости со старыми версиями GPSS World.

D – последовательность печати. 1 для включения CEC и FEC в стандартный отчет. Необязателен. Должен быть *Null* или иметь тип *PosInteger*.

Действие

Команда **START** – команда очереди. При получении она помещается в очередь команд.

START используется для установки начала процесса моделирования. Процесс моделирования не закончится, пока количество проходов, заданное операндом А, не станет равно нулю или отрицательным. Блок TERMINATE используется для уменьшения количества проходов.

Процесс моделирования может завершиться раньше, если выполнится команда HALT или будет выполнено условие остановки.

Операнды В и D используются для контроля генерации отчета. Если операнд В опущен, будет записан стандартный отчет. Если операнд В имеет значение NP, которая значит «не печатать», отчет не будет записан. Если операнд D ненулевой, будет выдан отчет о состояниях текущей последовательности событий (SEC) и последовательности будущих событий (FEC). В противном случае они в отчет внесены не будут. Подробное описание контроля над содержанием отчета можно найти в главах 11 и 12.

Операнд С существует для совместимости со старыми версиями GPSS. Он использовался для периодического создания отчета. Эта функция доступна командами START и REPORT.

При выполнении команды START устанавливается количество проходов.

Все блоки генерации, отмеченные как «незапущенный» снабжаются одной заявкой.

Генераторы случайных чисел не сбрасываются.

Если время имеет значение 0, выполняется сброс статистических счетчиков. См. описание команды RESET.

Пример

START 1000,1

В этом примере, количество проходов устанавливается в 1000 и начинается процесс моделирования. Когда количество проходов достигает 0 или отрицательного значения (через блоки TERMINATE), записывается стандартный отчет с включением информации о последовательности текущих событий (SEC) и последовательности будущих событий (FEC).

Связанные SNA

- TG1 – количество проходов.

Связанные блоки

- TERMINATE – удаляет заявку и уменьшает количество проходов.

STEP

Команда STEP заставляет процесс моделирование обработать определенное число объектов.

STEP A

Операнд

А – количество блоков. Обязателен. Должен быть положительным целым числом, более точно *PosInteger*.

Пример

STEP 1

Команда говорит системе обработать только один блок, затем остановиться. Процесс моделирования переходит в состояние остановки.

Действие

Команда STEP – команда очереди. При получении она помещается в очередь команд.

Команда говорит системе, что нужно смоделировать только определенное количество блоков. Когда система достигает указанного блока, объект моделирования посылает сообщение строке статуса и в окно журнала, если оно открыто. Сообщение передает номер активной заявки, текущий блок активной заявки, и следующий блок, в который должна попасть заявка.

Процесс моделирования, запущенный командой STEP не останавливается, когда количество проходов становится равным 0, моделирование заканчивается, когда достигнуто указанное число блоков.

Когда выполняется команда STEP.

Количество проходов не устанавливается.

Инициализируется каждый блок, помеченный как «не запущенный».

Генераторы случайных чисел не сбрасываются.

Горячие клавиши

Команда STEP 1 может быть вызвана сочетанием клавиш [Ctrl] + [Alt] + [1]. Окно GPSS World должно быть активно в этот момент.

Связанные окна

- Шаг моделирования отображается в окне блоков.

STOP

Команда STOP устанавливает или удаляет условие остановки.

STOP A,B,C

Операнды

A – число заявки. Должен быть положительным целым числом. Если операнд опущен, любая заявка будет удовлетворять условию. Необязателен. Должен быть *Null* или *PosInteger*.

B – число блока. Если операнд опущен, любой блок будет удовлетворять условию. Необязателен. Должен быть *Null*, *Name* или *PosInteger*.

C – ON или OFF. Если операнд C опущен, предполагается, что он равен ON. Необязателен. Операнд должен быть ON, OFF или *Null*.

Пример

STOP 100,52

Команда добавляет условие остановки, которое значит, что процесс моделирования остановится, когда 100 заявок пройдут через блок 52.

Действие

Команда STOP – команда очереди. При получении она помещается в очередь команд.

STOP с опцией ON добавляет условие остановки, но не запускает процесс моделирования. Должны быть введены команды START, STEP или CONTINUE.

Когда в процессе моделирования выполняется условие остановки, объект моделирования посылает сообщение в строку статуса и в окно журнала, если оно открыто. Сообщение содержит номер активной заявки, текущий блок активной заявки и следующий блок, в который должна попасть заявка.

Когда производится остановка по условию, это условие остается в силе. Если будет дана команда CONTINUE, она проигнорирует текущее условие, но не отменит его. И при выполнении этого условия снова, будет опять произведена остановка. Отменить условие можно, дав команду STOP с опцией OFF. Условие также удаляется, когда вся модель отработана.

Команды STOP можно подавать мышкой в окне блоков. Это описано в главе 5.

Если опущен операнд A, любая заявка будет удовлетворять условию, если опущен операнд B, любой блок будет удовлетворять условию. Команда STOP без параметров остановит процесс моделирования немедленно.

Может быть установлено любое число условий остановки.

Команда STOP с опцией OFF отменит все условия остановки, которые удовлетворяют операндам A и B.

Примеры

STOP

Без параметров, такая команда приведет в немедленной остановке процесса моделирования.

STOP „OFF

Эта команда отменяет все условия остановки.

STOP 2

Эта команда остановит процесс моделирования, когда придут две активные заявки.

STOP ,Chair

Эта команда остановит процесс моделирования, когда в блок Chair придет заявка.

STOP ,Chair,OFF

Эта команда отменит все условия, которые содержать блок Chair.

Связанные окна

Для установки условий остановки можно использовать мышку, нажимая определенные кнопки в окне блоков.

STORAGE

Команда STORAGE определяет объект памяти.

NAME STORAGE A

Название/Операнды

NAME – название объекта. Обязательно.

A – емкость памяти. Обязателен. Должен быть *PosInteger*.

Пример

MotorPool STORAGE 20

Команда определяет объект памяти с именем MotorPool с емкостью 20 ячеек.

Действие

Команда STORAGE – команда очереди. При получении она помещается в очередь команд. Команда определяет объект памяти. Когда заявка поступает в блок ENTER, запрашивается свободное количество памяти для хранения заявки. Если памяти хватает, заявка входит в блок ENTER, а свободное место в памяти уменьшается. Если заявка не может поместиться, она переходит в режим ожидания в последовательность задержек объекта памяти. Более подробно об объектах памяти написано в главе 4.

Объект памяти может быть переопределен командой STORAGE с таким же именем.

Если вы хотите обращаться к объекту памяти, по номеру, а не по имени, необходимо определить его командой EQU. Это необходимо, если вы хотите обращаться к объектам памяти в блоках SELECT или COUNT.

Связанные SNA

- *REntnum* – используемая емкость. Свободное место, доступное для заявок в объекте памяти *Entnum*.
- *SEntnum* – Память в пользовании. Возвращает количество занятой заявками памяти объекта *Entnum*.
- *SAEntnum* – Среднее использование памяти. Возвращает среднее по времени количество использованной памяти объекта *Entnum*.
- *SCEntnum* – Общее количество ячеек памяти для хранения заявок объекта *Entnum*.
- *SEEntnum* – Память свободна. Возвращает 1, если память *Entnum* полностью свободна, иначе 0.
- *SFEntnum* – Память заполнена. Возвращает 1, если память *Entnum* полностью заполнена, иначе 0.
- *SREntnum* – использование памяти. Часть от общего использования, представленное в среднем значении использования памяти *Entnum*. *SREntnum* выражено в частях/тыс. и возвращает реальное значение между 0-1000 включительно.
- *SMEntnum* – Максимальное значение использования памяти *Entnum*.
- *STEntnum* – Среднее время в ячейке памяти *Entnum*.

- *SVEntnum* – память в состоянии доступности. Возвращает 1, если память доступна, иначе 0.

Связанные блоки

- ENTER – занимает или ожидает свободного места памяти.
- LEAVE – освобождает место в памяти.

Связанные окна

- Объекты памяти видны в окне Памяти.
- В окнах графиков и выражений отображаются SNA.

TABLE

Команда TABLE инициализирует таблицу частот распределения.
NAME TABLE A,B,C,D

Название/Операнды

NAME – Название объекта. Обязательно. Может быть длиной до 32 символов.

A – Аргумент. Необязателен. Элемент данных, частота распределения которого будет занесена в таблицу. Должен иметь тип *Name*, *Number*, *String*, *ParenthesizedExpression*, или *SNA*. Игнорируется ANOVA, но должен быть определен при использовании блоков TABULATE.

B – Верхний предел первого класса частоты. Максимальное значение, которое попадает в первый класс. Обязателен. Должен быть типа *Number* или *String*.

C – Размер классов частоты. Разность между верхним пределом и нижним пределом каждого класса частоты. Обязателен. Должен быть типа *Number* или *String*.

D – Число классов частоты. Обязателен. Должен быть типа *PosInteger*.

Действие

Команда TABLE – команда очереди. При получении она помещается в очередь команд.

Команда TABLE используется для определения GPSS таблицы, которая хранит частоту распределения, его относительные частоты, среднее и стандартное отклонение.

Главная цель – это сбор данных, Таблица может быть использована для сведения остатков для анализа расхождений. Специальное назначение рассмотрено ниже.

Для сбора элемента данных, заявка должна попасть в блок TABULATE, который ссылается на Таблицу. При входе в блок TABULATE рассчитывается аргумент (операнд A команды TABLE). Если он меньше или равен значению операнда B команды TABLE, выбирается первый класс частот. Если аргумент не попадает в первый класс, то класс выбирается делением значения аргумента на значения операнда C. Нижний предел класса частоты включен в предыдущий класс частоты. Если таблица недостаточна большая, чтобы принять это значение, выбирается последний класс частоты.

Затем, целое число в выбранном классе увеличивается на значение операнда B. По умолчанию приращение равно 1.

Наконец, обновляются счетчики для стандартного и среднего отклонения.

Таблица может быть переопределена или инициализирована заново с помощью команды TABLE с таким же именем, как и старая таблица.

Пример

SalesTable TABLE P\$Price,9.95,10,10

В этом простом примере распределение параметра Price заявок записывается в таблицу SalesTables. Команда TABLE создает таблицу с 10 классами частот.

Все значения переменной Price меньше 9.95 попадают в первый класс частоты. Это значит, что класс частоты увеличивается на 1. Однако в операнде B блока TABULATE доступен фактор оценки. Добавление фактора оценки к текущему числу соответствующего класса частоты накладывает свой отпечаток. Фактор оценки, применяется к среднему и стандартному отклонению и имеет такой же эффект, как множественные записи в блоке TABULATE. Если значение параметра Price превышает 89.95, выбирается 10 и последний класс частоты. Если значение параметра Price не

попадает ни в первый, ни в последний класс частоты, будет выбран свободный класс с 2 по 9.

Например, если значение Price равно 29.49, будет выбран третий класс.

Статистика, накопленная в таблице, выводится в стандартном отчете. Пример можно посмотреть в главе 11.

Остатки ANOVA

Когда результирующая матрица передается в библиотечную процедуру ANOVA, можно поставить автоматическую запись остатков анализа. Чтобы сделать это, необходимо всего лишь определить таблицу с таким же именем, как и матрица, добавив суффикс «_RESIDUALS». Например, если вы планируете использовать

ANOVA(MainResult, 3, 1)

нужно определить таблицу:

MainResult_Residuals TABLE ,-5,.5.20,

передаваемую в вызов ANOVA.

Когда производятся такие вещи, процедура ANOVA игнорирует операнд A таблицы и происходит сведение остатков от анализа результирующей матрицы. Таблица ANOVA и ячейки статистики заносятся в окно журнала. Можно просмотреть остатки, открыв окно таблиц в таблице MainResult_Residuals.

Ограничения памяти

- Имеется ограничение на выделение памяти вод объекты таблиц. Об этом написано в главе 2.

Связанные SNA

- *TBEntnum* – среднее количество записей таблицы *Entnum*.
- *TCEntnum* – Количество записей в таблице *Entnum*.
- *TDEntnum* – стандартное отклонение в таблице *Entnum*.

Связанные блоки

- TABULATE – регистрирует изменение объекта данных в таблице.

Связанные окна

- Таблицы отображаются в окне таблиц (Table Window).
- В окнах графиков и выражений отображаются SNA.

VARIABLE

Команда VARIABLE определяет арифметическую переменную.

NAME VARIABLE X

Название/Операнд

NAME – название объекта. Обязательно.

X – выражение. Выражения описаны в разделе 3.4.

Пример

Var1 VARIABLE 5#LOG(Q\$WaitingLine)

Эта команда определяет переменную, значение которой будет вычислено при обращении к V\$Var1. Когда вызывается SNA, вычисляется значение, определенное в переменной Var1 и возвращается результат.

Действие

Команда VARIABLE– команда очереди. При получении она помещается в очередь команд.

Когда обрабатывается команда VARIABLE, объект моделирования создает или переопределяет переменную. Значение созданной переменной вычисляется тогда, когда система числовых атрибутов класса V имеет ссылку на эту переменную.

Значение выражения переменной вычисляется согласно правилам элементарной алгебры. Формальное определение можно найти в приложении. Можно использовать любые арифметические и логические операторы, приведенные в разделе 3.4. Если в выражении используются SNA, оно вычисляется для активной заявки. Именно значение,

неопределенное однозначно не может быть использовано в выражении. Их необходимо определить до выражения. Присвоение значений переменным производится командами EQU или с помощью PLUS процедур.

Созданные переменные не удаляются из системы. Однако, они могут быть переопределены командой VARIABLE.

При моделировании не в режиме совместимости GPSS/PC, команда FVARIABLE и VARIABLE обрабатываются одинаково.

Совместимость GPSS/PC

- Все SNA отбрасываются в этом режиме.
- Все промежуточные результаты вычислений выражения переменной отбрасываются.

Связанные SNA

- VEntnum – возвращает результат вычисления арифметической переменной Entnum.

Связанные окна

- В окнах графиков и выражений отображаются SNA.

Глава 7 – Блок выражений

Блок выражений используются для создания блока объектов. Блок выражений, который является частью Начальной трансляции модели, создает постоянный блок моделирования. Блок выражений создает одноразовый временный блок, который используется в режиме *ручного моделирования*. Подробнее об этом написано в разделе 2.3.

Модель – это простая последовательность Выражений модели. Выражение модели GPSS World является или GPSS выражением, или определением PLUS процедуры. GPSS выражения это или блоки выражений, или просто команды. Любое выражение модели может быть выполнено в текущем процессе моделирования как интерактивное выражение.

Глава содержит информацию о каждом блоке выражения, поддерживаемом GPSS World.

Блок выражений:

ADOPT – Изменяет установку сборки.

ADVANCE – Помещает заявку на последовательность будущих событий..

ALTER – Проверяет и модифицирует заявку в группе.

ASSEMBLE – Ждет и уничтожает указанные заявки..

ASSIGN – Изменяет параметр заявки.

BUFFER – Помещает заявку на последовательность текущих событий согласно приоритету.

CLOSE – Закрывает поток данных.

COUNT – Присваивает параметру заявки значение количества объектов.

DEPART – Удаляет заявку из очереди.

DISPLACE – Изменяет следующий блок для заявки.

ENTER – Занимает память или ждет ее освобождения.

EXAMINE – Проверка группы членов.

EXECUTE – Выполняет операцию определенную блоком.

FAVAIL – Изменяет состояние устройства на "доступно".

FUNAVAIL - Изменяет состояние устройства на "не доступно".

GATE – Проверяет объект и изменяет поток заявки.

GATHER – Ждет заявку.

GENERATE – Создает заявку и помещает ее в последовательность будущих событий..

INDEX – Изменяет параметр заявки.

INTEGRATION – Включает или выключает интегрирование переменных.

JOIN – Добавляет член в группу заявок или чисел.

LEAVE – Освобождает память.

LINK – Перемещает заявку в пользовательскую последовательность.

LOGIC – Модифицирует логический переключатель.

LOOP – уменьшает значение параметра, перескакивает в другой блок, если результат ненулевой.

MARK – Присваивает параметру заявки значение текущего системного времени.

MATCH – Ожидает попадания заявки в сопряженный блок MATCH.

MSAVEVALUE – Присваивает значение элементу матрицы.

OPEN – Открывает поток данных.

PLUS – Вычисляет PLUS выражение и сохраняет результат в параметр.

PREEMPT – Замещает заявку в устройстве.

PRIORITY – Изменяет приоритет заявки.

QUEUE – Добавляет заявку в очередь.

READ – Читают следующую строку из потока данных.

RELEASE – Освобождает устройство.

REMOVE – Удаляет заявку или число из группы.

RETURN – Освобождает устройство.

SAVAIL – Изменяет состояние памяти на "доступное".
SAVEVALUE – Запоминает значение в хранилище.
SCAN – Проверяет группу заявки, помещает значение в параметр.
SEEK – Перемещается на указанную строку в потоке данных.
SEIZE – Занимает устройство или дожидается его освобождения.
SELECT – Помещает выбранный номер объекта в параметр заявки.
SPLIT – Создает связанную заявку.
SUNAVAIL – Изменяет статус памяти на "не доступный".
TABULATE – Обновляет таблицу.
TERMINATE – Уничтожает заявку, уменьшает количество проходов.
TEST – Проверяет арифметическое условие и изменяет поток заявки.
TRACE – Устанавливает индикатор маршрута активной заявки.
TRANSFER – Перемещает в указанный блок.
UNLINK – Удаляет заявку с пользовательской последовательности.
UNTRACE – Выключает индикатор маршрута активной заявки.
WRITE – Записывает значение в поток данных.

Операнды

Операторы обычно содержат один или более операндов, которые Вам необходимо заполнить. Большинство операндов имеют несколько различных корректных форм. В описаниях, которые следуют далее, корректный класс операндов описан курсивом. Вы должны выбрать член класса и вписать его в поле операнда. Например, если одна из правильных форм операнда дана как *PosInteger*, вы выбрали тип: курсив обычно привлекает внимание, но Вы можете сделать ссылку на формальное определение в Приложении.

Окна

Для вас доступно широкое разнообразие окон, чтобы наблюдать собрание блоков и их эффектов на другие объекты в ваших моделях. Вообще, windows специализированы на типы объекта.

- **Model Window** - текстовое представление – полноэкранный текстовый редактор модели.
- **Block Input Window** - Drag and Drop создание модели.
- **Journal Window** - записывает события сеанса.
- **Blocks Window** - сетевое представление блочной динамики.
- **Expressions Window** - сетевое представление значений выражений.
- **Facilities Window** - сетевое представление динамики объекта средства.
- **Logicswitches Window** - сетевое представление динамики объекта Logicswitch.
- **Matrix Window** - сетевое представление динамики матричного поперечного сечения.
- **Plot Window** - сетевое представление графиков из 8 выражений.
- **Queues Window** - сетевое представление динамики объекта очереди.
- **Savevalues Window** - сетевое представление динамики объекта Savevalue.
- **Storages Window** - сетевое представление динамики объекта памяти.
- **Table Window** - сетевое представление динамики объекта таблицы.
- **Transaction Snapshot** - изображение части заявки моделирования.
- **CEC Snapshot** - изображение части цепочки текущих событий моделирования.
- **FEC Snapshot** - изображение части будущей цепочки событий моделирования.
- **Numeric Groups Snapshot** - изображение части числовых групп моделирования.
- **Userchains Snapshot** – изображение части пользовательских цепочек моделирования.
- **Transaction Groups Snapshot** - изображение части групп заявок моделирования.

ADOPT

Используйте блоки ADOPT, чтобы изменить набор активной заявки.

ADOPT A

Операнд

A – Сборка необходимых требований. Операнд должен быть *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

Пример

ADOPT 2000

Набор требований активной заявки принимает значение 2000. В действительности, активная заявка становится членом набора требований 2000.

Действие

Когда заявка введена в ADOPT блок, операнд A оценен в цифровой форме, и сокращен. Если результат меньше или равен нулю, происходит остановка из-за ошибки(Error Stop). Иначе, результат устанавливается в значение набора требований активной заявки.

Каждая заявка установлена при создании в набор требований. Для заявок, созданных в GENERATE блоке, в качестве начального значения присваивается то же самое число, что использовалось для набора требований и количества заявок. Для заявок, созданных в блоках SPLIT, образованные заявки помещены в тот же самый набор требований как и те заявки, от которых они образованы.

Наборы требований используют, чтобы объединит связанные заявки в ASSEMBLE и GATHER блоки. ADOPT блок предусматривает простое управление присваиванием набора требований.

Специальное ограничение

- необходимо быть положительным.

Режим отказа

Заявки никогда не отвергаются при вводе в блок ADOPT.

Связанные блоки

- **ASSEMBLE** - Ожидают и выводят из строя состав набора требований.
- **GATHER** - Ожидают состав набора требований.
- **MATCH** - Ожидают состав набора требований.
- **SPLIT** - Создает заявку (заявки) в том же самом наборе требований.

Связанные протоколы SNA

- **A1** - Набор требований. Возвратить набор требований активной заявки.
- **MBEntnum** - Соответствие в блоке. MBEntnum возвращает 1, если есть заявка в блоке Entnum. Он находится в том же самом наборе требований как активная заявка. Иначе MBEntnum возвращает 0.

Связывающее окно

- **Transaction Snapshot** - Изображение части заявки в моделировании.

ADVANCE

Блок ADVANCE задерживает продвижение заявки за указанное время моделирования.

ADVANCE A, B

Операнды

A - Среднее приращение времени. Дополнительное. Операнд должен быть *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA* or *SNA*Parameter*. Значение по умолчанию 0.

B – Половина диапазона времени или функция, функциональный модификатор. Дополнительный. Операнд должен быть *Null*, *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

Пример

ADVANCE 101.6,50.3

В этом примере создается блок, который выбирает случайное число между 51.3 и 151.9, включительно (то есть 101.6 плюс или минус 50.3), и задерживает ввод заявки за количество времени моделирования.

Действие

Блок ADVANCE вычисляет приращение времени и помещает заявку ввода в будущую цепочку событий (FEC) за это время моделирования.

Приращение времени может быть вычислено несколькими способами. Если определен только A операнд, он используется как приращение времени. Если A и B операнды присутствуют, и B не определяет функцию, и A и B оценены в числовой форме и случайное число между A-B, и A+B, включительно, используется как приращение времени. Вы можете выбрать, какое число генератора случайных чисел должно использоваться как источник случайного числа. Устанавливается "Случайно" на странице модели ПК параметрах настройки.

Выберите View / Settings / Model

После выберите страницу Random. Затем заполните количество объектов потока случайных чисел в отведенном " ADVANCE " блоке входа. Инсталляционное значение по умолчанию должно использовать поток случайных чисел, номер 1.

Если B - протокол SNA класса FN, названный функциональным модификатором, оценка B умножена на результат оценки операнда; продукт используется как приращение времени.

Если нуль вычислен как приращение времени (ADVANCE 0), Заявка ввода помещена в цепочку текущих событий перед одноранговым приоритетом. Такой блок ведет себя как нулевая операция. Дальнейшее рассмотрение цепочки текущих событий найдете в главе 9.

Специальное ограничение

- Если приращение времени вычислено как отрицательное значение, происходит Error Stop.

Режим отказа

Обычно, заявкам не отказывают при вводе в блок ADVANCE. Однако, так как выгруженным заявкам не разрешено присутствовать на будущей цепочке событий, выгруженные заявки не разрешено вводить в блок ADVANCE, если приращение времени будет положительным (отличным от нуля).

Когда заявки отказываются введенными, их индикатор задержки установлен и сохраняется, пока заявка не будет введена "Simultaneous" TRANSFER режим блока. Чтобы координировать часть множественных объектов, лучше использовать TEST блок и BOOLEAN VARIABLE вместо блока TRANSFER SIM.

Связывающие окна

- **Transaction Snapshot** - Изображение части заявок моделирования.
- **CEC Snapshot** - Изображение части цепочки текущих событий моделирования.
- **FEC Snapshot** - Изображение части будущей цепочки событий моделирования.

ALTER

Блок ALTER изменяет приоритет или параметр выбранных членов группы заявок.

ALTERO A, B, C, D, E, F, G

Операнды

O - Условный оператор. Чтобы происходило чередование отношения E к F. Этот выбор описан ниже. Дополнительный. Оператор должен быть *Null*, E, G, GE, L, LE, MAX, MIN, or NE

A - Группа заявок. Группа, участники которой будут проверены на чередование. Необходимая. Операнд должен быть *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

B - Предел. Максимальное число заявок, которые будут изменены. Значение по умолчанию - ALL. Дополнительный. Операнд должен быть PR, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

С - Измененный атрибут. Параметр члена заявки, который будет проверен или PR, чтобы изменить приоритет члена заявки. Операнд должен быть PR, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

D – Замены значений. Значение, которое заменит атрибут С. Required. Операнд должен *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

E – Значение для тестирования. PR или номер Параметра. Параметр члена заявки, который определяет, должна ли каждая группа членов заявок быть изменена или PR, чтобы использовать для определения приоритет заявки. Это оценка группы членов заявок. Дополнительная. Операнд должен быть PR, *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

F - Значение ссылки. Значение, с которым сравнивается операнд E. Оценено относительно активной заявки Дополнительный. Операнд должен быть *be Null*, *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

G - Дополнительный номер блока. Дополнительный адресат для активной заявки. Дополнительный. Операнд должен быть *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

Примеры

ALTER Inventory,ALL,Price,49.95

В этом простом примере у всех заявок в группе заявок под именем Inventory есть свой параметр под названием Price set, равный 49.95.

ALTER NE Bin7,10,Price,49.95,PartNum,99.95,Out

В этом примере, когда заявка вводится в ALTER блок, группа заявок Bin7 просканирована для заявок, которые не имеют значения 99.95 в своих параметрах по PartNum. Первые 10 заявок, которые проходят тестирование содержат значение их параметра Price set to 49.95. Если не найдены 10 заявок, которые проходят тестирование, заявка вводит блок метки Out. Иначе говоря, переходит к следующему последовательному блоку.

Действие

ALTER блок выбирает заявку в группе заявок и изменяет один из атрибутов каждой из них. Когда группа членов заявок изменена, ее операционному атрибуту, определенному операндом С, присваивают значение, определенное операндом D. Измененные транзакции не перемещены от их контекста. Однако, заявка, вводящая ALTER блок, может быть переадресована согласно операнду Г.

Если Вы не используете условный оператор, или операнды E или F, все заявки (операнд В) изменены до предела. В этом случае, отсутствует приоритет параметра тестирования, принимающий решение изменить атрибут части заявки.

Если Вы используете операнды E, F, или условный оператор, каждый член группы должен пройти тестирование прежде, чем будет изменен. Операнд E определяет, какой атрибут заявки участника должен быть проверен. Это может быть сравнение с минимумом или максимумом всех подобных атрибутов члена группы при использовании MIN или MAX как условного оператора. Изменены все заявки, которые проверены и имеют максимальный или минимальный атрибут. В этом случае, Вы не должны использовать операнд F.

Вы можете использовать условный оператор, чтобы определить отношения между операционным атрибутом (операнд E) и значением ссылки (операнд F), который инициализирует чередование заявок. Для равенства значение по умолчанию для условного оператора - E. Если вы не используете условный оператор, но вы используете операнд E и операнд F, значения должны быть равными для атрибута члена заявки, который будет изменен.

Вы можете сравнить атрибут члена Группы с операндом F, с или без условного оператора. В этом случае, условный оператор не должен быть MIN или MAX. Операнд E всегда при

тестировании обращается к члену группы. Однако, если любой другой операнд, связанный протоколом SNA заявок, оценен относительно заявки ввода.

Операнд В отключает просмотр Группы, когда он равняется числу заявок, которые были изменены. Значение по умолчанию – ALL. Если отсутствует тест атрибута, то есть, если Е не определен, заявки изменяются, пока счет чередования не станет равным В или пока вся Группа не будет проверена.

Операнд Г указывает дополнительный блок адресата, который будет принят заявкой ввода, при выполнении условия. Операнд Г используется для следующего блока при условиях:

- Никакая заявка не изменена.
- Счет измененных заявок, определенных В, не может быть достигнут.
- Если операнд Г не используется, заявка ввода всегда переходит в следующий Sequential блок.

Условные операторы

Условный оператор может быть Е, G, GE, L, LE, MAX, MIN, or NE. Если отсутствует используемый условный оператор, Е (равенство) принят. Когда условие - истина, проверяемая заявка изменена. Условия определены следующим образом:

- **Е** - Атрибут члена заявки, определенный операндом Е, должен быть равным значению ссылки, определенной операндом F для члена заявки участника, которая будет изменена.
- **G** - Атрибут члена заявки, определенный операндом Е, должен быть больше чем значение ссылки, определенной операндом F для члена заявки, которая будет изменена.
- **GE** - Атрибут члена заявки, определенный операндом Е, должен быть больше или равным значению ссылки, определенной операндом F для члена заявки, которая будет изменена.
- **L** - Атрибут члена заявки, определенный операндом Е, должен быть меньше чем значение ссылки, определенной операндом F для члена заявки, которая будет изменена.
- **LE** - Атрибут члена заявки, определенный операндом Е, должен быть меньше или равным значению ссылки, определенной операндом F для члена заявки, которая будет изменена.
- **MAX** - Атрибут члена заявки, определенный операндом Е, должен быть равным наибольшему такому атрибуту всех заявок в группе для членов заявки, которая будет изменена.
- **MIN** - Атрибут члена заявки, определенный операндом Е, должен быть равным наименьшему такому атрибуту всех заявок в группе для членов заявки, которая будет изменена.
- **NE** - Атрибут члена заявки, определенный операндом Е, должен быть неравным значению ссылки, определенной операндом F для члена заявки, которая будет изменена. Если никакой условный оператор не используется, Е принят.

Специальные ограничения

- Если использовался операнд Е, то вы должны использовать операнд F, или иначе условный оператор MIN or MAX.
- Если использовался операнд F, вы должны использовать операнд Е, и не должны использовать условный оператор MIN or MAX.
- Если используется условный оператор MIN or MAX, должен использоваться операнд Е, и не должен использоваться операнд F.

Режим отказа

Заявкам не бывает отказано во вводе в ALTER блок.

Связанные блоки

Заявки добавлены к группам заявок, и числа добавлены к числовым группам JOIN блока. На заявки в группах можно сослаться через ALTER, EXAMINE, REMOVE и SCAN блоки. На числа в числовых группах можно сослаться через EXAMINE и REMOVE блоки.

Связанный протокол SNA

- **GTEntnum** - Счет группы заявок. GTEntnum возвращает количество членов группы заявок Entnum.

Связывающие окна

- **Transaction Snapshot** - Изображение части заявки в моделировании.
- **Transaction Groups Snapshot** - Изображение части групп заявок моделирования.

ASSEMBLE

Ожидание заявок и вывод из действия.

ASSEMBLE A

Операнд

A – Расчет заявки. Необходимый. Операнд должен быть *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

Пример

ASSEMBLE 2

Это самый простой способ использовать ASSEMBLE блок. Первая заявка набора требований (см. Раздел 9.3) поставлена в ожидание, когда будет введен ASSEMBLE блок. Когда другая заявка в том же самом наборе требований вводит блок, он разрушен, и заявки ожидания продолжают.

Действие

Когда заявка вводит ASSEMBLE блок, цепочка соответствия блока просматривает заявку, ожидания тот же набор требований. Если нет никаких других членов того же самого набора требований, операнд оценен, сокращен, уменьшен на один и сохранен в памяти в заявке. Если это число - ноль, заявка немедленно пытается ввести следующий последовательный блок. Иначе заявка помещена в очередь, прикрепленную к ASSEMBLE блоку, названному Match Chain, чтобы ожидать прибытия других членов набора требований.

Если заявка ожидания найдена при вводе ASSEMBLE блока, заявка ввода разрушена и расчет заявок, который был сохранен в цепочке заявок, уменьшается на один. Когда этот счет становится равным 0, заявка ожидания удалена от цепочки Match. Если эта заявка не была выгружена в Facility, она пытается ввести следующий последовательный блок.

Выгруженным заявкам, которые завершают требования в ASSEMBLE блоке, не разрешается оставить блок, пока все прерывания по приоритету не будут очищены. Рассмотрение механизма прерывания по приоритету найдете в Разделе 9.4. Выгруженные заявки, которые были удалены от цепочки Match, не участвуют в более поздних требованиях, даже если они остаются в ASSEMBLE Блоке.

ASSEMBLE блоки, отличаются от GATHER тем, что они последующих заявках разрушаются при переводе.

Специальное ограничение

- Заявки, которые в настоящее время резервируются, не разрешаются покидать ASSEMBLE блоки.

Режим отказа

Заявкам не бывает отказано во вводе в ASSEMBLE блок..

Связанные блоки

- **ADOPT** - Набор групп требований активной заявки.
- **GATHER** - Ожидание членов набора требований.
- **MATCH** - Ожидание членов набора требований.
- **SPLIT** - Создает заявку в том же наборе требований.

Связанные протоколы SNA

- **A1** - набор требований. Возвращает набор требований активной заявки.
- **MBEntnum** - Match блок. *MBEntnum* возвращает 1, если есть заявка в блоке *Entnum*, который находится в том же наборе требований, что и активная заявка. Иначе *MBEntnum* возвращает 0.

Связывающие окна

- **Blocks Window** - Сетевое представление блочной динамики.
- **Transaction Snapshot** - Изображение части заявки в моделировании.
- **CEC Snapshot** - Изображение части цепочки текущих событий в моделировании.
- **FEC Snapshot** - Изображение части будущей цепочки событий в моделировании.

ASSIGN

ASSIGN блок, используются, чтобы поместить или изменить значение в операционном параметре.

ASSIGNA, B, C

Операнды

A - Количество параметров активной заявки. Необходимый. Операнд должен быть *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*, followed by +, -, or *Null*.

B - Значение. Необходимый. операнд должен быть *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

C - Функциональное число. Дополнительный. Операнд должен быть *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* or *SNA*Parameter*.

Примеры

ASSIGN 2000,150.6

Это - самый простой способ использовать ASSIGN блок. Значение 150.6 направлено на параметр номер 2000 заявки ввода. Если такой параметр не существует, то он создается.

ASSIGN TEXT,"Look on my works, ye Mighty, and despair."

В этом примере строка направлена на параметр активной заявки под TEXT. Если такой параметр не существует, то он создается.

ASSIGN 2000+,-3

В этом примере [+] после операнда A указывает, что значение операнда B должно быть добавлено к оригинальному значению параметра. Этот блок добавляет-3 к значению, содержащемуся в операционном параметре 2000. Если такого операционного параметра нет, то каждый создан и обнулен перед суммированием. В этом случае, значение операционного параметра становится-3.

ASSIGN 2000,-,3

В этом примере [-] после операнда A указывает, что значение операнда B должно быть вычтено из оригинального значения параметра. Этот блок вычитает-3 из значения, содержащегося в операционном параметре 2000. Если такого операционного параметра нет, то каждый создан и обнулен перед вычитанием. В этом случае, значение операционного параметра становится 3.

Действие

Когда заявка вводится в ASSIGN блок, значение операционного параметра, идентифицированного в операнд, установлено согласно B и C операндами. Операционный параметр создан в случае необходимости.

Вы можете назначать, прибавлять или вычитать из числового эквивалента значения операционного параметра. Если отсутствует операнд C, операнд B оценен и используется как новое значение, или его числовой эквивалент используется как приращение или декремент. Сложение и вычитание определены + или - суффиксом после операнда. Если нет такого суффикса, операнд B оценен, и результат передан значению операционного параметра.

Произвольно, операнд C может использоваться, чтобы определить число функции, здесь это " Function Modifier ". Если определен, то функция оценена, умноженным числовым эквивалентом оцененного операнда B, и результат прибавлен, вычтен, или назначен

значением операционного параметра в зависимости от дополнительного суффикса операнда. Обратите внимание, что операнд C определяет функциональное число объекта или название (не должно быть перед ним FN или FN\$). Если протокол SNA класса FN используется, функция GPSS оценена, и результат используется, чтобы определить вторую функцию GPSS, которую тоже необходимо оценить.

Специальное ограничение

- Должна быть положительной, но может сопровождаться + или - суффиксами.

Режим отказа

Заявкам не бывает отказано во вводе в ASSIGN блок.

Связанный протокол SNA

Parameter или *Parameter - значение параметра. PParameter возвращает значение параметра Parameter. (примечание: например. P1 или *1 или P\$NAME приведет к значению в параметре 1 в первых двух случаях и в конечном случае вызванном параметре NAME).

Связывающее Окно

- **Transaction Snapshot** - изображение части заявок в моделировании.

BUFFER

BUFFER блок помещает активную заявку в цепочку текущих событий после проверки ее одноранговых приоритетов.

BUFFER

Операнды

отсутствуют

Пример

BUFFER

Этот пример создает блок, который дает заявкам, кроме активной заявки, возможность, быть запланированными.

Действие

Когда заявка вводится в BUFFER блок, она помещена в цепочку текущих событий после заявок равного приоритета.

Операционный планировщик пытается переместить активную заявку, насколько возможно при моделировании. В действительности, операционный планировщик удаляет активную заявку из CEC, вызывает подпрограмму для следующего последовательного блока (NSB), и заменяет заявку на CEC *IN FRONT* ее рангами (то есть тот же самый приоритет) на CEC. Эта замена модифицирует PRIORITY и BUFFER блоки. После того, как заявка вводит BUFFER блок, она заменена *BEHIND* его рангов на CEC. Более детальное рассмотрение операционного планирования находится в главе 9.

BUFFER блоки используются, чтобы позволить недавно появившимся заявкам опережать активную заявку. Активная заявка вводится в блок, который осуществляет возможность, при которой должен перейти к завершению прежде, чем активная заявка должна будет продолжиться. Это необходимо, чтобы следовать за такими BUFFER блоками, чтобы позволить появившимся заявкам немедленно вернуться в моделировании.

Режим отказа

Заявкам не бывает отказано во вводе в BUFFER блок.

Связывающие окна

- **CEC Snapshot** - изображение части цепочки текущих событий в моделировании.
- **FEC Snapshot** - изображение части будущей цепочки событий в моделировании.

CLOSE

CLOSE блок заканчивает поток данных и восстанавливает код его ошибки.

CLOSE A, B, C

Операнды

А - Операционный параметр. Дополнительный. Операнд должен быть *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* or *SNA*Parameter*.

В - Поток данных. Дополнительный. Операнд должен быть *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*. Значение по умолчанию 1.

С - Альтернативный адрес. Дополнительный. Операнд должен быть *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

Пример

CLOSE Error_Parm,2

В этом примере CLOSE блок заканчивает операцию потока данных 2 и освобождает все ресурсы, связанные с ним. Внутренний код ошибки потока данных 2 помещен в параметр *Error_Parm* активной заявки.

Действие

CLOSE блок завершает поток данных и восстанавливает его код ошибки.

Если операнд **А** используется, он оценен в числовой форме, сокращен, и используется как значение параметра активной заявки. Код ошибки потока данных помещен в этот параметр.

Если операнд **В** используется, он оценен в числовой форме, сокращен, и используется как значение объекта потока данных. Результат должен быть положительным целым числом.

Если операнд **В** не используется, поток данных номер 1 закрыт.

Если операнд **С** используется, он оценен в числовой форме, сокращен, и используется как объекта *Entity* блока альтернативного адреса. Когда код ошибки потока данных является отличным от нуля, активная заявка переходит к дополнительному блоку альтернативного адреса после того, как он вводит CLOSE блок.

Глава 4 (4.16) содержит полное рассмотрение потоков данных, включая описания кода ошибки, в разделе Data Streams.

Разделение на блоки Условия

Моделирование блокировано, в то время как CLOSE восстанавливает код ошибки.

Специальные Ограничения

- если **А** и **В**, определены положительно.
- если **С** определено, должно быть в корректном блоке моделирования.

Режим отказа

Заявкам не бывает отказано во вводе в CLOSE блок.

Связанные Блоки

- **OPEN** - создают поток данных.
- **READ** - читают текстовую строку от потока данных.
- **WRITE** - записывают текстовую строку в поток данных.
- **SEEK** - набор текущих позиций строки потока данных.

COUNT

Блок COUNT размещает расчет объекта в параметр активной заявки.

COUNT O A, B, C, D, E

Операнды

О - Условный оператор или логический оператор. Условия его выбора объяснены ниже. Необходимый. Оператор должен быть *FNV*, *FV*, *I*, *LS*, *LR*, *NI*, *NU*, *SE*, *SF*, *SNE*, *SNF*, *SNV*, *SV*, *U*, *E*, *G*, *GE*, *L*, *LE*, *MIN*, *MAX*, or *NE*.

А - Число параметра для применения расчета. Необходимый. Операнд должен быть *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

В - Число или название объекта в нижнем конце диапазона. Номер первого объекта, который будет проверен. Тип объекта неявно определен логическим оператором или Операндом **Е**. Необходимый. Операнд должен быть *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

С - Число или название объекта в верхнем конце диапазона. Номер последнего объекта, который будет проверен. Необходимый. Операнд должен быть *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

D - Значение ссылки для Операнда E. Требуется только когда находится в Условном Режиме. Дополнительный. Операнд должен быть *Null*, *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

Настя
Андрей
Оля

SV – Память (Storage) должна быть доступна для продолжения вычислений.

U – Устройства (Facility) должны использоваться для продолжения вычислений.

Условные операторы

Как и логические, условные операторы являются обязательными. Они могут быть следующих видов: E, G, GE, L, LE, MAX, MIN, и NE. Условия определяются по следующим правилам:

- E – значение автоматического (машинного) SNA должно быть равно значению операнда D для продолжения вычислений.
- G – значение автоматического (машинного) SNA должно быть больше значения операнда D для продолжения вычислений.
- GE – значение автоматического (машинного) SNA должно быть больше либо равно значению операнда D для продолжения вычислений.
- L – значение автоматического (машинного) SNA должно быть меньше значения операнда D для продолжения вычислений.
- LE – значение автоматического (машинного) SNA должно быть меньше либо равно значению операнда D для продолжения вычислений.
- MAX – значение автоматического (машинного) SNA должно быть равно максимальному из всех найденных значений SNA для продолжения вычислений.
- MIN – значение автоматического (машинного) SNA должно быть равно минимальному из всех найденных значений SNA для продолжения вычислений.
- NE – значение автоматического (машинного) SNA должно быть неравно значению операнда E для продолжения вычислений.

Особые ограничения

- D и E являются обязательными, если O – условный оператор.
- В процессе вычислений, C должно быть больше либо равно B.

Режим отказа

Заявке никогда не будет отказано в доступе в блок COUNT.

Вспомогательные окна

- Facilities Window (Окно устройств) – Возможность просмотра изменения состояния устройств в реальном времени.
- Logicswitches Window (Окно логических переключений) – Просмотр действия логических схем в реальном времени.
- Storages Window (Окно памяти) – Просмотр изменения содержимого памяти в реальном времени.

DEPART (ОТПРАВКА)

Блок DEPART ведет статистику уменьшения содержимого очереди.

DEPART A,B

Операнды

A – Имя или номер очереди. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

В – Количество единиц, на которое необходимо уменьшить количество требований в очереди. Значение по умолчанию 1. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *String*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Пример

DEPART WaitingLine

В этом примере количество требований в очереди *WaitingLine* уменьшается на единицу и обновляется содержание статистики.

Алгоритм работы

Когда заявка поступает в блоке DEPART, операнд А вычисляется, округляется и используется для поиска очереди по номеру. Если необходимо, очередь создается. Если очередь создана, попытка уменьшения количества требований в очереди приведет к аварийной остановке. Для изменения содержимого очереди может быть использовано интерактивное ручное моделирование.

Операнд В определяет значение, используемое для уменьшения количества требований в очереди. Если значение В определено, операнд В вычисляется в числовом формате, округляется и используется в качестве результата. Если В не определено, используется значение по умолчанию 1.

В результате, обновляется статистика использования очереди.

Особые ограничения

- А и В должны быть положительными, если они определены.
- Если количество требований в очереди становится отрицательным, фиксируется аварийная остановка.

Режим отказа

Заявке никогда не будет отказано в доступе в блок DEPART.

Родственные SNA

- *QEntnum* – Текущее количество требований в очереди. Текущее значение количества требований в очереди *Entnum*.
- *QAEntnum* – Среднее количество требований в очереди. Среднее по времени количество требований в очереди *Entnum*.
- *QCEntnum* – Общее количество требований в очереди. Сумма всех поступивших в очередь требований *Entnum*.
- *QMEntnum* – Максимальное количество требований в очереди. Максимальное (высшая отметка) количество требований в очереди *Entnum*.
- *QTEntnum* – Среднее время ожидания в очереди. Среднее время пребывания в очереди *Entnum*.
- *QXEntnum* – Среднее время ожидания в очереди без нулевых значений. Среднее время пребывания в очереди *Entnum* без учета значений с нулевым временем пребывания.
- *QZEntnum* – Очередь нулевых входных значений. Количество поступивших в очередь заявок *Entnum* с нулевым временем пребывания.

Вспомогательное окно

- *Queues Window* (Окно очереди) – Просмотр изменения содержимого очереди в реальном времени.

DISPLACE (ПЕРЕМЕЩЕНИЕ)

Блок DISPLACE перемещает заявки.

DISPLACE A,B,C,D

Операнды

А – Номер заявки. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

В – Место назначения перемещаемой заявки. Название или номер блока. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

С – Номер параметра. Параметр перемещаемой заявки. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Д – Альтернативное место назначения для активной заявки. Название или номер блока. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Пример

DISPLACE X\$Culprit,Compensate,Residual,NotCaught

В этом примере, блок DISPLACE перемещает заявку, номер которой содержится в буферной переменной *Culprit*. Если требуемая заявка существует, она перемещается в следующий блок, отмеченный как *Compensate*. Если она участвует в FEC (цепи будущих событий), время до ее повторного поступления на моделирование вычисляется и хранится в параметре, который называется *Residual*. Если заявка была в FEC и такого параметра не существует, он создается. Если требуемой заявки не существует, то после поступления в блок DISPLACE активная заявка перемещается в альтернативное место назначения, отмеченное как *NotCaught*.

Алгоритм работы

Блок DISPLACE перемещает любую заявку в любой блок. Перемещенная заявка удаляется из Transaction chains (очереди заявок), как это было описано выше, и отправляется на вход блока назначения.

Когда заявка поступает в блок DISPLACE, операнд А вычисляется, округляется и используется для поиска заявки, которую необходимо переместить. Если такой заявки не существует, процесс прекращается и активная заявка отправляется в альтернативное место назначения, определенное операндом D, если такое существует.

Если заявка существует, она передается в новый блок, расположение которого определено операндом В.

Если перемещаемая заявка находится в FEC, она забирается оттуда и ее оставшееся время обработки вычисляется как время, которое необходимо заявке, чтобы покинуть FEC минус текущее время. Если используется операнд С, создается новый параметр.

Когда заявка перемещена, она передается в новый блок назначения и исключается из очередей:

- FEC (Future Events Chain) (цепи будущих событий)
- PENDING (INTERRUPT-MODE PREEMPT) CHAINS (очереди неоконченной обработки в приоритетном режиме прерывания)
- DELAY (MAJOR PRIORITY) CHAINS (очереди прерванной по старшинству приоритетов обработки)
- USER CHAINS (накопительной системы)
- RETRY CHAINS (повторной очереди)

И не удаляется из очередей:

- CEC (Current Events Chain) (цепи текущих событий)
- INTERRUPT (preempted) CHAINS (очереди приоритетных прерываний)
- GROUP CHAINS (общей очереди)

Если заявка перемещена, приоритетное прерывание обслуживания на устройствах не отменяется.

Режим отказа

Заявке никогда не будет отказано в доступе в блок DISPLACE.

Родственные SNA

- XN1 – Номер активной заявки.

Вспомогательные окна

- Blocks Window (окно блоков) – Просмотр смены блоков в реальном времени.
- Transaction Snapshot (текущее состояние заявки) – Изображение состояния заявки в моделировании.
- CEC Snapshot (текущее состояние CEC) – Изображение состояния цепи текущих событий в моделировании.
- FEC Snapshot (текущее состояние FEC) – Изображение состояния цепи будущих событий в моделировании.

Библиотека вспомогательных процедур

- QueryXNExist – устанавливает существование заявки.
- QueryXNParameter – возвращает значение параметра заявки.
- QueryXNAssemblySet – возвращает набор заявок.
- QueryXNPriority – возвращает значение приоритета заявки.
- QueryXNM1 – возвращает время маркировки (Mark Time) заявки.

ENTER (ПОСТУПЛЕНИЕ)

Когда заявка поступает на вход блока ENTER, она может либо сразу занимать, либо ожидать необходимое количество единиц памяти.

ENTER A,B

Операнды

A – Имя или номер памяти. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

B – Количество единиц, на которое следует уменьшать емкость памяти. Значение по умолчанию 1. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Пример

ENTER Toolkit,2

В этом примере текущей заявке требуются две единицы памяти из доступных единиц памяти, обозначенных как *Toolkit*. Если в памяти не хватает единиц, заявка отправляется на ожидание в Delay Chain (очередь задержки) к памяти.

Алгоритм работы

Когда заявка поступает на вход блока ENTER, операнд A вычисляется и используется для поиска памяти по номеру. Если память не существует, фиксируется аварийная остановка. Содержимое памяти должно заполняться при помощи команды STORAGE.

Если память существует, операнд B используется для определения запроса к памяти. Если значение B определено, операнд B вычисляется, округляется и используется в качестве результата. Если B не определено, используется значение по умолчанию 1.

Если память доступна, и если в ней достаточно единиц хранения для удовлетворения запроса, заявка получает разрешение на вход в блок ENTER и запрос удовлетворяется посредством уменьшения текущего количества единиц памяти на определенное число. В противном случае, заявка отправляется в Delay Chain (очередь задержки) к памяти в порядке приоритета.

Когда ячейки (единицы) памяти освобождаются блоком LEAVE, первая, претендующая на обработку, в соответствии с выбранной дисциплиной, заявка получает разрешение на вход в блок ENTER. Каждое такое поступление разрешается еще до того, как текущая заявка покидает блок LEAVE. Это предотвращает «линейное подавление» (line-bucking).

Когда заявка поступает на вход блока ENTER или блока LEAVE, обновляется статистика содержимого памяти.

Особые ограничения

- A должно быть положительным.
- B должно быть неотрицательным.

- Если А запрашивает больший объем памяти, чем имеется в наличии, это приведет к аварийной остановке (например, операнд В, превышающий определенный размер памяти, заданный операндом А)

Режим отказа

- Активной заявке будет отказано в доступе к блоку ENTER, если она превышает свободный объем памяти.
- Активной заявке будет отказано в доступе к блоку ENTER, если отсутствует доступ к памяти.

Если заявке отказано в поступлении в блок ENTER, срабатывает ее индикатор задержки (Delay Indicator), который накапливается до тех пор, пока заявка не оказывается в режиме Simultaneous (синхронном) блока TRANSFER. Синхронный режим блока передачи используется редко. Логические переменные способны более эффективно контролировать координацию состояния определенного числа ресурсов при использовании блока TEST.

Вспомогательные команды

Память должна быть определена командой STORAGE перед тем, как она сможет обновляться в блоке ENTER. Команда STORAGE должна существовать в модели или же быть приписана объекту моделирования интерактивно, перед тем как заявка поступит в блок. Любые попытки поступления заявки в блок ENTER до того, как память будет определена, вызовут аварийную остановку.

Память может быть переопределена при помощи интерактивной команды STORAGE.

Родственные SNAs

- *REntnum* – Неиспользуемая емкость памяти. Объем памяти (или "маркер" пустоты) доступный для использования поступающей заявкой или содержимым самой памяти *Entnum*.
- *SEntnum* – Занятая память. *SEntnum* возвращает объем содержащейся в памяти информации (или "маркер" пустоты) использующийся в данный момент активной заявкой или содержимым самой памяти *Entnum*.
- *SAEntnum* – Средняя емкость используемой памяти. *SAEntnum* возвращает среднюю по времени емкость используемой памяти (или "маркер" пустоты) *Entnum*.
- *SCEntnum* – Объем используемой памяти. Общее количество единиц (ячеек) памяти, которые были заняты (или "маркер" пустоты) *Entnum*.
- *SEEntnum* – Память пуста. *SEEntnum* возвращает 1, если память *Entnum* не используется, 0 в противном случае.
- *SFEntnum* – Память заполнена. *SFEntnum* возвращает 1, если память *Entnum* заполнена, 0 в противном случае.
- *SREntnum* – Использование памяти. Часть общего использования, представленная в виде средней занятости памяти *Entnum*. *SREntnu*, выражается в долях тысячи и, следовательно, возвращает вещественное значение от 0 до 1000, включительно.
- *SMEntnum* – Максимальный объем используемой памяти *Entnum*. "Высшая планка".
- *STEntnum* – Среднее время хранения единицы памяти *Entnum*.
- *SVEntnum* – Состояние памяти. *SVEntnum* возвращает 1, если память *Entnum* доступна, 0 в противном случае.

Вспомогательные окна

- Storages Window (Окно памяти) – Просмотр изменения содержимого памяти в реальном времени.

EXAMINE (ПРОВЕРКА)

Блок EXAMINE может использоваться для тестирования на принадлежность к числовой группе или к группе заявок.

EXAMINE A,B,C

Операнды

A – Номер группы, члены которой подлежат проверке. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

B – Числовой режим. Значение, которое будет проверяться на принадлежность группе. Обязательный. В качестве операндов должны быть указаны *Null*, *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

C – Номер блока. Альтернативное место назначения для активной заявки, если в группе не окажется ни одного члена. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Примеры

EXAMINE ValidColors,P\$Color,NotCorrectColor

В этом простом примере, если числовая группа *ValidColors* не содержит значения, содержащегося в параметре заявки *Color*, то активная заявка направляется в раздел блока *NotCorrectColor*. Если значение цвета попадает с числовую группу, активная заявка отправляется в следующий блок (Next Sequential Block (NSB)).

EXAMINE ValidXNs,,NotValidXN

В этом примере, если активная заявка не является членом группы заявок *ValidXNs*, она направляется в раздел блока *NotValidXN*. Если же заявка является членом этой группы, она отправляется в следующий блок (Next Sequential Block (NSB)).

Алгоритм работы

Блок EXAMINE работает в двух режимах: числовом ("Numeric Mode") и режиме заявок ("Transaction Mode"). Если используется операнд B, блок EXAMINE работает в числовом режиме. В этом случае значение операнда B проверяется на принадлежность к числовой группе. Если операнд B не используется, блок работает в режиме заявок. В этом случае, активная заявка проверяется на принадлежность к определенной группе заявок.

Когда заявка поступает на вход блока ENTER, операнд A вычисляется и подбирается соответствующая ему группа. Если такой группы не существует, она создается.

Если используется операнд B, он вычисляется численно и результат проверяется на соответствие определенной числовой группе. Если операнд B не используется, активная заявка проверяется на принадлежность к группе заявок, определенной операндом A.

При использовании дробных чисел в числовой группе наблюдается некоторая потеря эффективности работы.

Если проверка на принадлежность к группе не прошла, операнд C вычисляется, округляется и используется в качестве номера блока назначения для активной заявки. Если принадлежность установлена, активная заявка направляется в следующий блок (Next Sequential Block (NSB)).

Особые ограничения

- A и C должны быть положительными.
- C должно содержать номер существующего блока.

Режим отказа

Заявке никогда не будет отказано в доступе в блок EXAMINE.

Родственные блоки

Заявки и числа распределяются по группам при помощи блока JOIN. К заявкам в группе заявок могут обращаться блоки ALTER, EXAMINE, REMOVE, и SCAN. К числам из числовых групп могут обращаться блоки EXAMINE и REMOVE.

Родственные SNAs

- *GNEntnum* – Подсчет количества членов числовой группы. *GNEntnum* возвращает количество членов числовой группы *Entnum*.
- *GTEntnum* – Подсчет количества членов группы заявок. *GTEntnum* возвращает количество членов группы заявок *Entnum*.

Вспомогательные окна

- Numeric Groups Snapshot (Текущее состояние числовой группы) – Изображение состояния числовой группы в моделировании.
- Transaction Groups Snapshot (Текущее состояние группы заявок) – Изображение состояния группы заявок в моделировании.

EXAMINE (ОБРАБОТКА)

Блок EXAMINE может активировать работу любого блока моделирования для активной заявки.

EXAMINE A

Операнды

A – Номер блока. Имя или номер блока, вызываемого на выполнение. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Пример

EXAMINE P\$ActiveBlock

Этот блок EXAMINE будет вызывать блок, номер которого указан в параметре *ActiveBlock*, для работы с активной заявкой.

Алгоритм работы

Когда заявка поступает на вход блока EXAMINE, операнд **A** вычисляется и используется для нахождения блока по указанному номеру или имени. Затем активируется работа указанного блока относительно активной заявки.

Если активной заявке отказано в доступе в соответствующий блок, она остается в блоке EXAMINE .

После того как работа нужного блока активирована, активная заявка направляется в блок, следующий за блоком EXAMINE. Однако, если в этом блоке указано другое место назначения для активной заявки, не являющееся блоком EXAMINE , блок самостоятельно перенаправляет заявку в следующий блок без помощи блока EXAMINE .

Особые ограничения

- **A** должно содержать номер существующего блока.
- Блок EXAMINE не может активировать другой блок EXAMINE

Режим отказа

Заявке никогда не будет отказано в доступе в блок EXAMINE .

Вспомогательные окна

- Blocks Window (Окно блоков) – Возможность просмотра смены блоков в реальном времени.

FAVAIL (ПРОВЕРКА ДОСТУПНОСТИ УСТРОЙСТВА)

Блок FAVAIL определяет, находится ли устройство в доступном состоянии.

FAVAIL A

Операнды

A – Номер устройства. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Пример

FAVAIL Teller1

При поступлении активной заявки в этот блок, осуществляется проверка на предмет занятости устройства *Teller1*.

Алгоритм работы

Блок FAVAIL определяет, что устройство свободно. Если устройство свободно, блок FAVAIL дает заявке право его занять. Это обсуждается более подробно в главе 9.

Если же устройство занято, блок FAVAIL не делает ничего.

Блок FAVAIL отменяет действие блока FUNAVAIL на устройства, но не влияет на перемещение заявки.

Режим отказа

Заявке никогда не будет отказано в доступе в блок FAVAIL.

Родственный блок

- FUNAVAIL – Устанавливает устройство в состояние «недоступно».

Родственные SNAs

Связанные с устройствами SNAs:

- FEntnum – Занятость устройства. Если устройство Entnum в данный момент занято, FEntnum возвращает 1. В противном случае, FEntnum возвращает 0.
- FCEntnum – Количество итераций работы устройства. Сколько раз устройство Entnum было занято заявкой.
- FIEntnum – Прерывания работы устройства Entnum. Если работа устройства Entnum в данный момент была прервана, FIEntnum возвращает 1. В противном случае, FIEntnum возвращает 0.
- FREntnum – Использование устройства. Время, в течение которого устройство Entnum было занято. FREntnum выражается в долях тысячи и, следовательно, возвращает вещественное значение от 0 до 1000, включительно.
- FTEntnum – Среднее время занятости (удерживания) устройства. Среднее время, в течение которого устройство Entnum было занято заявкой.
- FVEntnum – Доступность устройства. FVEntnum возвращает 1, если устройство Entnum доступно, 0 в противном случае.

Вспомогательные окна

- Facilities Window (Окно устройств) – Возможность просмотра изменения состояния устройств в реальном времени.

FUNAVAIL (УСТАНОВЛЕНИЕ УСТРОЙСТВА В НЕДОСТУПНОЕ СОСТОЯНИЕ)

Блок FUNAVAIL используется для установления устройства в недоступное для заявки состояние.

FUNAVAIL A,B,C,D,E,F,G,H

Операнды

A – Номер или имя устройства. Обязательный. В качестве операндов должны быть указаны Name, PosInteger, ParenthesizedExpression, SNA или SNA*Parameter.

B – Режим REmove (прерывания) или COntinue (продолжения) обработки для обрабатываемой заявки. Необязательный. В качестве операндов должны быть указаны RE, CO, или Null.

C – Номер блока. Следующий блок, в который должна быть передана обрабатываемая устройством Заявка. Необязательный. В качестве операндов должны быть указаны Null, Name, PosInteger, ParenthesizedExpression, SNA, или SNA*Parameter.

D – Номер параметра. Параметр для получения оставшегося времени обработки, если обрабатываемая заявка была удалена из FEC. Необязательный. В качестве операндов должны быть указаны Null, Name, PosInteger, ParenthesizedExpression, SNA или SNA*Parameter.

E – Режим REmove (прерывания) или COntinue (продолжения) обработки для прерванной заявки. Необязательный. В качестве операндов должны быть указаны RE, CO или Null.

F – Номер Блока. Новый блок для заявки, обработка которой на устройстве была прервана. Необязательный. В качестве операндов должны быть указаны Null, Name, PosInteger, ParenthesizedExpression, SNA или SNA*Parameter.

G – REmove (прерывания) или COninue (продолжения) обработки для Заявки, обработка которой на устройстве не была завершена или была приостановлена. Необязательный. В качестве операндов должны быть указаны RE, CO или *Null*.

H – Номер блока. Новый блок для заявки, обработка которой на устройстве не была завершена или была приостановлена. Необязательный. В качестве операндов должны быть указаны *Null, Name, PosInteger, ParenthesizedExpression, SNA* или *SNA*Parameter*.

Примеры

FUNAVAIL Teller1

Это наиболее простой способ использования блока. Устройство *Teller1* замораживается на определенный период, в течение которого оно будет недоступно. Это оказывает влияние на некоторые типы заявок, отличных от активной заявки. Если обрабатываемая заявка находится в блоке ADVANCE, она будет автоматически перенаправлена в FEC, когда устройство снова станет доступно. Время пребывания в блоке ADVANCE (т.е. «время остаточной обработки») высчитывается и сохраняется автоматически, когда заявка забирается из FEC. Время остаточной обработки становится инкрементом времени при перенаправлении заявки в FEC. Заявка, обработка которой на устройстве *Teller1* не была завершена или была приостановлена, не будет никуда перемещена до тех пор, пока устройство снова не будет доступно.

FUNAVAIL TELLER,RE,TLR2,300,RE,MGR,CO

В этом примере устройство *TELLER* устанавливается в состояние «недоступно». Это означает, что прибытие следующей заявки будет задержано. С заявками, обрабатываемыми на устройстве *TELLER* или обработка которых была прервана на этом устройстве, производятся действия в соответствии с операндами B–H. Если заявка, обрабатывавшаяся на устройстве в данный момент (момент его установления в недоступное состояние), присутствует в FEC, она выводится из очереди и время остаточной обработки сохраняется в ее параметре номер 300. Эта заявка перемещается в блок *TLR2*. Заявки, текущая обработка которых на устройстве *TELLER* была прервана, изымаются из группы претендующих на устройство заявок и перенаправляются в блок *MGR*. До тех пор пока операнд G не примет значение CO, заявка, удерживаемая в данный момент на устройстве *TELLER* остается на устройстве, несмотря на то, что последнее находится в недоступном состоянии.

Алгоритм работы

Сложность блока FUNAVAIL обуславливается тремя классами заявок, которые может обрабатывать устройство:

1. обрабатываемые заявки (owning Transaction) (операнды B–D),
2. прерванные заявки (preempted Transactions) в Interrupt Chain (очереди прерванной обработки) (операнды E–F) и
3. ожидающие заявки (delayed Transactions) в Pending Chain (очереди неоконченной обработки) или Delay Chain (очереди приостановленной обработки) (операнды G–H).

Блок FUNAVAIL позволяет исключать устройство из процесса и контролировать положение ожидающих, обрабатываемых и прерванных заявок. Заявки, прибывающие на устройство в то время, пока оно находится в недоступном состоянии, будут задержаны и им не будет разрешено занять устройство. Блок FUNAVAIL не может оказывать влияние на устройства, уже находящиеся в недоступном состоянии.

При использовании параметра REmove (прерывание), заявка изымается из группы претендующих на устройство заявок. Если параметр REmove используется для заявок, обработка которых на устройстве не была завершена или была приостановлена, то есть, если параметр G находится в состоянии RE, то параметр H должен использоваться для перенаправления заявок.

При использовании параметра COninue (продолжение), заявки, стоящие в очереди на определенное устройство, могут продолжить поступать на устройство, даже в течение

периода, когда устройство недоступно. В этом случае, статистика использования устройства учитывает и это время тоже.

При использовании блока альтернативного назначения, заявки изымаются из их текущего местоположения и перенаправляются в новый блок. Заявки, обработка которых не была завершена или была приостановлена, управляемые операндами G и H, не могут быть перенаправлены без использования параметра REmove. Обрабатываемая заявка, управляемая операндами B и D, и прерванные заявки, управляемые операндами E и F, могут продолжить претендовать на устройство или даже быть перемещены к новому месту назначения. Это делается при помощи определения альтернативного места назначения без использования соответствующего параметра RE.

Если в операнде B параметр RE не указывается, любая обрабатываемая на устройстве заявка прерывается. Такие заявки не могут покинуть Блоки ASSEMBLE, GATHER и MATCH до тех пор, пока приоритетное прерывание обслуживания не будет разблокировано.

Когда заявка перемещается при помощи операндов C, F или H, она передается в новый блок назначения и изымается из очередей:

- FEC (Future Events Chain) (цепи будущих событий)
- PENDING (INTERRUPT-MODE PREEMPT) CHAINS (очереди неоконченной обработки в приоритетном режиме прерывания)
- DELAY (MAJOR PRIORITY) CHAINS (очереди прерванной по старшинству приоритетов обработки)
- USER CHAINS (накопительной системы)
- RETRY CHAINS (повторной очереди)

И не удаляется из очередей:

- CEC (Current Events Chain) (цепи текущих событий)
- INTERRUPT (preempted) CHAINS (очереди приоритетных прерываний)
- GROUP CHAINS (общей очереди)

Когда заявка перемещается из текущего местопребывания в любое другое при помощи использования альтернативного места пребывания, любые наложенные на нее блоком условия прекращают свое действие, но приоритеты сохраняются. Перемещаемая заявка перенаправляется на вход определенного в операнде C блока. Когда обрабатываемая заявка RELEASE (ОСВОБОЖДАЕТ) или RETURN (ВОЗВРАЩАЕТСЯ) на устройство, следующая обрабатываемая заявка выбирается из начала Pending Chain (очереди незаконченной обработки), Interrupt Chain (очереди приоритетных прерываний) или Delay Chain (очереди приостановленной обработки) – в таком порядке.

Обрабатываемая заявка (The Owning Transaction)

Операнды B и D используются для управления обрабатываемой заявкой.

Если на B установлено значение CO, обрабатываемая заявка не теряет прав на обработку (остается на устройстве). Также она может быть передана к новому месту назначения операндом C.

Если на B установлено значение RE, обрабатываемая заявка удаляется из группы претендующих на устройство заявок. Это означает, что эта заявка может продолжить пребывание в системе моделирования без ограничений на приоритетное прерывание обслуживания на этом устройстве (тем не менее, за ней оставляется право на приоритетное прерывания обслуживания на других устройствах). Это также означает, что обрабатываемая заявка не должна пытаться RELEASE (ОСВОБОДИТЬ) или RETURN (ВЕРНУТЬСЯ) на устройство. Операнд C должен использоваться для переадресации данной заявки.

Если на B установлен *Null*, обработка текущей заявки прерывается и она помещается в Interrupt Chain (очереди прерываний) устройства. Если она было взята из FEC, и операнд C не был использован, она автоматически восстановится в FEC, с использованием

автоматического сохранения времени остаточной обработки, когда устройство вновь станет доступно.

Операнд С может использоваться независимо от операнда В. Он обуславливает перемещение заявки и определяет ее новый блок назначения. Если необходимо вернуть заявку в FEC, при использовании операнда С, следует явно перенаправить ее в блок ADVANCE (ПРОДВЖЕНИЯ). Операнд D обуславливает сохранение времени остаточной обработки в параметре обрабатываемой заявки. Значение времени остаточной обработки затем становится доступно для явного перенаправления, при использовании значения параметра в качестве операнда А в блоке ADVANCE.

Прерванная заявка (Preempted Transactions)

Операнды Е и F обеспечивают контроль за судьбой заявок, обработка которых на устройстве в данный момент была прервана. Если на Е установлено значение СО, прерванные заявки не удаляются из группы претендующих на устройство заявок, и могут занять устройство в течение всего периода его недоступности. Прерванные заявки могут быть переданы к новому месту назначения при помощи операнда F.

Если на Е установлено значение RE, прерванные заявки удаляются из группы претендующих на устройство заявок. Это означает, что заявка может продолжить пребывание в системе моделирования без ограничений на приоритетное прерывание обслуживания на этом устройстве (тем не менее, за ней оставляется право на приоритетное прерывания обслуживания на других устройствах). Это также означает, что обрабатываемая заявка не должна пытаться RETURN (ВЕРНУТЬСЯ) или RELEASE (ОСВОБОДИТЬ) устройство.

Необязательный операнд F может выполнить переадресацию такой заявки.

Если на Е установлен *Null*, прерванные заявки оставляются в Interrupt Chain (очереди прерываний) устройства в течение всего периода его недоступности.

Операнд F может использоваться независимо от операнда Е. Он обуславливает перемещение прерванной заявки и определение ее нового блока назначения. Прерванные заявки могут не существовать в FEC, поэтому время остаточной обработки для них несущественно. Если на Е установлен *Null*, право на приоритетное прерывание обслуживания для перемещаемых заявок сохраняется.

Режим отложенного прерывания заявок (Pending Interrupt Mode Transactions)

Операнды G и H обеспечивают контроль за судьбой заявок в Pending Chain (очереди неоконченной обработки) в режим отложенного прерывания PREEMPT или Delay Chain (очереди приостановленной обработки).

Если на G установлен параметр СО, задержанные заявки не удаляются из группы претендующих на устройство заявок, и могут занять устройство в течение всего периода его недоступности.

Если на G установлено значение RE, задержанные заявки удаляются из группы претендующих на устройство заявок и могут продолжить пребывание в системе моделирования. Эти заявки не должны пытаться RETURN (ВЕРНУТЬСЯ) или RELEASE (ОСВОБОДИТЬ) устройство. Операнд H должен использоваться для переадресации такой заявки.

Если на G установлен *Null*, задержанные заявки оставляются в Pending Chain (очередь неоконченной обработки) или Delay Chain (очередь приостановленной обработки) и не могут претендовать на устройство в течение всего периода его недоступности. Использование операнда H несостоятельно при установлении G в *Null*.

Особые ограничения

- Если В установлен в RE, использование С обязательно.
- При использовании H, G должно быть установлено в RE.
- Если G установлен в RE, использование H обязательно.

Режим отказа

Заявке никогда не будет отказано в доступе в блок FUNAVAIL.

Родственный блок

- FAVAIL – Устанавливает устройство в доступное состояние.

Родственные SNAs

- FEntnum – Занятость устройства. Если устройство *Entnum* в данный момент занято, FEntnum возвращает 1. В противном случае, FEntnum возвращает 0.
- FCEntnum – Количество итераций работы устройства. Сколько раз устройство *Entnum* было занято заявкой.
- FIEntnum – Прерывания работы устройства *Entnum*. Если работа устройства *Entnum* в данный момент была прервана, FIEntnum возвращает 1. В противном случае, FIEntnum возвращает 0.
- FREntnum – Использование устройства. Время, в течение которого устройство *Entnum* было занято. FREntnum выражается в долях тысячи и, следовательно, возвращает вещественное значение от 0 до 1000, включительно.
- FTEntnum – Среднее время занятости (удерживания) устройства. Среднее время, в течение которого устройство *Entnum* было занято заявкой.
- FVEntnum – Доступность устройства. FVEntnum возвращает 1, если устройство *Entnum* доступно, 0 в противном случае.

Вспомогательные окна

- Facilities Window (Окно устройств) – Возможность просмотра изменения состояния устройств в реальном времени
- CEC Snapshot (Текущее состояние CEC) – Изображение состояния цепи текущих событий в моделировании.
- FEC Snapshot (Текущее состояние FEC) – Изображение состояния цепи будущих событий в моделировании.

GATE (ШЛЮЗ)

Блок GATE просеивает поток заявок, на основе состояния определенного модуля.

GATE O A,B

Операнды

O – Условный оператор. Указание условия, по которому будет тестироваться модуль. Обязательный. Варианты значений оператора: FNV, FV, I, LS, LR, M, NI, NM, NU, SE, SF, SNE, SNF, SNV, SV, или U.

A – Имя или номер тестируемого модуля. Тип модуля определяется логическим оператором. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

B – Номер блока назначения в случае неудачного прохождения теста. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, *SNA*Parameter*.

Примеры

GATE SNF MotorPool

В этом примере «режима отказа» блока GATE, активная заявка поступает в блок, если блок памяти *MotorPool* не заполнена (т.е., если хотя бы 1 ячейка памяти свободна). Если же память заполнена, активная заявка блокируется до тех пор, пока 1 или несколько ячеек памяти не освободятся.

GATE SNE MotorPool,CupboardIsBare

В этом примере «альтернативного режима выхода» блока GATE, активная заявка всегда поступает в блок GATE. Если блок памяти *MotorPool* не пуст (т.е., хотя бы 1 ячейка памяти используется), Заявка направляется в следующий блок (NSB). Если память пуста (тест не прошел), активная заявка направляется на вход блока с названием *CupboardIsBare*.

Алгоритм работы

Блок GATE функционирует в двух режимах: «отказа» и «альтернативного выхода».

Если операнд В не используется, блок GATE функционирует в режиме отказа. Когда заявка пытается поступить в работающий в режиме отказа блок GATE, и тест проходит неудачно, заявка блокируется до тех пор, пока тест не будет пройден вновь с удачным результатом. Если тест был пройден удачно, активная заявка поступает в блок GATE, а затем направляется в следующий блок.

Заблокированные заявки помещаются в Retry Chain (повторную очередь) тестирующего модуля. Когда состояние какого-либо модуля изменяется, заблокированная заявка возобновляется, проверка, определенная блоком GATE, повторяется, и, если она проходит успешно, заявка получает разрешение поступить в блок GATE. Однако, интеграция пользовательских переменных не вызывает реактивацию заблокированной заявки. Чтобы зафиксировать уровень одной ли нескольких непрерывных переменных, необходимо использовать команду INTEGRATE. Это рассмотрено в главе 1, в разделе «Непрерывные переменные».

Если операнд В используется, блок GATE функционирует в режиме альтернативного выхода. Когда заявка пытается поступить в работающий в таком режиме блок GATE, и тест проходит неудачно, заявка перенаправляется в блок альтернативного назначения, определенный операндом В, и помещается в СЕС (цепь текущих событий) в начало группы заявок с таким же приоритетом. Если тест был пройден удачно, активная заявка поступает в блок GATE, а затем направляется в следующий блок.

Условные операторы

Условные операторы являются обязательной частью. Они могут быть следующих видов: FNV, FV, I, LS, LR, M, NI, NM, NU, SE, SF, SNE, SNF, SNV, SV или U. Когда условие выполняется, Заявка проходит блок GATE и перенаправляется к следующему блоку. Условия описываются следующими правилами:

- FNV – Устройство, однозначно определенное операндом А, должно быть недоступно для успешного прохождения теста.
- FV – Устройство, однозначно определенное операндом А, должно быть доступно для успешного прохождения теста.
- I – Работа устройства, однозначно определенного операндом А, должна быть в данный момент прервана для успешного прохождения теста.
- LS – Логический переключатель, однозначно определенный операндом А, должен быть в состоянии «включено» для успешного прохождения теста
- LR – Логический переключатель, однозначно определенный операндом А, должен быть в состоянии «выключено» для успешного прохождения теста
- M – блок Match, однозначно определенный операндом А, должен содержать родственную (активной заявке) заявку, ожидающую условия соответствия (совпадения) для успешного прохождения теста
- NI – Работа устройства, однозначно определенного операндом А, не должна быть прервана для успешного прохождения теста.
- NM – блок Match, однозначно определенный операндом А, не должен содержать родственную (активной заявке) заявку, ожидающую условия соответствия (совпадения) для успешного прохождения теста.
- NU – Устройство, однозначно определенное операндом А, не должно использоваться для успешного прохождения теста.
- SE – Память, однозначно определенная операндом А, должна быть пуста для успешного прохождения теста. Ни одна ячейка памяти не должна быть занята.
- SF – Память, однозначно определенная операндом А, должна быть заполнена для успешного прохождения теста. Все ячейки памяти должны быть заняты.
- SNE – Память, однозначно определенная операндом А, должна быть не пуста для успешного прохождения теста. Хотя бы одна ячейка памяти должна быть занята.

- SNF – Память, однозначно определенная операндом A, должна быть не заполнена для успешного прохождения теста. Хотя бы одна ячейка памяти должна быть свободна.
- SNV – Память, однозначно определенная операндом A, должна быть в состоянии «недоступна» для успешного прохождения теста.
- SV – Память, однозначно определенная операндом A, должна быть в состоянии «доступна» для успешного прохождения теста.
- U – Устройство, однозначно определенное операндом A, должно использоваться для успешного прохождения теста.

Особые ограничения

- В, если оно определено, должно определять местоположение существующего Блока.
- Блоки GATE обладают высокой мощностью, но они могут занимать очень много компьютерного времени на обработку неудачных тестов. Возможно, понадобится даже модифицировать систему моделирования для уменьшения количества повторений неудачных тестов. Это можно организовать посредством помещения заявок, у которых нет шанса успешно пройти тест, в User Chain (накопительную систему), используя блоки LINK и UNLINK.
- Класс SNA MB не должен использоваться в одиночку для определения условий блокировки в блоке GATE. Вместо этого должны использоваться блоки MATCH.

Режим отказа

Блок GATE, работающий в режиме отказа, откажется пропускать заявку. Если результат теста будет отрицательным. Заявка, которой было отказано в доступе, будет заблокирована до тех пор, пока тест не будет пройден успешно.

Когда заявке отказывается в доступе, ее индикатор задержки включается и накапливается до тех пор, пока заявке не будет позволено пройти «синхронный» режим блока TRANSFER. Однако, с появлением булевых переменных в языке GPSS, более рационально использовать блок TEST и булевы переменные при необходимости скоординировать состояние составных модулей.

Вспомогательные окна

- Blocks Window (Окно блоков) – Просмотр смены блоков в реальном времени.
- Facilities Window (Окно устройств) – Возможность просмотра изменения состояния устройств в реальном времени.
- Logicswitches Window (Окно логических переключений) – Просмотр действия логических схем в реальном времени.
- Storages Window (Окно памяти) – Просмотр изменения содержимого памяти в реальном времени.

GATHER (НАКОПЛЕНИЕ)

Блок GATHER ожидает родственные заявки.

GATHER A

Операнд

A – Счетчик заявок. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Пример

GATHER 2

Это простейший способ использования блока GATHER. Первая заявка из набора (Assembly Set) (см. раздел 9.3) поступает в блок GATHER остается ждать. Когда другая заявка из того же набора поступает в блок, обе родственные заявки освобождаются и помещаются в CEC (цепь текущих событий).

Алгоритм работы

Когда заявка поступает в блок GATHER, в Match Chain (цепи совпадений) блока происходит поиск ожидающей заявки из того же набора. Если в блоке еще нет ни одного члена из этого набора, операнд A вычисляется, округляется, уменьшается на единицу и сохраняется в области памяти активной заявки. Если это число меньше либо равно нулю, заявка немедленно пытается поступить на следующий блок. В противном случае, заявка помещается в специальную очередь блока GATHER, называемую Match Chain (цепь совпадений), для ожидания прибытия в блок другого члена из своего набора.

Если активная заявка прибывает для поиска других членов из своего набора, уже находящихся в цепи совпадений, активная заявка также помещается в цепь, и счетчик заявок, хранящийся в первой помещенной в цепь заявке, уменьшается на единицу. Когда этот счетчик становится равен нулю, все родственные заявки удаляются из цепи совпадений. Все заявки, обработка которых не была прервана на каком-либо устройстве, посещаются в CEC за равными им по приоритету заявками.

Прерванным заявкам, которые были собраны в блоке GATHER, не разрешается покинуть блок до тех пор, пока все приоритетные прерывания обслуживания не будут разблокированы. Более подробное описание механизма приоритетного прерывания обслуживания можно найти в разделе 9.4. Прерванные заявки, удаленные из цепи совпадений, не принимают участия в дальнейшем разделении, даже если они остаются в блоке GATHER.

Блок GATHER отличается от блока ASSEMBLE тем, что все заявки, пришедшие после первой, в блоке ASSEMBLE ликвидируются.

Особые ограничения

- A должно быть положительным.
- Заявкам, обработка которых в текущий момент была прервана, не разрешается покидать блок GATHER.

Режим отказа

Заявке никогда не будет отказано в доступе в блок GATHER.

Родственные блоки

- ADOPT – Устанавливает наборы (Assembly Set) активных заявок
- ASSEMBLE – Ожидает и ликвидирует членов наборов.
- MATCH – Принуждает ожидать членов набора.
- SPLIT – Создает заявки из того же набора.

Родственные SNAs

- A1 – Набор. Возвращает набор активной заявки.
- MBEntnum – Совпадения в блоке. MBEntnum возвращает 1 если в блоке Entnum есть заявка из того же набора, что и активная заявка. В противном случае, MBEntnum возвращает 0.

Вспомогательные окна

- Blocks Window (Окно блоков) – Просмотр смены блоков в реальном времени.
- Transaction Snapshot (Текущее состояние заявки) – Изображение состояния заявки в моделировании.
- CEC Snapshot (Текущее состояние CEC) – Изображение состояния цепи текущих событий в моделировании.
- FEC Snapshot (Текущее состояние FEC) – Изображение состояния цепи будущих событий в моделировании.

GENERATE (ГЕНЕРАЦИЯ)

Блок GENERATE создает Заявки для поступления в систему моделирования.

GENERATE A,B,C,D,E

Операнды

A – Среднее значение интервала времени генерации. Необязательный. В качестве операндов должны быть указаны *Null, Name, Number, String, ParenthesizedExpression* или *DirectSNA*. Параметры заявки можно не использовать.

B – Интервал времени генерации или функция модификации. Необязательный. В качестве операндов должны быть указаны *Null, Name, Number, String, ParenthesizedExpression* или *DirectSNA*. Параметры Заявки можно не использовать.

C – Начальное время задержки. Время инкриминируется для первой заявки. Необязательный. В качестве операндов должны быть указаны *Null, Name, Number, String, ParenthesizedExpression*, или *DirectSNA*. Параметры заявки можно не использовать.

D – Создание ограничения. Необязательный. По умолчанию ограничений нет. В качестве операндов должны быть указаны *Null, Name, PosInteger, String, ParenthesizedExpression*, или *DirectSNA*. Параметры заявки можно не использовать.

E – Уровень приоритета. Необязательный. Значение по умолчанию 0. В качестве операндов должны быть указаны *Null, Name, integer, String, ParenthesizedExpression* или *DirectSNA*. Параметры заявки можно не использовать.

Пример

GENERATE 0.1

Это простейший способ использования Блока GENERATE. Этот блок подает заявку с нулевым приоритетом на вход системы моделирования каждую десятую долю единицы времени.

Алгоритм работы

Когда моделирование начинается, или подается интерактивная команда, какой-либо из незагруженных блоков GENERATE вызывается для формирования первой заявки. Такие заявки направляются на вход блока GENERATE и помещаются в FEC, если они содержат положительное временное приращение. Операнд C может быть использован для задания положительного времени приращения для первой заявки. В противном случае, первое временное приращение высчитывается из операндов A и B.

Использование параметров для операндов блока GENERATE запрещено. Только что сгенерированные заявки не имеют параметров, и их поступление в такой блок GENERATE приведет к аварийной остановке.

Перед тем, как будет создана новая заявка, операнд D высчитывается численно для определения, все ли необходимые заявки были созданы. Если предел создания не превышен, процесс продолжится. Затем блок GENERATE создает новую заявку, приписывая ей следующий номер заявки с приоритетом, указанным в операнде E, и время ожидания получает значение из абсолютных системных часов. Новая заявка представляется в виде нового набора заявок из одного члена.

Интервал прибытия для новой заявки высчитывается из значений операндов A, B и C. Если определен только операнд A, он высчитывается численно и используется в качестве временного приращения. Если имеются в наличии операнды A и B, и B не задан функцией, оба они высчитываются численно и случайное значение из интервала A–B и A+B, включительно, используется в качестве временного приращения. Вы можете задать начальные значения для генератора случайных чисел. Это задается в пункте «Random» настроек модели (Model Settings Notebook).

Выберите **Edit / Settings (Правка / Настройки)**

Затем выберите пункт **Random**. Заполните требуемое случайное значение в поле с пометкой «GENERATE». Значение по умолчанию 1.

Когда операнд B принадлежит к классу SNA FN, он называется «функцией модификации». В этом случае, временной инкремент высчитывается как произведение результата работы функции на вычисленный операнд A.

Если операнд C определен, он высчитывается численно и используется в качестве временного инкремента для первой заявки. Если вы хотите, чтобы первая заявка прибыла

в нулевое время, необходимо использовать отдельный блок GENERATE с ограничением создания, заданном в операнде D и нулевыми значениями операндов A и B. Если операнд A и/или B используются, и C=0, C будет интерпретировано как ноль. В приведенном ниже примере в нулевое время создаются три заявки.

GENERATE,,0,3

Если временной инкремент строго положителен (не нулевой), заявка помещается в FEC, если он равен нулю, заявка становится в CEC за равными ей по приоритету заявками, если он отрицателен, фиксируется аварийная остановка.

Особые ограничения

- Значения времени должны быть неотрицательны.
- Операнды A и D должны использоваться одновременно.
- GENERATE – единственный блок, который не может использоваться в ручном режиме моделирования для интерактивного создания заявок.

Режим отказа

Заявке никогда не будет отказано в доступе в блок GENERATE.

Вспомогательные окна

- CEC Snapshot (Текущее состояние CEC) – Изображение состояния цепи текущих событий в моделировании.
- FEC Snapshot (Текущее состояние FEC) – Изображение состояния цепи будущих событий в моделировании.

INDEX (ИНДЕКСИРОВАНИЕ)

Блок INDEX обновляет параметр активной заявки.

INDEX A,B

Операнды

A – Номер параметра. Параметр с исходным значением. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

B – Численное значение. Число, которое должно быть добавлено в содержание параметра. Результат направляется в параметр 1. Обязательный. В качестве операндов должны быть указаны *Name*, *Number*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Пример

INDEX 2,11.7

В этом пример, когда заявка попадает в блок INDEX, ее параметр номер 1 определяется как сумма 11,7 и значения параметра 2 активной заявки.

Алгоритм работы

Блок INDEX прибавляет численный эквивалент операнда B к численному эквиваленту значения любого параметра заявки и записывает результат в параметр 1.

Если требуемый (исходный) параметр не существует, фиксируется аварийная остановка. В противном случае, если для активной заявки не существует параметр номер 1, он создается.

Особые ограничения

- A должно быть положительным.

Режим отказа

Заявке никогда не будет отказано в доступе в блок INDEX.

Родственные SNA

- *PPparameter* или **Parameter* – Значение параметра. Возвращает значение параметра *Parameter*.

Вспомогательные окна

- Transaction Snapshot (Текущее состояние заявки) – Изображение состояния заявки в моделировании.

INTEGRATION (ИНТЕГРАЦИЯ)

Блок INTEGRATION разрешает или запрещает интеграцию пользовательских переменных.

INTEGRATION A,B

Операнды

A – Пользовательская переменная. Обязательный. В качестве операндов должно быть указано *Name*.

B – Состояние интеграции. В качестве операндов должно быть указано *Null*, ON или OFF.

Примеры

INTEGRATION Population

Это простейший способ использования блока INTEGRATION. Когда заявка поступает в блок INTEGRATION, интеграция пользовательской переменной *Population* устанавливается в разрешенное состояние. Это вызывает автоматическую интеграцию пользовательской переменной относительно времени моделирования. По умолчанию операнд B имеет значение ON.

INTEGRATION Population,OFF

В этом примере, поступление активной заявки в блок INTEGRATION обеспечивает запрет на интеграцию пользовательской переменной *Population*. Автоматическое обновление переменной *Population* будет прекращено с тех пор, пока переменная снова не будет разрешена.

Алгоритм работы

Блок INTEGRATION устанавливает состояние интеграции в положение ON или OFF, то есть, разрешено или запрещено. По умолчанию значение операнда B – ON, и оно должно быть определено, когда необходимо активизировать интеграцию.

Интеграция задается при помощи команд INTEGRATE и автоматически устанавливается в активное состояние. Для постоянной интеграции, которая никогда не будет прервана, при моделировании нет необходимости в блоке INTEGRATION. Каждая команда INTEGRATION может также установить одно или два граничных выражения. Когда значение интегрируемой переменной перемещается из одного края граничных значений в другой, новая заявка создается и отправляется в блок назначения, заданный в команде INTEGRATE. Таким же образом непрерывно интегрируемая переменная может быть частью отдельного процесса моделирования. Вы можете использовать их для выполнения важных действий, касающихся состояния интегрируемых переменных или, просто для перемещения границ.

Пользовательские переменные могут получать новые значения как непрерывно, так и через интеграцию. Для этого можно использовать команду EQU или оператор присваивания PLUS. Если вы хотите, чтобы такие назначения производились в процессе моделирования, вам необходимо определить процедуру PLUS, которая выполняет присваивание. Например, если вы определили процедуру PLUS следующим образом:

PROCEDURE Setpop(Pop_Level) BEGIN

Population = Pop_Level ;

END ;

Вы можете реинициализировать пользовательскую переменную *Population* при помощи использования блока PLUS следующим образом:

PLUS (Setpop(200))

или же при помощи использования выражения в скобках, которое осуществит действие setpop() в каком-либо другом блоке.

Смотри главу 4 для детального рассмотрения непрерывного моделирования.

Режим отказа

Заявке никогда не будет отказано в доступе в блок INTEGRATION.

Родственные Команды

- INTEGRATE – Определение производной от пользовательской переменной для интеграции и активации интеграции.

Вспомогательные окна

- Expressions Window (Окно выражений) – Просмотр значения выражений в реальном времени.
- Plot Window (Окно сценариев) – Просмотр содержимого более 8 выражений в реальном времени.

JOIN (ОБЪЕДИНЕНИЕ)

Блок JOIN добавляет активные заявки в группу заявок или добавляет число в числовую группу.

JOIN A,B

Операнды

A – Номер группы. Группа, в которую будет добавлен член. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

B – Численно значение. Число, которое будет добавлено в численную группу. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Пример

JOIN SoldItems

Это простейший способ использования блока JOIN. Заявка поступает в блок и становится членом группы заявок *SoldItems*.

Алгоритм работы

Блок JOIN функционирует в числовом режиме, когда число определено операндом B, в противном случае, – в режиме заявок.

В числовом режиме операнды A и B вычисляются численно и число, определенное операндом B включается в числовую группу, определенную операндом A. Если это число уже является членом группы, никакое действие не производится. Имеет место некоторая потеря эффективности при использовании в численной группе нецелых чисел.

В режиме заявок, поступившая на вход блока заявка включается в группу заявок, указанную в операнде A. Если заявка уже является членом этой группы, никакое действие не производится.

Числовые группы не зависят от групп заявок, даже если их номера совпадают.

Заявки ничем не ограничены благодаря своей принадлежности к группе заявок. После того как заявка становится членом группы, она направляется к следующему блоку. Единственный способ, посредством которого заявка может быть удалена из группы, – обработка ее в блоках TERMINATE или ASSEMBLE, или определение ее на удаление в блоке REMOVE.

Особые ограничения

- A должно быть положительным.

Режим отказа

Заявке никогда не будет отказано в доступе в блок JOIN.

Родственные блоки

Заявки могут быть направлены в группы заявок блоками: ALTER, EXAMINE, REMOVE и SCAN.

Числа в числовые группы могут быть определены блоками: EXAMINE и REMOVE.

Родственные SNAs

- *GNEntnum* – Счетчик числовой группы. *GNEntnum* возвращает количество членов числовой группы *Entnum*.

- *GTEntnum* – Счетчик группы заявок. *GTEntnum* возвращает количество членов группы заявок *Entnum*.

Вспомогательные окна

- Numeric Groups Snapshot (Текущее состояние числовой группы) – Изображение состояния числовой группы в моделировании.
- Transaction Groups Snapshot (Текущее состояние группы заявок) – Изображение состояния группы заявок в моделировании.

LEAVE (УХОД)

Блок LEAVE увеличивает количество доступных ячеек памяти.

LEAVE A,B

Операнды

A – Номер или имя памяти. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

B – Количество ячеек (единиц) памяти. Значение по умолчанию 1. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Пример

LEAVE RepairMen,10

В этом примере, когда заявка поступает в блок УХОДА, количество доступных ячеек памяти *RepairMen* увеличивается на 10.

Алгоритм работы

Когда заявка поступает в блок LEAVE, операнд A вычисляется и округляется, и находится соответствующая область памяти. Если такой области не существует, фиксируется аварийная остановка. Область памяти должна быть определена при помощи команд STORAGE.

Количество освобождаемых единиц памяти находится при вычислении и округлении операнда B. Если операнд B не используется, берется значение по умолчанию 1.

Затем доступная область памяти увеличивается на значение операнда B. Если результат превышает реальную емкость памяти, фиксируется аварийная остановка.

Если ошибки не зафиксировано, заявка направляется в следующий блок и помещается в СЕС перед равными по приоритету заявками.

Когда память становится доступна, очередь задержки к памяти проверяется в порядке понижения приоритетов на предмет нахождения заявки, чей запрос теперь может быть удовлетворен. Выбираются первые подходящие заявка. Им позволяется поступить в блок ENTER, который отклоняет их затем, помещает в СЕС за равными и по приоритету заявками. Это делается до того, как текущая активная заявка в блоке LEAVE продолжает моделирование. Таким образом, ни одна другая заявка не может нарушить очередность заявок, ожидающих в очереди задержки к памяти. Это можно увидеть в окне блоков. В нем будет видно, как заявка поступает в блок ENTER и остается в нем. Текущая активная заявка в блоке LEAVE только затем возобновляет движение.

Особые ограничения

- A должно обращаться к заранее определенной области памяти, созданной командами STORAGE.
- B должно быть неотрицательным.
- Попытка освободить больший объем памяти, чем был определен изначально, приведет к аварийной остановке.

Режим отказа

Заявке никогда не будет отказано в доступе в блок LEAVE

Родственные команды

Область памяти должна быть определена командами STORAGE до того, как она будет обновлена в блоке LEAVE. Команды STORAGE должны быть определены в модели или

должны быть переданы имитационному объекту интерактивно, до того как заявка сможет войти в блок LEAVE. Любая попытка сделать это до того, как память будет определена, приведет к аварийной остановке. Область памяти может быть переопределена при помощи интерактивных команд STORAGE.

Родственные SNAs

- *REntnum* – Неиспользуемая емкость памяти. Объем памяти (или "маркер" пустоты) доступный для использования поступающей заявкой или содержимым самой памяти *Entnum*.
- *SEntnum* – Занятая память. *SEntnum* возвращает объем содержащейся в памяти информации (или "маркер" пустоты) использующийся в данный момент активной заявкой или содержимым самой памяти *Entnum*.
- *SAEntnum* – Средняя емкость используемой памяти. *SAEntnum* возвращает среднюю по времени емкость используемой памяти (или "маркер" пустоты) *Entnum*.
- *SCEntnum* – Объем используемой памяти. Общее количество единиц (ячеек) памяти, которые были заняты (или "маркер" пустоты) *Entnum*.
- *SEEntnum* – Память пуста. *SEEntnum* возвращает 1, если память *Entnum* не используется, 0 в противном случае.
- *SFEntnum* – Память заполнена. *SFEntnum* возвращает 1, если память *Entnum* заполнена, 0 в противном случае.
- *SREntnum* – Использование памяти. Часть общего использования, представленная в виде средней занятости памяти *Entnum*. *SREntnum*, выражается в долях тысячи и, следовательно, возвращает вещественное значение от 0 до 1000, включительно.
- *SMEntnum* – Максимальный объем используемой памяти *Entnum*. "Высшая планка".
- *STEntnum* – Среднее время хранения единицы памяти *Entnum*.
- *SVEntnum* – Состояние памяти. *SVEntnum* возвращает 1, если память *Entnum* доступна, 0 в противном случае.

Вспомогательные окна

- Storages Window (Окно памяти) – Просмотр изменения содержимого памяти в реальном времени.

LINK (СОЕДИНЕНИЕ)

Блок LINK управляет местоположением Активной Заявки в Поток Заявок.

LINK A,B,C

Операнды

A – Номер очереди. Очередь которая может принять поступающую заявку. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

B – Систематизация очереди. Местоположение новой заявки в очереди. Обязательный. В качестве операндов должны быть указаны LIFO, FIFO, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

C – Расположение следующего блока. Блок назначения для заявок, для которых индикатор соединения (Link Indicator) очереди находился в положении Выкл. (reset). Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Пример

LINK OnHold,FIFO

В этом примере активная заявка помещается в конец накопительной системы (*User Chain*) *OnHold*. Она удаляется из всех очередей, кроме группы заявок и очереди прерываний. Другими словами, она не освобождается от приоритета. Заявка оставляется в

накопительной системе до тех пор, пока некоторая другая заявка не поступит в блок UNLINK и не удалит ее. В приведенном примере заявка помещается в конец накопительной системы *OnHold*.

Алгоритм работы

Простейшая операция блока LINK вызывается, когда операнд С не используется. В этом случае, поступившая заявка всегда помещается в накопительную систему, определенную операндом А. Она удаляется из всех остальных очередей, кроме групп заявок и очереди прерываний.

Местоположение заявки в накопительной системе управляется операндом В. Если выставлен параметр FIFO (First-In-First-Out), вновь прибывшая заявка будет помещена в конец накопительной системы. Если указан параметр LIFO (Last-In-First-Out), вновь прибывшая заявка будет помещена в начало накопительной системы. Все остальные параметры, используемые в операнде В вычисляются для активной заявки и заново для всех заявок в накопительной системе, начиная с начала, пока значение активной заявки больше. Когда заявка, значение которой меньше, чем у активной. Будет найдена, активная заявка помещается в накопительной системе перед ней. Если значение активной заявки никогда не превысит значение уже находящихся в накопительной системе заявок, активная заявка поместится в конец очереди. Это соответствует порядку по убыванию. Исключением является класс SNAs Р. Они помещаются в порядке возрастания значения параметра.

Если вы не используете параметры LIFO или FIFO в операнде В, заявки располагаются в соответствии с классами SNA, такими как PR, M1 или Р. Тем не менее, не прямая адресация также может быть полезна. Если используется класс PR, Заявки помещаются в порядке понижения приоритетов. И, в итоге, если номер параметра определен, заявка помещается в очередь сразу за заявками, значения параметров которых меньше, чем у поступившей заявки.

Моделирование усложняется при использовании операнда С.

Флаг, называемый индикатором соединения (Link Indicator) является частью любой накопительной системы. Индикатор соединения полезен при использовании накопительных систем для управления очередностью заявок в ресурсе. Он включен (set), если предполагаемый ресурс «занят» и выключен (reset), если предполагаемый ресурс «свободен». Когда индикатором соединения накопительной системы выключен, если операнд С используется, блок LINK не поместит заявку в накопительную систему. Индикатор соединения полезен при выдаче разрешения первой из нескольких заявок освободить накопительную систему и для обусловливания помещения последующих заявок в очередь.

Если операнд С используется, поступившая заявка НЕ будет помещена в накопительную систему, если индикатор соединения выключен. Вместо этого, заявка будет отправлена в блок, определенный операндом С, и затем индикатор соединения будет включен. Последующие заявки, поступившие в блок LINK, будут помещены в накопительную систему.

Индикатор соединения управляется одновременно блоками LINK и UNLINK. Он выключается, если блок UNLINK обнаруживает, что накопительная система пуста. Полезно было бы истолковать работу индикатора соединения как «условие занятости» предполагаемого ресурса. Когда индикатор соединения включен, ресурс занят. Когда заявка обнаруживает, что ресурс не используется, ей не приходится ожидать в очереди. Заявка будет направлена в блок, определенный операндом С и блок LINK затем включит индикатор соединения накопительной системы.

Рассмотрим прибытие двух заявок в блок LINK, одну за другой. Если, например, первая заявка не поступила в блок UNLINK перед тем как прибыла вторая заявка, вторая заявка обнаружит, что индикатор соединений включен и будет помещена в накопительную систему. В этом примере, когда первая заявка поступает в блок UNLINK, вторая удаляется

из накопительной системы. В это время индикатор соединений остается включенным. Затем, когда вторая заявка поступает в блок UNLINK, и в накопительной системе не остается ни одной ожидающей заявки, индикатор, в конце концов, выключается.

Накопительная система позволяет манипулировать механизмами очередей заявок на гораздо более сложном уровне, чем это позволяют делать устройства и память. При использовании блоков LINK и UNLINK становится возможным выполнение сложнейших алгоритмов планирования.

Накопительная система может использоваться для уменьшения количества холостого использования компьютерного времени на неудачные проверки в блоках GATE, TEST, TRANSFER BOTH, TRANSFER ALL. Можно ускорить процесс моделирование посредством помещения заблокированных Заявок в накопительную систему, когда для них нет возможности успешного прохождения условий проверки. Затем нужно будет вернуть каждую заявку обратно в тестирующий блок, когда появится возможность ее разблокировать.

Особые ограничения

- А, В, и С, если они определены, должны быть положительны.
- С, если оно определено, должно указывать на местоположение существующего блока.

Режим отказа

Заявке никогда не будет отказано в доступе в блок LINK.

Родственные SNAs

- *SAEntnum* – Среднее количество заявок в накопительной системе. Среднее по времени количество заявок в накопительной системе *Entnum*
- *CCEntnum* – Общее количество заявок в накопительной системе. Сумма всех поступивших в накопительную систему заявок *Entnum*
- *CHEntnum* – Текущее количество заявок в накопительной системе. Текущее количество заявок в накопительной системе *Entnum*.
- *CMEntnum* – Максимальное количество заявок в накопительной системе. Максимальное количество заявок в накопительной системе *Entnum*. «Максимальная отметка».
- *CTEntnum* – Среднее время пребывания в накопительной системе. Средняя продолжительность пребывания заявки в накопительной системе *Entnum*.

Вспомогательные окна

- Userchains Snapshot (Текущее состояние накопительной системы) – Изображение состояния накопительной системы в реальном времени.

LOGIC (ЛОГИКА)

Блок LOGIC переключает логические схемы

LOGIC O A

Операнды

O – Логический оператор. Обязательный. В качестве операндов должны быть указаны S, R или I.

A – Номер логической схемы. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Пример

LOGIC S PowerSwitch

В этом примере логическая схема *PowerSwitch* отаается в состоянии «истина» или «включена».

Алгоритм работы

Блок LOGIC используется для включения, выключения или инвертирования лгических сем. логическая схема имеет два состояния: включена (1) и выключена (0). Если в качестве логического оператора указаны S или R, состояние логической схемы

определяется операндом А, приведенном во включенное или выключенное состояние, соответственно.

Если в качестве логического оператора указан I, состояние логической схемы, определяемое Операндом А, инвертируется. Это означает, что если он был включен, он будет выключен. А если был выключен, то будет включен.

Логические операторы

Логические операторы являются обязательной составляющей блока. Они могут принимать значение S, R или I по следующим правилам:

- S – Логический переключатель включен (set).
- R – Логический переключатель выключен (reset).
- I – Логический переключатель инвертирован.

Режим отказа

Заявке никогда не будет отказано в доступе в блок LOGIC.

Родственные SNA

- *LSEntnum* – Логический переключатель включен. *LSEntnum* возвращает 1 если логический переключатель *Entnum* включен, 0 в противном случае.

Вспомогательные окна

- *Logicswitches Window* (Окно логических переключений) – Просмотр действия логических схем в реальном времени.

LOOP (СВЯЗЬ)

Блок LOOP изменяет параметр и управляет местом назначения активной заявки на основе полученного результата.

LOOP A,B

Операнды

A – Счетчик содержимого параметра. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

B – Номер блока. Следующий блок, если счетчик не обнулится после декрементирования. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Пример

LOOP Customer_Count,Start_Over

Предположим, что блок *Start_Over* предшествует в моделировании блоку LOOP. В этом примере, когда заявка поступает в блок LOOP. Ее параметр *Customer_Count* уменьшается на 1. Если результат не равен нулю, заявка перемещается в блок *Start_Over*. Это обуславливает циркуляцию заявки до тех пор, пока значение параметра *Customer_Count* не будет равно нулю. Тогда заявка отправится на следующий блок.

Алгоритм работы

Когда заявка попадает в блок LOOP, операнд A вычисляется, округляется и используется для нахождения параметра заявки по этому номеру. Если такого параметра не существует, фиксируется аварийная остановка. В противном случае, значение параметра уменьшается на 1.

Если новое значение параметра больше нуля и операнд B определен, заявка направляется к определенному операндом B месту назначения. В противном случае, заявка отправляется к следующему блоку.

Особые ограничения

- A должно быть положительным.

Режим отказа

Заявке никогда не будет отказано в доступе в блок LOOP.

Родственные SNA

RParameter или **Parameter* – Значение параметра. *RParameter* возвращает значение параметра *Parameter*.

Вспомогательные окна

- Blocks Window (Окно блоков) – Просмотр смены Блоков в реальном времени.
- Transaction Snapshot (Текущее состояние заявки) – Изображение состояния заявки в моделировании.

MARK (МАРКИРОВКА)

Блок MARK заносит значение абсолютного системного времени в параметр активной заявки.

MARK A

Операнд

A – Номер параметра. Параметр, получающий значение системного времени. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Примеры

MARK Beginning

В этом примере, когда заявка попадает в блок MARK, ее параметру *Beginning* присваивается значение, равное значению абсолютного системного времени, AC1.

MARK

В этом примере, когда заявка попадает в блок MARK, ее времени маркировки присваивается значение, равное значению абсолютного системного времени.

Алгоритм работы

Когда заявка попадает в блок MARK и операнд A не определен, ее времени маркировки присваивается значение, равное значению абсолютного системного времени

Если операнд A определен, он численно вычисляется и округляется. Параметру активной заявки с найденным номером присваивается значение, равное значению абсолютного системного времени. Если параметр не существует, он создается.

Затем активная заявка направляется в следующий блок (NSB – Next Sequential Block).

Значение времени, установленное блоком MARK может быть извлечено классами SNAs M1 и MP. M1 возвращает время пребывания требования в системе, которое вычисляется как абсолютное системное время минус время маркировки. MP возвращает значение, равное абсолютному системному времени минус значение параметра заявки.

Особые ограничения

- A, если оно определено, должно быть положительным.

Родственные SNAs

- *MPEntnum* – Время пребывания требования в системе, параметр. Текущее значение абсолютного системного времени минус значение параметра *Entnum*.
- M1 – Время пребывания требования в системе. M1 возвращает значение абсолютного системного времени минус время маркировки заявки.

Режим отказа

Заявке никогда не будет отказано в доступе в блок MARK.

Вспомогательные окна

- Transaction Snapshot (Текущее состояние заявки) – Изображение состояния заявки в моделировании.
- CEC Snapshot (Текущее состояние CEC) – Изображение состояния цепи текущих событий в моделировании.
- FEC Snapshot (Текущее состояние FEC) – Изображение состояния цепи будущих событий в моделировании.

MATCH (СОГЛАСОВАНИЕ)

Пара блоков MATCH заставляют заявки ожидать друг друга.

MATCH A

Операнд

A – Номер или имя блока. Сопряженный блок MATCH, который будет проверяться на согласование (совпадение набора) заявки. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Пример

**A_Is_Done MATCH B_Is_Done
B_Is_Done MATCH A_Is_Done**

В этом примере рассматриваются два сопряженных блока MATCH. Они обычно помещаются в системе моделирования и один, в конечном счете, принимает заявку в тот же набор. Если блок MATCH расположен за набором блоков, осуществляющих некий процесс, пара родственных (согласованных) заявок пройдет через блок MATCH только когда обе заявки закончат соответствующий процесс.

Алгоритм работы

Когда заявка поступает в блок MATCH, операнд A численно вычисляется, округляется, и используется для нахождения сопряженного блока MATCH. Если такого блока не существует, фиксируется аварийная остановка.

Если сопряженный блок MATCH содержит (в своей цепи согласования) заявку из того же набора, родственная заявка удаляется из цепи согласования. Если в текущий момент ее обработка не была прервана на каком-либо устройстве, она помещается в СЕС за заявками с таким же приоритетом. Также, если обработка активной заявки не была прервана на каком-либо устройстве, она помещается в СЕС, но перед заявками с таким же приоритетом.

Если одна из согласуемых заявок была в текущий момент прервана на устройстве, ей не разрешается покинуть блок MATCH до тех пор, пока все приоритетные прерывания обслуживания не будут разблокированы.

Если, когда активная заявка поступает в блок MATCH, ни одной согласуемой заявки не обнаружено, она направляется на ожидание в цепь согласования блока MATCH.

Особые ограничения

- Операнд A должен определять местоположение существующего блока MATCH.
- Прерванным заявкам не разрешается покинуть блок MATCH.

Режим отказа

Заявке никогда не будет отказано в доступе в блок MATCH

Родственные блоки

- ADOPT – Устанавливает наборы (Assembly Set) активных заявок
- ASSEMBLE – Ожидает и ликвидирует членов наборов.
- GATHER – Ожидает членов набора.
- SPLIT – Создает заявки из того же набора.

Родственные SNAs

- A1 – Набор. Возвращает набор активной заявки.
- MBEntnum – Согласование в блоке. MBEntnum возвращает 1, если в блоке Entnum есть заявка из того же набора, что и активная. В противном случае, MBEntnum возвращает 0.

Вспомогательные окна

- Blocks Window (Окно блоков) – Просмотр смены блоков в реальном времени.
- Transaction Snapshot (Текущее состояние заявки) – Изображение состояния заявки в моделировании.
- CEC Snapshot (Текущее состояние СЕС) – Изображение состояния цепи текущих событий в моделировании.
- FEC Snapshot (Текущее состояние FEC) – Изображение состояния цепи будущих событий в моделировании.

MSAVEVALUE (МАТРИЦА)

Блок MSAVEVALUE обновляет значения элементов матрицы.

MSAVEVALUE A,B,C,D

Операнды

A – Номер или имя матрицы, с необязательным указанием + или –. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*, с указанием в конце +, –, или *Null*.

B – Номер строки. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

C – Номер столбца. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

D – Значение, которое необходимо сохранить, прибавить или вычесть. Обязательный. В качестве операндов должны быть указаны *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Примеры

MSAVEVALUE Sales+,Part23,Cust77,6.234

В этом примере, когда заявка поступает в блок MSAVEVALUE, элемент матрицы *Sales*, с номером строки *Part23* и номером столбца *Cust77* увеличивается на 6.234. Названия строк и столбцов необходимо задать ранее в выражениях EQU.

MSAVEVALUE Parts,Part23,Description,"Zippo lighter"

В этом примере строка добавляется в матрицу в качестве элемента. Матрица должна быть определена в любом месте модели или же задана интерактивно, при помощи команды MATRIX, которая задает элементы матрицы.

Алгоритм работы

Когда заявка поступает в блок MSAVEVALUE, операнд A численно вычисляется и по его значению находится соответствующая матрица. Если такой матрицы не существует, фиксируется аварийная остановка. матрица должна быть определена при помощи группы команд MATRIX.

Операнды B и C численно вычисляются для нахождения требуемого элемента матрицы. Если такого элемента не существует, фиксируется аварийная остановка.

Операнд D высчитывается и используется в операции обновления. Если в операнде A в качестве суффикса указан +, эта операция будет заключаться в прибавлении к элементу численного значения операнда D; если указан –, это будет вычитание. Если суффикс для операнда A не задан, элементу матрицы будет присвоено новое значение, равное численному эквиваленту операнда D.

Матрица задается командой MATRIX. Она может содержать более 6 измерений. Однако только первые два измерения могут быть доступны блок MSAVEVALUE. Поэтому, все оставшиеся координаты считаются равными 1.

Процедура сложения PLUS может получать доступ к любым элементам матриц. Если необходимо использовать матрицы с количеством измерений, больше двух, придется создавать одну или более процедур сложения PLUS для доступа к ним. матрицы, определенные командой MATRIX имеют глобальный контекст и доступны всем процедурам сложения. Кроме того, для вызова процедуры PLUS могут быть созданы и временные локальные матрицы. Это будет обсуждаться позже в главе 8.

Режим отказа

Заявке никогда не будет отказано в доступе в блок MSAVEVALUE.

Особые ограничения

- Блоки MSAVEVALUE могут получать доступ только к первым двум измерениям многомерных матриц. Индексам третьего и выше измерений будет присвоено значение 1. Для доступа и изменения любых элементов может быть использована процедура PLUS.

Родственные SNA

- $MXEntnum(m,n)$ – Значение элемента матрицы. Возвращается значение из строки m , столбца n матрицы $Entnum$. Значениями строки и столбца могут быть $Name$, $PosInteger$, или $P class SNA$

Вспомогательные окна

- Expressions Window (Окно выражений) – Просмотр значения выражений в реальном времени.
- Plot Window (Окно сценариев) – Просмотр содержимого более 8 выражений в реальном времени.
- Matrix Window (Окно матрицы) – Возможность просмотра изменения значений элементов матрицы в реальном времени.

OPEN (ИНИЦИАЛИЗАЦИЯ)

Блок OPEN инициализирует поток данных.

OPEN A,B,C

Операнды

A – Дескриптор потока данных. Обязательный. В качестве операндов должны быть указаны $Name$, $PosInteger$, $String$, $ParenthesizedExpression$, SNA или $SNA*Parameter$.

B – Номер потока данных. Необязательный. В качестве операндов должны быть указаны $Null$, $Name$, $PosInteger$, $ParenthesizedExpression$, SNA or $SNA*Parameter$. Значение по умолчанию 1.

C – Номер или имя альтернативного блока назначения. Необязательный. В качестве операндов должны быть указаны $Null$, $Name$, $PosInteger$, $ParenthesizedExpression$, SNA , или $SNA*Parameter$.

Пример

OPEN ("MYFILE.TXT"),2,Error_Block

В этом примере блок OPEN создает тип потока данных, известный как поток ввода/вывода, и присваивает ему идентификационный номер 2. Если в процессе инициализации фиксируется ошибка, код ошибки сохраняется внутри и активная заявка направляется в блок, помеченный как блок *Error_Block*.

Алгоритм работы

Блок OPEN обеспечивает создание потока данных и устанавливает указатель текущей позиции в 1.

Поток данных – это последовательность текстовых строк, используемый в среде моделирования GPSS. Каждый поток данных идентифицируется уникальным номером. Существуют 3 типа потоков данных:

1. Потоки ввода/вывода (I/O Streams),
2. Потоки памяти (In-Memory Streams),
3. Канальные потоки (Pipe Streams).

операнд A определяется как строка. Если это нулевая строка, создается поток памяти. Если это название канала (например, "\pipe\mypipe"), создается канальный поток. Иначе, создается поток ввода/вывода и операнд A рассматривается в качестве спецификации файла. Если путь к файлу не включен в его спецификацию, предполагается, что используется директория отчета о проекте Project's Report. Проекты обсуждаются в главе 1 этого руководства.

Если операнд B используется, он вычисляется численно и используется в качестве номера потока данных, для дальнейшего обращения по нему блоков READ, WRITE и CLOSE. Значение по умолчанию 1.

Если используется операнд C, любая ошибка, возникающая в процессе инициализации, вынуждает активную заявку направляться в блок с указанным в нем номером.

В любом случае, если обнаружена ошибка, ее код сохраняется внутри. Блок OPEN может использоваться для извлечения кода ошибки.

Глава 4 (4.16) содержит полное описание потоков данных, включая описание кодов ошибок.

Условия Блокировки

Моделирование блокируется на время инициализации потока данных.

Режим отказа

Заявке никогда не будет отказано в доступе в блок OPEN.

Родственные блоки

- CLOSE – Закрывает поток данных.
- READ – Извлекает текстовую строку из потока данных.
- WRITE – Записывает текстовую строку в потока данных.
- SEEK – устанавливает указатель текущей позиции потока данных.

PLUS (СЛОЖЕНИЕ)

Блок PLUS вычисляет выражение и, по возможности, помещает результат в Параметр.

PLUS A,B

Операнды

A – Выражение. Обязательный. В качестве операндов должны быть указаны *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

B – Номер параметра. Параметра, в который записывается результат. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Пример

PLUS (normal(2,100,2)+100.26),Result_Parm

Алгоритм работы

Когда заявка поступает в блок PLUS, вычисляется операнд A.

Если операнд B определен, он вычисляется численно, округляется и используется для идентификации параметра активной заявки. Если такого параметра не существует, он создается. Затем, результат, вычисленный в операнде A, помещается в него.

Процедуры сложения PLUS, если возвращаемое значение явно не задано, возвращают значение 0.

Режим отказа

Заявке никогда не будет отказано в доступе в блок PLUS.

Вспомогательные окна

- Expressions Window (Окно выражений) – Просмотр значения выражений в реальном времени.
- Plot Window (Окно сценариев) – Просмотр содержимого более 8 выражений в реальном времени.

PREEMPT (ПЕРЫВАНИЕ)

Блок PREEMPT смещает занимающую устройство заявку.

PREEMPT A,B,C,D,E

Операнды

A – Номер или имя устройства. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

B – Приоритетный режим. PR для приоритетного режима или режима прерываний, если не указано ничего. Необязательный. В качестве операндов должны быть указаны PR или *Null*.

C – Имя или номер блока. Новое место назначения для заявки, занимавшей устройство. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

D – Номер параметра. Параметр прерванной заявки для получения оставшегося времени обработки прерванной заявки, если прерванная заявка удаляется из FEC. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

E – Режим удаления (REmove Mode). Удаляет прерванную заявку из группы претендующих на устройство заявок. Необязательный. В качестве операндов должны быть указаны RE, или *Null*. При использовании RE, необходимо указать место назначения в операнде C.

Примеры

PREEMPT Teller1

Это простейший способ использования блока PREEMPT. Когда заявка поступает в блок PREEMPT, ей дается право занять устройство до тех пор, пока его работа не будет прервана. Если заявка должна освободить устройство, ей не разрешается постоянно храниться в FEC, и ее передвижение по системе моделирования ограничивается.

PREEMPT Teller1,,Teller2,101,RE

В этом примере Блок PREEMPT функционирует в режиме прерываний, потому как операнд B пропущен. Это означает, что заявка может войти в блок и занять устройство, если оно еще не занято другой заявкой. Приоритеты заявок игнорируются. Когда заявка прервана, все оставшееся время в Блоке ADVANCE записывается в параметр 101 прерванной заявки. Прерванная заявка удаляется из группы претендующих на устройство заявок и отправляется в Блок помеченный как Teller2. Ошибка зафиксирована, если такая Заявка позже попытается RELEASE (ОСВОБОДИТЬ) или RETURN (ВЕРНУТЬСЯ) на устройство *Teller1*, не заняв его снова.

PREEMPT Teller1,PR,Teller2

В этом примере блок PREEMPT работает в приоритетном режиме. Это означает, что заявка может войти в блок и занять устройство, если устройство уже не занято заявкой с равным или более высоким приоритетом. Любая прерванная заявка не удаляется из группы претендующих на устройство заявок. Это означает, что такая заявка долее не может находиться в FEC или покидать блоки ASSEMBLE, GATHER или MATCH. С учетом этого ограничения, прерванные заявки могут продолжить пребывание в системе моделирования и, в этом случае, будут отправлены в блок *Teller2*. Такие заявки, за которыми было оставлено право претендовать на устройство *Teller1*, обычно, вновь занимают его спустя некоторое время. Прерванные заявки могут ВЕРНУТЬСЯ (RETURN) на устройство или ОСВОБОДИТЬ (RELEASE), даже если они его не занимают. Заявки удаляются из группы претендующих на устройство посредством удаления их из цепи прерывания устройства.

Алгоритм работы

Блок PREEMPT действует подобно блоку SEIZE, когда устройство свободно. В противном случае, блок PREEMPT работает либо в приоритетном режиме, либо в режиме прерывания. Это определяется операндом B.

В приоритетном режиме, только заявка с более высоким приоритетом может вытеснить занимающую устройство заявку. Если же приоритета заявки для этого недостаточно, она помещается в очередь ожидания устройства (очередь прерванной обработки, Delay Chain) в порядке значения приоритета.

В режиме прерывания, если работа устройства уже прервана, активная заявка помещается в очередь неоконченной обработки (Pending Chain). Заявкам в очереди неоконченной обработки предоставляется право занять устройство перед прерванными заявками или заявками в очереди прерванной обработки.

Операнды C и E определяют, какое действие необходимо произвести с заявкой, занимающей устройство в данный момент, обработка которой будет прервана. Прерванные заявки не разрешено оставаться в FEC. Прерванные заявки, у которых еще не закончилось время в блоке ADVANCE, могут быть помещены в FEC автоматически,

пропустив операнды C и D. В противном случае, если вы самостоятельно решите поместить заявку в FEC, для возобновления неоконченного времени необходимо использовать операнд D и, в конечном счете, отправить прерванную заявку в блок ADVANCE.

Прерванная заявка может быть удалена из группы претендующих на устройство заявок (т.е., из всех очередей на устройство) при помощи использования параметра RE операнда E. Опция RE удаляет ограничения, наложенные на прерванную заявку после ее прерывания на устройстве, и делает любые последующие попытки ВЕРНУТЬСЯ (RETURN) на устройство или ОСВОБОДИТЬ (RELEASE) устройство условием ошибки и остановки.

Прерванная заявка не может существовать в FEC. Более детальное обсуждение прерываний можно найти в разделе 9.4. Все вновь прерванные заявки в блоке ADVANCE, находящиеся в FEC, удаляются из FEC с сохранением значения времени оставшейся обработки. Если операнд D используется, время оставшейся обработки также сохраняется в параметре заявки. Если такого параметра не существует, он создается. Время оставшейся обработки используется для перенаправления заявки в FEC, когда она снова получит право занять все устройства, на которые претендует. В противном случае, можно передать прерванную заявку новому блоку при помощи операнда C.

Прерванная заявка оставляется в группе претендующих на устройство, даже если она была перемещена при помощи операнда C, до тех пор, пока в операнде E будет выставлен параметр RE. Если прерванная заявка, которая до сих пор претендует на устройство, попытается войти в блок TERMINATE, это приведет к аварийной остановке. Такие заявки должны поступить в блоки RELEASE или RETURN до того, как им будет позволено завершиться (TERMINATE).

В противном случае, если вы намереваетесь ЗАВЕРШИТЬ (TERMINATE) прерванную заявку, можно использовать параметр RE, чтобы удостовериться, что она случайно вновь возобновила свое право занять устройство перед завершением. Такая заявка не может ОСВОБОДИТЬ (RELEASE) или ВЕРНУТЬСЯ (RETURN) на устройство.

Когда заявка перемещается при помощи операнда C, она передается в новый блок, удаляется из очередей:

- FEC (Future Events Chain) (цепи будущих событий)
- PENDING (INTERRUPT-MODE PREEMPT) CHAINS (очереди неоконченной обработки в приоритетном режиме прерывания)
- DELAY (MAJOR PRIORITY) CHAINS (очереди прерванной по старшинству приоритетов обработки)
- USER CHAINS (накопительной системы)
- RETRY CHAINS (повторной очереди)

И не удаляется из очередей:

- CEC (Current Events Chain) (цепи текущих событий)
- INTERRUPT (preempted) CHAINS (очереди приоритетных прерываний)
- GROUP CHAINS (общей очереди)

Когда заявка изымается из места ее текущего пребывания с использованием альтернативного места назначения, она удаляется из блокирующих условий, но приоритетное прерывание обслуживания разблокировано не будет.

Когда операнд C не используется, прерванная заявка изымается из FEC и будет возвращена туда автоматически. Прерванные заявки, которые не были перемещены при помощи операнда C, ожидают поступления в блоки RELEASE) или RETURN для возобновления права занимать устройство. Если такая заявка прибывает в блоки RELEASE или RETURN до восстановления прав на устройство, заявка удаляется из группы претендующих на устройство заявок. Условия для фиксации ошибки отсутствуют. Заявка может быть прервана на произвольном количестве устройств и продолжать находиться в системе моделирования.

Тем не менее, существуют два ограничения:

- Ей будет отказано в доступе в Блок ADVANCE с положительным значением времени
- Ей не будет позволено покинуть Блоки ASSEMBLE, GATHER или MATCH до тех пор, пока все ее приоритетные прерывания обслуживания не будут разблокированы.

Работа устройства может быть прервана произвольное количество раз. Тем не менее, однажды прерванная заявка не сможет вновь пытаться занять устройство, на котором она была прервана. Заявка может быть прервана на произвольном количестве устройств.

Особые ограничения

- Если на Е установлен параметр RE, С должно использоваться.
- Заявка не может прерывать саму себя.

Режим отказа

Заявке будет отказано в поступлении в блок PREEMPT, если в режиме прерываний устройство в текущий момент было прервано. Такие заявки помещаются в конец очередь неоконченной обработки устройства.

Заявке будет отказано в поступлении в блок PREEMPT, если в приоритетном режиме устройство в текущий момент занято заявкой с равным или более высоким приоритетом, чем у активной заявки. Активная заявка помещается в порядке приоритета в очередь прерванной обработки устройства.

Заявке будет отказано в поступлении в блок PREEMPT, если устройство находится в недоступном состоянии. Такие заявки помещаются в очередь Прерванной обработки устройства в порядке приоритета, в соответствии с дисциплиной FIFO внутри приоритетов.

Родственные блоки

- DISPLACE – Перемещает любую заявку.
- FAVAIL – Устанавливает устройство в доступное состояние.
- FUNAVAIL – Устанавливает устройство в недоступное состояние.
- RELEASE – отказывает в праве занимать устройства или удаляет заявку из группы претендующих на устройство заявок
- RETURN – отказывает в праве занимать устройства или удаляет заявку из группы претендующих на устройство заявок
- SEIZE – Запрашивает разрешение или ждет возможности занять устройство.

Родственные SNA

- *FEntnum* – Занятость устройства. Если устройство *Entnum* в данный момент занято, *FEntnum* возвращает 1. В противном случае, *FEntnum* возвращает 0.
- *FCEntnum* – Количество итераций работы устройства. Сколько раз устройство *Entnum* было занято заявкой.
- *FIEntnum* – Прерывания работы устройства *Entnum*. Если работа устройства *Entnum* в данный момент была прервана, *FIEntnum* возвращает 1. В противном случае, *FIEntnum* возвращает 0.
- *FREntnum* – Использование устройства. Время, в течение которого устройство *Entnum* было занято. *FREntnum* выражается в долях тысячи и, следовательно, возвращает вещественное значение от 0 до 1000, включительно.
- *FTEntnum* – Среднее время занятости (удерживания) устройства. Среднее время, в течение которого устройство *Entnum* было занято заявкой.
- *FVEntnum* – Доступность устройства. *FVEntnum* возвращает 1, если устройство *Entnum* доступно, 0 в противном случае.

Вспомогательные окна

- Facilities Window (Окно устройств) – Возможность просмотра изменения состояния устройств в реальном времени

- CEC Snapshot (Текущее состояние CEC) – Изображение состояния цепи текущих событий в моделировании.
- FEC Snapshot (Текущее состояние FEC) – Изображение состояния цепи будущих событий в моделировании.

PRIORITY (ПРИОРИТЕТЫ)

Блок PRIORITY устанавливает значение приоритета активной заявки.

PRIORITY A,B

Операнды

A – Новое значение приоритета. Обязательный. В качестве операндов должны быть указаны *Name, integer, String, ParenthesizedExpression, SNA*, или *SNA*Parameter*.

B – Буферная опция. Помещает активную заявку следом за равными ей по приоритету Заявками в CEC. Необязательный. В качестве операндов должны быть указаны BU или *Null*.

Пример

PRIORITY 10

В этом примере любой поступившей в блок заявке устанавливается приоритет 10.

Алгоритм работы

Когда заявка поступает в блок PRIORITY, операнд A численно вычисляется, округляется и устанавливается в качестве приоритета для активной заявки.

Заявка перенаправляется к следующему блоку и помещается в CEC в соответствии с новым значением приоритета. Если в операнда B используется опция BU, заявка помещается следом за равными ей по приоритету заявкам. В противном случае, она помещается перед ними.

Приоритеты заявок – это вещественные числа. Когда заявка при создании заявки приоритет не задается явно, по умолчанию, ей присваивается приоритет 0. Среда GPSS работает наиболее эффективно, когда используемые в моделировании приоритеты задаются последовательно. Например, используются числа 0, 1, 2, вместо – 200, 0, 23.

Режим отказа

Заявке никогда не будет отказано в доступе в блок PRIORITY.

Родственные SNA

- PR – Приоритет заявки. Значение приоритета активной заявки.

Вспомогательные окна

- Transaction Snapshot (Текущее состояние заявки) – Изображение состояния заявки в моделировании.
- CEC Snapshot (Текущее состояние CEC) – Изображение состояния цепи текущих событий в моделировании.
- FEC Snapshot (Текущее состояние FEC) – Изображение состояния цепи будущих событий в моделировании.

QUEUE (ОЧЕРЕДЬ)

Блок QUEUE обновляет статистику использования очереди, отражая увеличение ее содержания.

QUEUE A,B

Операнды

A – Номер или имя очереди. Обязательный. В качестве операндов должны быть указаны *Name, PosInteger, ParenthesizedExpression, SNA*, или *SNA*Parameter*.

B – Количество единиц, на которое необходимо увеличит содержимое очереди. Значение по умолчанию 1. Необязательный. В качестве операндов должны быть указаны *Null, Name, PosInteger, ParenthesizedExpression, SNA*, или *SNA*Parameter*.

Пример

QUEUE WaitingLine

В этом примере содержимое очереди *WaitingLine* увеличивается на единицу и обновляется соответствующая статистика.

Алгоритм работы

Когда заявка поступает в блок QUEUE, операнд А высчитывается, округляется и используется для нахождения по номеру нужной очереди. При необходимости, такая очередь создается.

Операнд В определяет значение, на которое необходимо увеличить содержимое очереди. Если операнд В был определен, он высчитывается, округляется и используется в качестве инкремента. Если операнд В определен не был, используется значение по умолчанию 1. В конечном итоге, обновляется относящаяся к очереди статистика.

Особые ограничения

- В, если он определен, должен быть положительным.

Режим отказа

Заявке никогда не будет отказано в доступе в блок QUEUE.

Родственные SNA

- *QEntnum* – Текущее количество требований в очереди. Текущее значение количества требований в очереди *Entnum*.
- *QAEntnum* – Среднее количество требований в очереди. Среднее по времени количество требований в очереди *Entnum*.
- *QCEntnum* – Общее количество требований в очереди. Сумма всех поступивших в очередь требований *Entnum*.
- *QMEntnum* – Максимальное количество требований в очереди. Максимальное (высшая отметка) количество требований в очереди *Entnum*.
- *QTEntnum* – Среднее время ожидания в очереди. Среднее время пребывания в очереди *Entnum*.
- *QXEntnum* – Среднее время ожидания в очереди без нулевых значений. Среднее время пребывания в очереди *Entnum* без учета значений с нулевым временем пребывания.
- *QZEntnum* – Очередь нулевых входных значений. Количество поступивших в очередь заявок *Entnum* с нулевым временем пребывания.

Вспомогательное окно

- Queues Window (Окно очереди) – Просмотр изменения содержимого очереди в реальном времени.

READ (ЧТЕНИЕ)

Блок READ извлекает текстовую строку из потока данных

READ A,B,C

Операнды

А – Параметр заявки. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

В – Номер потока данных. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*. Значение по умолчанию 1.

С – Номер или имя альтернативного блока назначения. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Пример

READ Text_Parm,1,Done

В этом примере блок READ извлекает текстовую строку из потока данных номер 1 и помещает копию строки в параметр заявки *Text_Parm*. Если такого параметра не существует, он создается. Если фиксируется ошибка, активная заявка помещается в блок, помеченный как *Done*, в противном случае она отправляется на следующий блок.

Алгоритм работы

Когда заявка поступает в блок READ, операнд A вычисляется численно, округляется и используется для идентификации параметра заявки, в который получит следующую строку. Если строка получена успешно и ни одного параметра активной заявки не существует, такой создается.

Если используется операнд B, он вычисляется численно, округляется и используется в качестве номера потока данных. Это должно быть положительное вещественное число. Если операнд B не используется, подразумевается поток данных номер 1.

Если используется операнд C, любая ошибка, зафиксированная в процессе READ, вынудит активную заявку направиться в блок с номером C.

В любом случае, если обнаружена ошибка, ее код сохраняется внутри. Блок TERMINATE может использоваться для извлечения кода ошибки. Глава 4 (4.16) содержит полное описание потоков данных, включая описание кодов ошибок.

Если поток данных является канальным потоком, текстовая строка считывается из названия канала и возвращается в параметр заявки. На это время моделирование блокируется.

Если поток данных не является канальным потоком, текстовая строка определяется указателем текущей позиции, первый соответствующий строке индекс ассоциируется с потоком данных. В этом случае, строка, отмеченная указателем текущей позиции, даже если это нулевая строка, возвращается в параметр в качестве результата READ. Затем, указатель текущей позиции инкрементируется, т.е., передвигается к следующему элементу строки. Если строка для чтения отсутствует, не возвращается ничего, и активная заявка посылается в альтернативный блок назначения без сохранения кода какой-либо ошибки.

Считывания могут требовать большого объема вычислений. Производительность можно увеличить посредством использования большого количества информации в небольшом количестве текстовых строк.

Дальнейшее обсуждение

Глава 4 (4.16) содержит полное описание потоков данных в разделе «Потоки Данных».

Режим отказа

Заявке никогда не будет отказано в доступе в блок READ.

Условия блокировки

Моделирование блокируется на время READ текстовой строки.

Родственные блоки

- OPEN – Создает поток данных.
- CLOSE – Закрывает поток данных.
- WRITE – Записывает текстовую строку в потока данных.
- SEEK – Устанавливает указатель текущей позиции потока данных.

RELEASE (ОСВОБОЖДЕНИЕ)

Блок RELEASE отказывает в праве занимать устройства или удаляет заявку из группы претендующих на устройство заявок.

RELEASE A

Операнд

A – Номер устройства. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Пример

RELEASE Teller1

В этом примере когда заявка, занимающая устройство *Teller1*, поступает в блок RELEASE, ей отказывается в праве занимать устройство.

Алгоритм работы

Когда заявка поступает в блок RELEASE, он удаляет ее из группы претендующих на устройство заявок. Это может быть выполнено двумя способом:

Если активная заявка занимает устройство, ей отказывается в этом праве, и она направляется в следующий блок.

Если активная заявка была прервана на устройстве, она удаляется из очереди прерываний устройства. На право занимать устройство воздействие не оказывается, так как другая заявка в данный момент занимает устройство. Если активная заявка освобождается от всех приоритетных прерываний обслуживания, она может без помех передвигаться по системе.

В любом случае, активная заявка теряет право занимать устройство или удаляется из группы претендующих на устройство заявок, и пытается поступить в следующий блок. Если на устройстве нет ни занимающих его, ни прерванных на нем заявок, фиксируется аварийная остановка.

Если активной заявке отказано в праве занимать устройство, следующая поступающая на обработку заявка берется из очереди неоконченной обработки, очереди прерываний и, наконец, очереди прерванной обработки. Если на этом устройстве не было совершено прерываний в режиме прерывания, первой заявке из очереди предоставляется право занять устройство. В противном случае, право занять устройство возвращается последней прерванной заявке. Если и очередь неоконченной обработки (ожидающие режима прерываний заявки), и очередь прерываний (прерванные заявки) пуста, заявка с наивысшим приоритетом из обычной очереди прерванной обработки дается право занять устройство. Если ни одна заявка не находится в режиме ожидания, устройство становится свободным.

Когда новая заявка, которой предоставляется право занять устройство, выбирается из очереди прерванной обработки, она немедленно поступает на вход блоков SEIZE или REEMPT, и затем перераспределяется посредством помещения в СЕС следом за Заявками с равным приоритетом. После этого текущая активная заявка в блоке RELEASE продолжает свое движение.

Особые ограничения

- Поступающая заявка должна занимать устройство или в данный момент быть прерванной на нем. Иначе, фиксируется аварийная остановка.

Режим отказа

Заявке никогда не будет отказано в доступе в блок RELEASE.

Родственные SNA

- *FEntnum* – Занятость устройства. Если устройство *Entnum* в данный момент занято, *FEntnum* возвращает 1. В противном случае, *FEntnum* возвращает 0.
- *FCEntnum* – Количество итераций работы устройства. Сколько раз устройство *Entnum* было занято заявкой.
- *FIEntnum* – Прерывания работы устройства *Entnum*. Если работа устройства *Entnum* в данный момент была прервана, *FIEntnum* возвращает 1. В противном случае, *FIEntnum* возвращает 0.
- *FREntnum* – Использование устройства. Время, в течение которого устройство *Entnum* было занято. *FREntnum* выражается в долях тысячи и, следовательно, возвращает вещественное значение от 0 до 1000, включительно.
- *FTEntnum* – Среднее время занятости (удерживания) устройства. Среднее время, в течение которого устройство *Entnum* было занято заявкой.
- *FVEntnum* – Доступность устройства. *FVEntnum* возвращает 1, если устройство *Entnum* доступно, 0 в противном случае.

Вспомогательные окна

- Facilities Window (Окно устройств) – Возможность просмотра изменения состояния устройств в реальном времени.

REMOVE (УДАЛЕНИЕ)

Блок REMOVE удаляет членов числовой группы или группы заявок.

REMOVE O A,B,C,D,E,F

Операнды

O – Условный оператор. Отношение D к E для активации удаления. Условия выбора описаны ниже. Необязательный. В качестве операндов должны быть указаны *Null*, E, G, GE, L, LE, MAX, MIN или NE.

A – Номер группы. Группа, член которой буде удален. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

B – Предел удаления. Максимальное количество подлежащих удалению заявок. Необязательный. В качестве операндов должны быть указаны ALL, *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

C – Числовое значение. Числовое значение, подлежащее удалению из числовой группы. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

D – Тестовое значение. Номер PR или параметра. Атрибут члена, который определяет, все ли члены группы заявок должны быть удалены или PR для использования приоритета заявки для поределения необходимости удаления. Он вычисляется относительно члена группы заявок. Необязательный. В качестве операндов должны быть указаны PR или *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

E – Контрольное значение. Значение, с которым сравнивается операнд D. Значение адреса вычисляется относительно активной заявки. Необязательный. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

F – Номер блока. Альтернативное место назначения для поступающей заявки. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Примеры

REMOVE Inventory

Это простейший способ использования блока REMOVE. Заявка, поступившая в блок REMOVE, проверяется на принадлежность группе заявок Inventory. Если заявка является членом этой группы, она удаляется из нее.

REMOVE G 3,10,,300,11.6,Jump_Block

В этом примере группа заявок 3 сканируется на предмет наличия в ней заявок со значением параметра 300, большим, чем 11,6. Первые 10 найденных заявок удаляются из группы заявок. Если 10 подходящих заявок найти не удалось, поступившая заявка пытается войти в блок под названием *Jump_Block*. В противном случае, она отправляется в следующий блок.

Алгоритм работы

Блок REMOVE допускает использование числовых значений из числовых групп или заявок из групп заявок. Заявки не изымаются из своего контекста. Однако существует одно исключение. Заявка, поступившая в блок REMOVE, может быть перенаправлена в соответствии со значением операнда F.

Блок REMOVE работает в числовом режиме, если используется операнд C. В числовом режиме операнды A и C численно вычисляются, и число, определенное операндом C, проверяется на принадлежность к числовой группе. Определенной операндом A, который округляются. Если результат из операнда C является членом числовой группы, он удаляется

из нее. Если числовое значение не является членом числовой группы, и используется операнд F, поступившая заявка направляется в блок, определенный операндом F. В противном случае, поступившая заявка поступает в следующий блок. Только операнда A, C и F могут использоваться в числовом режиме. При использовании дробных значений в числовых группах наблюдается некоторая потеря эффективности работы.

Блок REMOVE работает в режиме заявок, если операнд C не используется. В режиме заявок имеются несколько возможностей для выбора подлежащей(их) удалению из группы заявок заявки(ок). Если операнды B, с и E не задаются, удаляется только поступившая в блок REMOVE заявка. Такой режим работы называется «одиночно удаление». В противном случае, выполняется «сканирование группы».

При групповом сканировании можно проверить каждую заявку на удаление посредством использования условных операторов и/или операндов D и E. Также, можно ограничить количество удаляемых заявок при помощи операнда B.

Операнд D всегда направляется в группу заявок после проверки. Стоит заметить, что любые SNA могут быть использованы в операнда D. Любые SNA, которым для вычисления необходима заявка, используют члена текущей группы заявок после прохождения проверки. Результат возвращается любыми SNA, отличными от PR, который используется как номер параметр, чье значение возвращается в качестве итогового результата.

При групповом сканировании можно использовать условные операторы для определения отношения между атрибутом заявки (операнд D) и значением адреса (операнд E), которое будет инициировать удаление заявки. Оба они вычисляются численно. Значение по умолчанию для логического оператора равно E. Если условные операторы не используются, но используются операнды D и E, из значения должно быть равно, для того чтобы заявка была удалена из группы заявок. Если в качестве условных операторов используются MIN или MAX, все заявки с большим или меньшим атрибутом (операнд D) удаляются, пока их количество не достигнет предела удаления (операнд B).

Операнд B прекращает сканирование группы, когда он становится равен количеству удаленных заявок. Значение по умолчанию для него ALL (BCE). Если при неопределенном D атрибутивная проверка не проводится, заявки удаляются до тех пор, пока количество удаленных заявок не станет равно B или пока группа не опустеет.

Операнд F используется и в числовом режиме и в режиме заявок. Он определяет альтернативное место назначения, в которое будет направлена поступившая заявка, когда изъятие будет завершено. Операнд F используется для определения блока назначения при следующих условиях:

- В числовом режиме, если численное значение (операнд C) не являлось членом группы
- При одиночном удалении, если поступившая в блок заявка не являлась членом группы заявок.
- В режиме группового сканирования, если ни одна заявка не была удалена
- В режиме группового сканирования, если количество подлежащих удалению членов, определенное операндом B, не может быть достигнуто.

Если операнд F не используется, поступившая заявка всегда отправляется в следующий блок.

Условные операторы

Условные операторы могут быть следующих видов: E, G, GE, L, LE, MAX, MIN или NE. Если ни один условный оператор не используется, подразумевается E (равенство). Если условие верно, проверяемая Заявка удаляется из Группы. Условия определяются по следующим правилам:

- E – Атрибут заявки, определенный операндом D, должен быть равен соответствующему значению, определенному операндом E, для того чтобы заявка была удалена из группы

- G – Атрибут заявки, определенный операндом D, должен быть больше соответствующего значения, определенного операндом E, для того чтобы заявка была удалена из группы
- GE – Атрибут заявки, определенный операндом D, должен быть больше либо равен соответствующего значения, определенного операндом E, для того чтобы заявка была удалена из группы.
- L – Атрибут заявки, определенный операндом D, должен быть меньше соответствующего значения, определенного операндом E, для того чтобы заявка была удалена из группы
- LE – Атрибут заявки, определенный операндом D, должен быть меньше либо равен соответствующего значения, определенного операндом E, для того чтобы заявка была удалена из группы
- MAX – Атрибут заявки, определенный операндом D, должен быть равен наибольшему атрибуту всех заявок группы, для того чтобы заявка была удалена из группы
- MIN – Атрибут заявки, определенный операндом D, должен быть равен наименьшему атрибуту всех заявок группы, для того чтобы заявка была удалена из группы.
- NE – Атрибут заявки, определенный операндом D, должен быть неравен соответствующему значению, определенному операндом E, для того чтобы заявка была удалена из группы

Если ни один из условных операторов не используется в групповом сканировании, предполагается использование операнда C.

Особые ограничения

- При использовании числовых групп необходимо использовать условные операторы
- При использовании числовых групп необходимо использовать операнды B, D или E.
- При использовании операнда D, необходимо использовать операнд E или, в качестве замены, условные операторы MIN или MAX.
- При использовании операнда E, необходимо использовать операнд D
- При использовании условных операторов MIN или MAX операнд D должен использоваться, а операнд E использовать не должен.

Режим отказа

Заявке никогда не будет отказано в доступе в блок REMOVE.

Родственные блоки

Заявки и числа распределяются по группам при помощи блока JOIN. К заявкам в группе заявок могут обращаться блоки ALTER, EXAMINE, REMOVE, и SCAN. К числам из числовых групп могут обращаться блоки EXAMINE.

Родственные SNAs

- *GNEntnum* – Счетчик количества членов числовой группы. *GNEntnum* возвращает количество членов числовой группы *Entnum*.
- *GTEntnum* – Счетчик количества членов группы заявок. *GTEntnum* возвращает количество членов группы заявок *Entnum*.

Вспомогательные окна

- Transaction Groups Snapshot (Текущее состояние группы заявок) – Изображение состояния группы заявок в моделировании.
- Numeric Groups Snapshot (Текущее состояние числовой группы) – Изображение состояния числовой группы в моделировании.

RETURN (ОСВОБОЖДЕНИЕ)

Блок RETURN отказывает в праве занимать устройства или удаляет заявку из группы претендующих на устройство заявок.

RETURN A

Операнд

A – Номер устройства. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Пример

RETURN Teller1

В этом примере когда заявка, занимающая устройство *Teller1*, поступает в блок RETURN, ей отказывается в праве занимать устройство.

Алгоритм работы

Когда заявка поступает в блок RETURN, он удаляет ее из группы претендующих на устройство заявок. Это может быть выполнено двумя способами:

Если активная заявка занимает устройство, ей отказывается в этом праве, и она направляется в следующий блок.

Если активная заявка была прервана на устройстве, она удаляется из очереди прерываний устройства. На право занимать устройство воздействие не оказывается, так как другая заявка в данный момент занимает устройство. Если активная заявка освобождается от всех приоритетных прерываний обслуживания, она может без помех передвигаться по системе.

В любом случае, активная заявка теряет право занимать устройство или удаляется из группы претендующих на устройство заявок, и пытается поступить в следующий блок. Если на устройстве нет ни занимающих его, ни прерванных на нем заявок, фиксируется аварийная остановка.

Если активной заявке отказано в праве занимать устройство, следующая поступающая на обработку заявка берется из очереди неоконченной обработки, очереди прерываний и, наконец, очереди прерванной обработки. Если на этом устройстве не было совершено прерываний в режиме прерывания, первой заявке из очереди предоставляется право занять устройство. В противном случае, право занять устройство возвращается последней прерванной заявке. Если и очередь неоконченной обработки (ожидающие режима прерываний заявки), и очередь прерываний (прерванные заявки) пуста, заявка с наивысшим приоритетом из обычной очереди прерванной обработки дается право занять устройство. Если ни одна заявка не находится в режиме ожидания, устройство становится свободным.

Когда новая Заявка, которой предоставляется право занять устройство, выбирается из очереди прерванной обработки, она немедленно поступает на вход блоков SEIZE или REEMPT, и затем перераспределяется посредством помещения в СЕС следом за заявками с равным приоритетом. После этого текущая активная заявка в блоке RETURN продолжает свое движение.

Особые ограничения

- Поступающая заявка должна занимать устройство или в данный момент быть прерванной на нем. Иначе, фиксируется аварийная остановка.

Режим отказа

Заявке никогда не будет отказано в доступе в блок RETURN.

Родственные SNA

- *FEntnum* – Занятость устройства. Если устройство *Entnum* в данный момент занято, *FEntnum* возвращает 1. В противном случае, *FEntnum* возвращает 0.
- *FCEntnum* – Количество итераций работы устройства. Сколько раз устройство *Entnum* было занято заявкой.

- *FIEntnum* – Прерывания работы устройства *Entnum*. Если работа устройства *Entnum* в данный момент была прервана, *FIEntnum* возвращает 1. В противном случае, *FIEntnum* возвращает 0.
- *FREntnum* – Использование устройства. Время, в течение которого устройство *Entnum* было занято. *FREntnum* выражается в долях тысячи и, следовательно, возвращает вещественное значение от 0 до 1000, включительно.
- *FTEntnum* – Среднее время занятости (удерживания) устройства. Среднее время, в течение которого устройство *Entnum* было занято заявкой.
- *FVEntnum* – Доступность устройства. *FVEntnum* возвращает 1, если устройство *Entnum* доступно, 0 в противном случае.

Вспомогательные окна

- Facilities Window (Окно устройств) – Возможность просмотра изменения состояния устройств в реальном времени.

SAVAIL (ПРОВЕРКА ДОСТУПНОСТИ ПАМЯТИ)

Блок SAVAIL определяет, находится ли память в доступном состоянии.

SAVAIL A

Операнды

A – Имя или номер памяти. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Пример

SAVAIL MotorPool

В этом примере, когда заявка поступает на вход, блока SAVAIL, память *MotorPool* проверяется на доступность.

Алгоритм работы

Блок SAVAIL определяет, находится ли память в доступном состоянии. Если заявки ожидают в очереди задержки к памяти, они получают шанс удовлетворить свой запрос к памяти, в соответствии с дисциплиной естественной очередности «first-fit-with-skip». Заявки, чьи запросы к памяти не могут быть удовлетворены, остаются в очереди задержки к памяти.

Если память уже находится в доступном состоянии, блок SAVAIL не выполняет никаких действий.

Режим отказа

Заявке никогда не будет отказано в доступе в блок SAVAIL.

Родственные команды

Область памяти должна быть определена командами STORAGE до того, как она будет обновлена в блоке SAVAIL. Команды STORAGE должны быть определены в модели или должны быть переданы имитационному объекту интерактивно, до того как заявка сможет войти в блок SAVAIL. Любая попытка сделать это до того, как память будет определена, приведет к аварийной остановке.

Область памяти может быть переопределена при помощи интерактивных команд STORAGE.

Родственные SNAs

- *REntnum* – Неиспользуемая емкость памяти. Объем памяти (или "маркер" пустоты) доступный для использования поступающей заявкой или содержимым самой памяти *Entnum*.
- *SEntnum* – Занятая память. *SEntnum* возвращает объем содержащейся в памяти информации (или "маркер" пустоты) использующийся в данный момент активной заявкой или содержимым самой памяти *Entnum*.
- *SAEntnum* – Средняя емкость используемой памяти. *SAEntnum* возвращает среднюю по времени емкость используемой памяти (или "маркер" пустоты) *Entnum*.

- *SCEntnum* – Объем используемой памяти. Общее количество единиц (ячеек) памяти, которые были заняты (или "маркер" пустоты) *Entnum*.
- *SEEntnum* – Память пуста. *SEEntnum* возвращает 1, если память *Entnum* не используется, 0 в противном случае.
- *SFEntnum* – Память заполнена. *SFEntnum* возвращает 1, если память *Entnum* заполнена, 0 в противном случае.
- *SREntnum* – Использование памяти. Часть общего использования, представленная в виде средней занятости памяти *Entnum*. *SREntnu*, выражается в долях тысячи и, следовательно, возвращает вещественное значение от 0 до 1000, включительно.
- *SMEntnum* – Максимальный объем используемой памяти *Entnum*. "Высшая планка".
- *STEntnum* – Среднее время хранения единицы памяти *Entnum*.
- *SVEntnum* – Состояние памяти. *SVEntnum* возвращает 1, если память *Entnum* доступна, 0 в противном случае.

Вспомогательные окна

- Storages Window (Окно памяти) – Просмотр изменения содержимого памяти в реальном времени.

SAVEVALUE (СОХРАНЕНИЕ)

Блок SAVEVALUE изменяет значения области сохранения значений

SAVEVALUE A,B

Операнды

A – номер области сохранения значений. Обязательный. Может сопровождаться суффиксами + или – для определения увеличения или уменьшения имеющегося значения. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter* с суффиксом +, –, или *Null*.

B – Хранимое, прибавляемое или вычитаемое значение. Обязательный. В качестве операндов должны быть указаны *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Примеры

SAVEVALUE Account,99.95

В этом приме область сохранения значений *Account* сохраняет значение 99,95.

SAVEVALUE The_Bard,"A rose by any other name ..."

В этом примере область сохранения значений *The_Bard* сохраняет строку. Если такая область сохранения значений не существует, она создается.

Алгоритм работы

Блок SAVEVALUE используется для сохранения, увеличения или уменьшения значения из области сохранения значений.

Операнд A вычисляется численно, округляется и используется в качестве номера области сохранения значений.

Операнд B вычисляется и используется для определения нового значения для области сохранения значений. Если операнд A имеет суффикс +, численный эквивалент операнда B прибавляется к численному эквиваленту прежнего значения. Если A имеет суффикс –, численный эквивалент операнда B вычитается из численного эквивалента прежнего значения. Если за операндом A не следует никакой знак, прежнее значение, содержащееся в области сохранения значений, заменяется на новое, содержащееся в операнда B.

Режим отказа

Заявке никогда не будет отказано в доступе в блок SAVEVALUE.

Related SNA

- *XEntnum* – область сохранения значений. Возвращается значение, содержащееся в области сохранения значений *Entnum*.

Вспомогательные окна

- Expressions Window (Окно выражений) – Просмотр значения выражений в реальном времени.
- Plot Window (Окно сценариев) – Просмотр содержимого более 8 выражений в реальном времени.
- Savevalues Window (Окно области сохранения значений) – Просмотр изменения содержимого области сохранения значений в реальном времени.

SCAN (СКАНИРОВАНИЕ)

Блок SCAN передает информацию от члена группы заявок активной заявке.

SCAN O A,B,C,D,E,F

Операнды

O – Условный оператор. Отношение B и C для выбора члена группы заявок. Этот выбор обуславливается ниже. Необязательный. В качестве операндов должны быть указаны *Null*, *E*, *G*, *GE*, *L*, *LE*, *MAX*, *MIN*, или *NE*.

A – Номер группы заявок. Группа, члены которой будут просканированы. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

B – Тестовое значение. Номер PR или параметра. Атрибут члена, который определяет должен ли быть выбран член группы заявок. Высчитывается относительно члена группы заявок. Необязательный. В качестве операндов должны быть указаны PR или *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

C – Контрольное значение. Значение, с которым сравнивается операнд B. По умолчанию 0. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

D – Возвращаемое значение. Номер PR или параметра. Атрибут члена, который определяет, какие члены должны быть установлен в качестве Параметра Активной Заявки. Обязательный. В качестве операндов должны быть указаны PR, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

E – Номер полученного параметра. Номер параметра поступившей заявки, который получит значение, возвращенное Операндом D. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*.

F – Номер альтернативного блока. Альтернативное место назначения для поступающей заявки. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Примеры

SCAN MAX Inventory,P\$Price,,P\$PartNumber,100

В этом примере все заявки из группы заявок *Inventory* проверяются на наличие среди них заявки с наибольшим параметром *Price*. Первая заявка с максимальный параметром *Price* выбирается определения значения операнда D. Значение параметра *PartNumber* выбранной группы заявок назначается в качестве параметра номер 100 активной заявки. Если группа заявок пуста, ничего не происходит.

SCAN E Lot11,PartNum,127,Price,Price,Phone

В этом примере, когда заявка поступает в блока SCAN, группа заявок *Lot11* сканируется на определение первой заявки, в параметре *PartNum* которой содержится значение 127. Если такая заявка находится, значение ее параметра *Price* перемещается в соответствующий параметр поступившей в блок SCAN заявки. Поступившая заявка затем направляется в следующий блок. Если ни одного подходящего члена группы не находится, поступившая заявка направляется в блок, обозначенный как *Phone*. Операнды D и E не обязательно должны определять одинаковые параметры заявки.

Алгоритм работы

Блок SCAN находит первую заявку в группе, удовлетворяющую поставленному условию, и сохраняет один из ее атрибутов в параметре активной заявки. Если такого параметра не существует, он создается. Если подходящий член группы не может быть найден, ничего не сохраняется.

Член группы заявок может быть выбран на основе проверки одного из его атрибутов. Это делается при помощи операндов В,С и/или условных операторов. Если ни один из этих способов проверки не используется, выбирается первая заявка в группе, если таковая существует.

Когда находится удовлетворяющий условию член группы заявок, его атрибут, определенный операндом D, копируется в параметр активной заявки, определенный операндом E. В этом случае, активная заявка всегда перемещается в следующий блок.

Если ни одной подходящей под условие заявки не найдено, активная заявка может быть переадресована в блок, определенный операндом F. В противном случае (операнд F не задан), она переходит в следующий блок.

Если используются операнды В, С или условный оператор, выбирается первый член группы заявок, удовлетворивший условию проверки. Он может быть сравнен с минимальным или максимальным атрибутом членов всей группы посредством использования в качестве условного оператора операторов MIN или MAX. Выбирается первая заявка из группы заявок, которая имеет максимальный или атрибут. При использовании операторов MIN или MAX запрещено использование оператора С.

Можно сравнить атрибуты членов группы с операндом С, с помощью условных операторов или без них. В этом случае должны использоваться любые условные операторы, кроме операторов MIN и MAX. Значение по умолчанию для операнда С равно 0.

Операнда В и D всегда направляются к членам группы заявок после проверки. Стоит заметить, что любые SNA могут применяться. Любые SNA, требующие определения заявки, используют для проверки текущую группу заявки. Результат, возвращаемый любыми SNA, отличными от PR, используется в качестве номера параметра, чье значение будет возвращено в качестве окончательного результата.

По умолчанию используется условный оператор равенства E. При использовании условных операторов совместно с операндами В и С, значение должно совпасть со значением атрибута члена группы заявок, для того этот член был выбран.

Операнд F определяет альтернативный блок назначения для поступившей заявки, если ни одна заявка из Группы не удовлетворила условию проверки. Если F не используется, поступившая заявка всегда направляется в следующий блок.

Условные операторы

Условные операторы могут быть следующих видов: E, G, GE, L, LE, MAX, MIN или NE. Если ни один условный оператор не используется, подразумевается E (равенство). Если условие верно, проверяемая заявка выбирается из группы. Условия определяются по следующим правилам:

- E – Атрибут заявки, определенный операндом В, должен быть равен соответствующему значению, определенному операндом С, для того чтобы заявка была удалена из группы
- G – Атрибут заявки, определенный операндом В, должен быть больше соответствующего значения, определенного операндом С, для того чтобы заявка была удалена из группы
- GE – Атрибут заявки, определенный операндом В, должен быть больше либо равен соответствующего значения, определенного операндом С, для того чтобы заявка была удалена из группы.

- L – Атрибут заявки, определенный операндом, должен быть меньше соответствующего значения, определенного операндом E, для того чтобы заявка была удалена из группы
- LE – Атрибут заявки, определенный операндом B, должен быть меньше либо равен соответствующего значения, определенного операндом C, для того чтобы заявка была удалена из группы
- MAX – Атрибут заявки, определенный операндом B, должен быть равен наибольшему атрибуту всех заявок группы, для того чтобы заявка была удалена из группы
- MIN – Атрибут заявки, определенный операндом B, должен быть равен наименьшему атрибуту всех заявок группы, для того чтобы заявка была удалена из группы.
- NE – Атрибут заявки, определенный операндом B, должен быть неравен соответствующему значению, определенному операндом C, для того чтобы заявка была удалена из группы

Особые ограничения

- При использовании операнда C, необходимо использовать операнд B
- При использовании условных операторов MIN или MAX операнд C должен использоваться, а операнд E использовать не должен.

Режим отказа

Заявке никогда не будет отказано в доступе в блок SCAN.

Родственные блоки

Заявки и числа распределяются по группам при помощи блока JOIN. К заявкам в группе заявок могут обращаться блоки ALTER, EXAMINE, REMOVE и SCAN.

Родственные SNA

- *GTEntnum* – Счетчик количества членов группы заявок. *GTEntnum* возвращает количество членов группы заявок *Entnum*.

Вспомогательные окна

- Transaction Groups Snapshot (Текущее состояние группы заявок) – Изображение состояния группы заявок в моделировании.
- Numeric Groups Snapshot (Текущее состояние числовой группы) – Изображение состояния числовой группы в моделировании.

SEEK (ПОИСК)

Блок SEEK устанавливает указатель текущей позиции потока данных

SEEK A,B

Операнды

A – Новая позиция указателя. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

B – Номер потока. данных. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Пример

SEEK 20,Data_Base

В этом примере блок SEEK изменяет положение указателя текущей позиции потока Данных *Data_Base* на 20.

Алгоритм работы

Поток данных – это последовательность текстовых строк, используемый в среде моделирования GPSS. Каждый поток данных идентифицируется уникальным номером. Каждый поток данных задается уникальным номером. Глава 4 (4.16) содержит полное описание потоков данных, включая описание кодов ошибок, в разделе «Потоки Данных».

Каждый поток данных имеет *указатель текущей позиции*. Это однозначный индекс следующей позиции строки, которая будет считана или в которую будет произведена запись. Когда заявка поступает в блок SEEK, операнд A вычисляется численно и используется в качестве следующего указателя текущей позиции.

Если операнд B используется, он численно вычисляется и используется в качестве номера потока данных. Если операнд B не используется, поиск производится в потоке данных номер 1.

Если фиксируется ошибка, ее код сохраняется внутри. Блок CLOSE может использоваться для извлечения кода ошибки.

Режим отказа

Заявке никогда не будет отказано в доступе в блок SEEK.

Родственные блоки

- OPEN – Создает поток данных.
- CLOSE – Закрывает поток данных.
- READ – Считывает текстовую строку из потока данных.
- WRITE – Записывает текстовую строку в потока данных.

SEIZE (ЗАХВАТ)

Когда активная заявка делает попытку войти в блок SEIZE, она запрашивает разрешение или ждет возможности занять устройство.

SEIZE A

Операнд

A – Имя или номер устройства. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Пример

SEIZE Teller1

В этом примере, когда заявка делает попытку войти в блок SEIZE, состояние устройства *Teller1* проверяется. Если оно свободно, активной заявке, позволяется войти в блок SEIZE и занять устройство. Затем активная заявка поступает в следующий блок (Next Sequential Block – NSB). Если устройство занято, активная заявка остается ожидать в очереди задержки устройства.

Алгоритм работы

Блок SEIZE позволяет заявке занять устройство. Если устройство свободно, заявка немедленно поступает на него, проходит блок SEIZE и делает попытку войти в следующий блок. Если устройство уже занято, заявка встает в очереди задержки устройства последней среди равных ей по приоритету и не проходит в блок SEIZE.

Режим отказа

Заявке отказывается в поступлении в блок SEIZE, если она не может немедленно занять устройство.

активной заявке отказывается в поступлении в блок SEIZE, если устройство находится в недоступном состоянии.

Когда заявке отказано в доступе, ее индикатор задержки включается и накапливается до тех пор, пока заявка не попадет в «синхронный» режим блока TRANSFER. Синхронный режим блока передачи используется редко. Логические переменные способны более эффективно контролировать координацию состояния определенного числа ресурсов при использовании блока TEST.

Родственные SNAs

Связанные с устройствами SNAs:

- *FEntnum* – Занятость устройства. Если устройство *Entnum* в данный момент занято, *FEntnum* возвращает 1. В противном случае, *FEntnum* возвращает 0.
- *FCEntnum* – Количество итераций работы устройства. Сколько раз устройство *Entnum* было занято заявкой.

- *FIEntnum* – Прерывания работы устройства *Entnum*. Если работа устройства *Entnum* в данный момент была прервана, *FIEntnum* возвращает 1. В противном случае, *FIEntnum* возвращает 0.
- *FREntnum* – Использование устройства. Время, в течение которого устройство *Entnum* было занято. *FREntnum* выражается в долях тысячи и, следовательно, возвращает вещественное значение от 0 до 1000, включительно.
- *FTEntnum* – Среднее время занятости (удерживания) устройства. Среднее время, в течение которого устройство *Entnum* было занято заявкой.
- *FVEntnum* – Доступность устройства. *FVEntnum* возвращает 1, если устройство *Entnum* доступно, 0 в противном случае.

Вспомогательные окна

- Facilities Window (Окно устройств) – Возможность просмотра изменения состояния устройств в реальном времени.

SELECT (ВЫБОР)

Блок SELECT выбирает объект и помещает его номер в параметр активной заявки.

SELECT O A,B,C,D,E,F

Операнды

O – Условный или логический оператор. Выбор обуславливается ниже. Обязательный. В качестве операндов должны быть указаны FNV, FV, I, LS, LR, NI, NU, SE, SF, SNE, SNF, SNV, SV, U, E, G, GE, L, LE, MIN, MAX, или NE.

A – Номер или имя параметра для занесения в него номера выбранного объекта. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

B – Нижняя граница диапазона объектов. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

C – Верхняя граница диапазона объектов. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

D – Контрольное значение для операнда E при работе в условном режиме. Необязательный. В качестве операндов должны быть указаны *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*. Не используется с MAX и MIN.

E – Имя класса SNA. Атрибут объекта, предназначенный для проверки в условном режиме. Обязателен только для условного режима. Тип SNA подразумевает тип объекта. Номер объекта не определяется в операнде E. Это производится автоматически после нахождения уровня номера. Разрешается использовать любой класс SNA. Необязательный. В качестве операндов должны быть указаны *Null* или *entitySNAclass*.

F – Номер альтернативного блока. Блок назначения, если ни один объект не выбран. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Примеры

SELECT SNF Not_Full,Bin1,Bin3

В этом примере номер объекта первой памяти, в которой содержатся объекты с номерами между *Bin1* и *Bin3*, будет сохранен, а параметре заявки *Not_Full*. Если такой параметр не существует, он будет создан. Благоразумно каждый раз перед поступлением в блок SELECT проверять, что удачный выбор возможен. Если он невозможен, в параметр будет помещен 0, и вход в блок ENTER вызовет аварийную остановку по причине того, что ни одна поступающая заявка не должна иметь нулевой номер поступления. Также можно использовать альтернативный выход, если ни один объект не подошел под ожидаемый критерий.

Если уровни искомых объектов определены при помощи буквенно-цифровых имен по возрастанию, необходимо сначала использовать EQU выражения для соотнесения

номеров категорий с именами диапазонов. EQU выражения должны вызываться в соответствии оригинальными значениями объектов.

10 Bin1 EQU 1

20 Bin2 EQU 2

30 Bin3 EQU 3

40 Bin1 STORAGE 11

50 Bin2 STORAGE 1000

60 Bin3 STORAGE 150

.

100 SELECT SNF 3,Bin1,Bin3,,,No_Room

Выше приведен пример передачи альтернативного места назначения заявкам, которые обнаружили, что все объекты памяти заполнены. Если сразу не проверить успех выбора при помощи блока TEST или булевых переменных, придется определять альтернативное место назначения для варианта неудачного выбора.

SELECT E Empty1,Queue1,Queue9,0,Q

В этом примере блок SELECT работает в условном режиме. Каждый объект очереди с номером между *Queue1* и *Queue9*, включительно, проверяется. Первая найденная очередь, чье содержимое в данный момент равно 0, выбирается. *EMPTY1* – название параметра поступающей заявки, получившее номер первой «пустой» очереди из определенного диапазона.

Алгоритм работы

Когда происходит поступление в блок SELECT, происходит проверка объекта, определенного операндом В. Если объект не существует, и не требует отдельной команды для своего определения, новый объект создается. После этого каждый объект в диапазоне, определенном операндами В и С, подвергается проверке. SNA строятся автоматически для каждого объекта. Класс SNA, используемый для построения SNA, берется из операнда Е или определяется логическим оператором.

Блок SELECT работает в логическом или условном режиме, в зависимости от того, используются логические или условные операторы.

Когда используются (описанные ниже) логические операторы, используются операнды А, В и С. Условие, определенное логическим оператором, проверяется для объектов, чьи номера находятся в интервале между В и С. Номер первого объекта, найденного в соответствии с заданным условием, помещается в параметр поступившей заявки, номер которого определен операндом А. Тип объекта предполагается логическим оператором. Если ни одного объекта найдено не было, в параметр активной заявки помещается 0. Если параметр не существует, он создается.

При использовании условных операторов, используются операнды А, В, С, Е, и, обычно, D. Операнды А, В, С используются для определения целевого параметра и диапазона номеров объектов, как и прежде. Но теперь условные операторы определяют взаимоотношения между операндами D и Е, которые, в которых должны содержаться выбранные объекты. Оба вычисляются численно.

В условном режиме класс SNA группируется со спецификатором объекта для построения SNA. Типы объектов для каждого класса SNA приведены разделе 3.4. Полные SNA строятся из этих классов и некоторого количества проверяемых объектов. Каждый такой SNA устанавливается для каждого объекта и сравнивается с контрольным значением операнда D. Операнд D устанавливается для поступившей заявки и является контрольным значением для сравнения с операндом Е, который определяется классом SNA (а, следовательно, и типом объекта) для оценивания. Условный оператор определяет отношение, которое операнд Е, вычисленный для каждого объекта, должен соотнести с операндом D, вычисленный в отношении поступившей заявки, для того чтобы объект был

выбран. При использовании в качестве условного оператора операторов MIN или MAX, операнд D игнорируется.

В обоих режимах, операнд F может использоваться для определения нового блока назначения для поступившей заявки, если ни один объект не был выбран. Если F не используется, поступившая заявка всегда направляется в следующий блок. Если F используется и ни один объект не был выбран, активная заявка перемещается в блок, определенный операндом F с параметром заявки, установленным операндом A в 0.

Логические операторы

И логические, и условные операторы являются обязательными. Логические операторы могут быть следующих видов: FNV, FV, I, LS, LR, NI, NU, SE, SF, SNE, SNF, SNV, SV или U. Когда значение логического оператора истинно, объект выбирается. Условия описаны ниже:

- FNV – Устройство, однозначно определенное операндом A, должно быть недоступно для того чтобы быть выбранным.
- FV – Устройство, однозначно определенное операндом A, должно быть доступно для того чтобы быть выбранным.
- I – Работа устройства, однозначно определенного операндом A, должна быть в данный момент прервана для того чтобы быть выбранным.
- LS – Логический переключатель, однозначно определенный операндом A, должен быть в состоянии «включено» для того чтобы быть выбранным
- LR – Логический переключатель, однозначно определенный операндом A, должен быть в состоянии «выключено» для того чтобы быть выбранным
- NI – Работа устройства, однозначно определенного операндом A, НЕ должна быть прервана для того чтобы быть выбранным.
- NU – Устройство, однозначно определенное операндом A, не должно использоваться для того чтобы быть выбранным.
- SE – Память, однозначно определенная операндом A, должна быть пуста для того чтобы быть выбранным. Ни одна ячейка памяти не должна быть занята.
- SF – Память, однозначно определенная операндом A, должна быть заполнена для того чтобы быть выбранным. Все ячейки памяти должны быть заняты.
- SNE – Память, однозначно определенная операндом A, должна быть не пуста для того чтобы быть выбранным. Хотя бы одна ячейка памяти должна быть занята.
- SNF – Память, однозначно определенная операндом A, должна быть не заполнена для того чтобы быть выбранным. Хотя бы одна ячейка памяти должна быть свободна.
- SNV – Память, однозначно определенная операндом A, должна быть в состоянии «недоступна» для того чтобы быть выбранным.
- SV – Память, однозначно определенная операндом A, должна быть в состоянии «доступна» для того чтобы быть выбранным.
- U – Устройство, однозначно определенное операндом A, должно использоваться для того чтобы быть выбранным.

Условные операторы

И условные, и логические операторы являются обязательными. Условные операторы могут быть следующих видов: E, G, GE, L, LE, MAX, MIN, or NE. Условия описаны ниже:

- E – Значение автоматического SNA, должно быть равно соответствующему значению, определенному операндом D, для того чтобы объект был выбран
- G – Значение автоматического SNA, должно быть больше соответствующего значения, определенному операндом D, для того чтобы объект был выбран
- GE – Значение автоматического SNA, должно быть больше либо равно соответствующего значения, определенному операндом D, для того чтобы объект был выбран

- L – Значение автоматического SNA, должно быть меньше соответствующего значения, определенному операндом D, для того чтобы объект был выбран
- LE – Значение автоматического SNA, должно быть меньше либо соответствующего значения, определенному операндом D, для того чтобы объект был выбран
- MAX – Значение автоматического SNA, должно быть равно наибольшему из всех значений таких SNA, для того чтобы объект был выбран.
- MIN – Значение автоматического SNA, должно быть равно наименьшему из всех значений таких SNA, для того чтобы объект был выбран.
- NE – Значение автоматического SNA, должно быть неравно соответствующему значению, определенному операндом D, для того чтобы объект был выбран

Особые ограничения

- И условные, и логические операторы являются обязательными.
- Вычисленные номера объектов должны быть положительными вещественными числами.
- D и E обязательны, если O условный оператор. Отличный от MIN и MAX.
- В процессе вычисления, C должен быть больше либо равен B.

Режим отказа

Заявке никогда не будет отказано в доступе в блок SELECT

Вспомогательные окна

- Facilities Window (Окно устройств) – Возможность просмотра изменения состояния устройств в реальном времени.
- Logicswitches Window (Окно логических переключений) – Просмотр действия логических схем в реальном времени.
- Storages Window (Окно памяти) – Просмотр изменения содержимого памяти в реальном времени.
- Transaction Snapshot (Текущее состояние заявки) – Изображение состояния заявки в моделировании.

SPLIT (ДРОБЛЕНИЕ)

Блок SPLIT создает заявку из того же набора, что и активная заявка.

SPLIT A,B,C

Операнды

A – Счетчик. Количество созданных родственных заявок. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

B – Номер блока. Место назначения для новой заявки. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

C – Номер параметра. Параметр для получения серийного номера. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Примеры

SPLIT 1

Это простейших способ использования блока SPLIT. Создается новая копия родительской Заявки, которая последует за родительской в следующий блок. Новая заявка имеет такой же приоритет, значение параметра и время маркировки, что и родительская.

SPLIT 3,P20,20

В этом примере каждый раз, когда происходит поступление в блок SPLIT, создаются 3 новые заявки. Каждая заявка имеет такой же приоритет, значение параметра и время маркировки, что и родительская, за исключением параметра номер 20.

Каждый потомок заявки будет иметь серийный номер в своем параметре номер 20, основанный на значении параметра номер 20 родительской заявки. Значение родительского параметра 20 сначала уменьшается на 1 (в случае, если он равен 0), а затем последовательно вычисляются серийные номера для потомков.

Блок назначения, определенный в операнда В, вычисляется для каждой вновь созданной заявки. Если 20-й параметр родительской заявки содержит номер блока, скажем, n , первый потомок отправится в блок $n+2$, второй – в $n+3$, третий в $n+4$.

Родительская заявка перейдет в следующий блок, идущий за блоком SPLIT.

Приведенный выше пример уникален, потому как параметр 20 используется и операндом С, и операндом В. Таким образом, серийный номер используется для определения Блока назначения.

Операнды В и С могут использоваться для различных целей, В – для содержания однозначного определения места назначения заявок–потомков, а С в качестве серийного номера, который может быть использован для индивидуального направления Заявок в некоторое место назначения при дальнейшем моделировании.

SPLIT 3,Pro,17

В этом примере каждый раз, когда происходит поступление в блок SPLIT, создаются 3 новые заявки. Родительская заявка переходит в следующий блок, потомки – в блок отмеченный как Pro. Параметр 17 получает сериализацию. Если параметр родительской заявки не предопределен, он будет создан и инициализирован в 0. В этом примере родительская заявка с нулевым значением параметра 17 (параметр, используемый для сериализации) будет, после прохождения блока SPLIT, иметь в 17-м параметре единицу, а в 17-м параметр потомков будут записаны 2, 3 и 4.

Алгоритм работы

Блок SPLIT создает новые заявки, использующие атрибуты родительских. Каждая заявка–потомок имеет такой же приоритет, значение параметра и время маркировки, что и родительская и находится в том же наборе. Если указатель пути установлен в родительской заявке, он будет включен и в заявке–потомке.

Новые заявки могут быть отправлены в альтернативное место назначения, посредством использования операнда В. Номер нового блока вычисляется для каждой новой заявки.

Необязательные операнд С определяет номер параметра вновь созданной заявки для хранения серийного номера. Нумеровка начинается со значения соответствующего параметра родительской заявки, плюс 1. Например, если необходимо создать 3 копии с серийными номерами в параметре 120, и поступающая заявка содержит в 120-м параметре значение 15, то новые заявки будут иметь значений 16, 17 и 18 в своих 120-х параметрах, соответственно. Если у родительской заявки не было такого параметра, он создается и нумерация начинается с 1.

Используя операнды В и С можно отправлять каждую новую заявку в различные места назначения, как это показано выше, во втором примере.

Родительская заявка и все заявки–потомки принадлежат к одному и тому же набору, называемом набором заявок (Assembly Set). Все заявки, принадлежащие к одному определенному набору, могут быть изменены посредством поступления в блок ADOPT. Членов одного набора называют «родственными». Родство заявок может быть установлено и использовано в блоках ASSEMBLE, GATHER или MATCH и GATE блоке для синхронизации и других целей.

Особые ограничения

- А, В, и С, если они определены, должны быть положительными.

Режим отказа

Заявке никогда не будет отказано в доступе в блок SPLIT.

Родственные блоки

- ADOPT – Устанавливает наборы (Assembly Set) активных заявок
- ASSEMBLE – Ожидает и ликвидирует членов наборов.

- GATHER – Ожидает членов набора.
- MATCH – Дождется члена набора.

Родственные SNAs

- A1 – Набор. Возвращает набор активной заявки.
- MBEntnum – Совпадения в блоке. MBEntnum возвращает 1 если в блоке Entnum есть заявка из того же набора, что и активная заявка. В противном случае, MBEntnum возвращает 0.

Вспомогательные окна

- Blocks Window (Окно блоков) – Просмотр смены блоков в реальном времени.
- Transaction Snapshot (Текущее состояние заявки) – Изображение состояния заявки в моделировании.
- CEC Snapshot (Текущее состояние CEC) – Изображение состояния цепи текущих событий в моделировании.
- FEC Snapshot (Текущее состояние FEC) – Изображение состояния цепи будущих событий в моделировании.

SUNAVAIL (ПРОВЕРКА НЕДОСТУПНОСТИ ПАМЯТИ)

Блок SUNAVAIL определяет, находится ли память в недоступном состоянии.

SUNAVAIL A

Операнд

A – Имя или номер памяти. Обязательный. В качестве операндов должны быть указаны Name, PosInteger, ParenthesizedExpression, SNA, или SNA*Parameter.

Пример

SUNAVAIL MotorPool

В этом примере память MotorPool устанавливается в недоступное состояние, когда заявка поступает на вход блока SUNAVAIL.

Алгоритм работы

Блока SUNAVAIL определяет, находится ли память в недоступном состоянии. Это означает, что обращающиеся к памяти заявки помещаются в очередь задержки к памяти. Ни одной заявке не будет позволено войти в блок ENTER, если память находится в недоступном состоянии. Если память уже находилась в недоступном состоянии, блок SUNAVAIL не производит никаких действий.

Особые ограничения

- В операнде A должно содержаться имя или номер существующей памяти

Режим отказа

Заявке никогда не будет отказано в доступе в блок SUNAVAIL

Родственные Команды

Область памяти должна быть определена командами STORAGE до того, как она будет обновлена в блоке SUNAVAIL. Команды STORAGE должны быть определены в модели или должны быть переданы имитационному объекту интерактивно, до того как заявка сможет войти в блок SUNAVAIL. Любая попытка сделать это до того, как память будет определена, приведет к аварийной остановке.

Область памяти может быть переопределена при помощи интерактивных команд STORAGE.

Родственные SNAs

- REntnum – Неиспользуемая емкость памяти. Объем памяти (или "маркер" пустоты) доступный для использования поступающей заявкой или содержимым самой памяти Entnum.

- *SEntnum* – Занятая память. *SEntnum* возвращает объем содержащейся в памяти информации (или "маркер" пустоты) использующийся в данный момент активной заявкой или содержимым самой памяти *Entnum*.
- *SAEntnum* – Средняя емкость используемой памяти. *SAEntnum* возвращает среднюю по времени емкость используемой памяти (или "маркер" пустоты) *Entnum*.
- *SCEntnum* – Объем используемой памяти. Общее количество единиц (ячеек) памяти, которые были заняты (или "маркер" пустоты) *Entnum*.
- *SEEntnum* – Память пуста. *SEEntnum* возвращает 1, если память *Entnum* не используется, 0 в противном случае.
- *SFEntnum* – Память заполнена. *SFEntnum* возвращает 1, если память *Entnum* заполнена, 0 в противном случае.
- *SREntnum* – Использование памяти. Часть общего использования, представленная в виде средней занятости памяти *Entnum*. *SREntnu*, выражается в долях тысячи и, следовательно, возвращает вещественное значение от 0 до 1000, включительно.
- *SMEntnum* – Максимальный объем используемой памяти *Entnum*. "Высшая планка".
- *STEntnum* – Среднее время хранения единицы памяти *Entnum*.
- *SVEntnum* – Состояние памяти. *SVEntnum* возвращает 1, если память *Entnum* доступна, 0 в противном случае.

Вспомогательные окна

- Storages Window (Окно памяти) – Просмотр изменения содержимого памяти в реальном времени.

TABULATE (ТАБУЛЯЦИЯ)

Блок TABULATE собирает набор значений в таблицу.

TABULATE A,B

Операнды

A – Номер или имя таблицы. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

B – Весовой коэффициент. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *Number*, *ParenthesizedExpression*, *SNA* или *SNA*Parameter*.

Пример

TABULATE Sales

Когда заявка поступает в блок TABULATE, таблица *Sales* находится. *Sales* должна быть определена командой TABLE. Затем связанная с таблицей статистика обновляется без учета весовых коэффициентов.

Алгоритм работы

Когда заявка поступает в блок TABULATE, операнд A вычисляется и используется для нахождения таблицы. Если такой таблицы не существует, фиксируется аварийная остановка.

Таблица обновляется согласно заданным в табличных выражениях (TABLE Statement) операндам. Если операнд B задан, он вычисляется и используется в качестве весового коэффициента. В противном случае, весовой коэффициент принимается равным 1.

Последующее обсуждение сбора табличной статистики можно найти в разделе 4.10 и главе 12.

Особые ограничения

- A должно быть положительным.
- B, если оно определено, должно быть положительным.
- A должно задавать имя или номер предопределенной таблицы.

Режим отказа

Заявке никогда не будет отказано в доступе в блок TABULATE.

Родственные команды

Таблица должна быть определена командами TABLE до того, как она будет обновлена в блоке TABULATE. Команды TABLE должны быть определены в модели или должны быть переданы имитационному объекту интерактивно, до того как заявка сможет войти в блок TABULATE. Любая попытка сделать это до того, как таблица будет определена, приведет к аварийной остановке.

Таблица может быть переопределена при помощи интерактивных команд TABLE

Родственные SNAs

- *TBEntnum* – Нет весового коэффициента для таблицы *Entnum*.
- *TCEntnum* – Счетчик количество входов значений без весового коэффициента в таблицу *Entnum*.
- *TDEntnum* – Среднеквадратическое отклонение входов значений без весового коэффициента в таблицу *Entnum*.

Вспомогательные окна

- Table Window (Окно таблицы) – Просмотр изменения содержимого таблицы в реальном времени

TERMINATE (ЗАВЕРШЕНИЕ)

Блок TERMINATE удаляет активную заявку из процесса моделирования и, по возможности, уменьшает счетчик завершения (Termination Count)

TERMINATE A

Операнд

A – Декремент счетчика завершения. По умолчанию равен 0. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Пример

TERMINATE 1

В этом примере когда заявка поступает в блок TERMINATE, она удаляется из процесса моделирования. Также, счетчик завершения моделирования, устанавливаемый командой START, уменьшается на 1.

Алгоритм работы

Когда заявка поступает в блок TERMINATE, операнд A вычисляется, округляется и используется в качестве декремента для счетчика завершения моделирования. Если операнд A не был определен, счетчик завершения не меняется.

Активная заявка затем удаляется из процесса моделирования, и выбирается новая активная заявка.

Счетчик завершения моделирования устанавливается в начальное положение командой START, при этом начинает вестись необязательный стандартный отчет.

Особые ограничения

- A, если оно определено, должно быть положительным **Refuse Mode**

Режим отказа

Заявке никогда не будет отказано в доступе в блок TERMINATE

Родственные SNA

- *TG1* – Счетчик завершения моделирования. Это значение инициализируется командой START и указывает на завершение работы системы имитационного моделирования, когда становится меньше либо равен нулю.

TEST (ТЕСТИРОВАНИЕ)

Блок TEST сравнивает значения нормальных SNA и направляет активную заявку к месту назначения на основе результатов.

TEST O A,B,C

Операнды

О – Оператор отношения. Отношение операнда А к операнду В для успешного прохождения теста. Обязательный. Варианты значения оператора: E, G, GE, L, LE, или NE.

А – Тестируемое значение. Обязательный. В качестве операндов должны быть указаны *Name, Number, String, ParenthesizedExpression, SNA*, или *SNA*Parameter*.

В – Контрольное значение. Обязательный. В качестве операндов должны быть указаны *Name, Number, String, ParenthesizedExpression, SNA*, или *SNA*Parameter*.

С – Номер блока назначения. Необязательный. В качестве операндов должны быть указаны *Null, Name, PosInteger, ParenthesizedExpression, SNA*, или *SNA*Parameter*.

Алгоритм работы

Блок TEST в двух режимах: «режиме отказа» или «режиме альтернативного выхода». В обоих случаях, операнда А и В вычисляются численно и сравниваются.

Если операнд С не используется, блок TEST работает в режиме отказа. Когда заявка пытается войти в работающий в режиме отказа блок TEST, и проверка проходит неудачно, заявка блокируется, т.е. ей не позволено войти в блок TEST до тех пор, пока тест не будет пройден успешно. Когда проверка проходит успешно, активная заявка направляется к следующему блоку.

Заблокированные заявки помещаются в повторную очередь всех участвующих в сравнении объектов. Когда состояние одного из этих объектов меняется, заблокированная заявка восстанавливается, определенная блок TEST проверка повторяется, и, в случае успешного ее прохождения, заявка получает разрешение пройти блок TEST. Однако, интеграция пользовательских переменных не вызывает реактивацию заблокированной Заявки. Чтобы зафиксировать уровень одной или нескольких непрерывных переменных, необходимо использовать команду INTEGRATE. Это рассмотрено в главе 1, в разделе «Непрерывные переменные».

Если операнд С используется, блок TEST функционирует в режиме альтернативного выхода. Когда заявка пытается поступить в работающий в таком режиме блок TEST, и тест проходит неудачно, Заявка перенаправляется в блок альтернативного назначения, определенный операндом С, и помещается в CEC (цепь текущих событий) в начало группы заявок с таким же приоритетом. Если тест был пройден удачно, активная заявка поступает в блок TEST, а затем направляется в следующий блок.

Пример

TEST G C1,70000

В этом примере «режима отказа» блок TEST, активная заявка поступает в блок TEST если значение системного времени превышает 70000. В противном случае, активная заявка блокируется до тех пор, пока условие не будет выполнено.

TEST G Q\$Teller1_Line,Q\$Teller2_Line,Teller1

В этом примере «режим альтернативного выхода» блок TEST, активная заявка всегда поступает в блок TEST. Если длина очереди *Teller1_Line* превышает длину очереди *Teller2_Line*, заявка направляется в следующий блок (NSB). В противном случае, активная заявка направляется на вход блока с названием *TELLER1*.

Операторы отношения

Операторы отношения являются обязательными. Они могут быть: E, G, GE, L, LE, или NE.

Успешное прохождение теста определяется по следующим правилам:

- E – Значение операнда А должно быть равно значению операнда В.
- G – Значение операнда А должно быть больше значения операнда В.
- GE – Значение операнда А должно быть больше либо равно значению операнда В.
- L – Значение операнда А должно быть меньше значения операнда В.
- LE – Значение операнда А должно быть меньше либо равно значению операнда В.

- NE – Значение операнда A должно быть не равно значению операнда B.

Особые ограничения

- С должен идентифицировать существующий в системе моделирования блок.
- Блоки TEST обладают высокой мощностью, но они могут занимать очень много компьютерного времени на обработку неудачных тестов. Возможно, понадобится даже модифицировать систему моделирования для уменьшения количества повторений неудачных тестов. Это можно организовать посредством помещения Заявок, у которых нет шанса успешно пройти тест, в накопительную систему (User Chain), используя блоки LINK и UNLINK.

Режим отказа

Блок TEST, работающий в режиме отказа, откажется пропускать заявку, если результат теста будет отрицательным. Заявка, которой было отказано в доступе, будет заблокирована до тех пор, пока тест не будет пройден успешно.

Когда заявке отказывается в доступе, ее индикатор задержки включается и накапливается до тех пор, пока заявке не будет позволено пройти «синхронный» режим блока TRANSFER. «синхронный» режим блока TRANSFER используется редко, потому как булевы переменные могут более рационально контролировать координацию состояния определенного набора ресурсов при использовании блока TEST.

Вспомогательные окна

- Blocks Window (Окно блоков) – Просмотр смены блоков в реальном времени.
- Transaction Snapshot (Текущее состояние заявки) – Изображение состояния заявки в моделировании.
- CEC Snapshot (Текущее состояние CEC) – Изображение состояния цепи текущих событий в моделировании.
- FEC Snapshot (Текущее состояние FEC) – Изображение состояния цепи будущих событий в моделировании.

TRACE (ТРАССИРОВКА)

Блок TRACE включает индикатор трассировки активной заявки.

TRACE

Операнды

Отсутствуют.

Пример

TRACE

В этом примере индикатор трассировки активной заявки будет включен до тех пор, пока она не поступит в блока UNTRACE.

Алгоритм работы

Когда заявка поступает в блок TRACE, ее индикатор трассировки включается. Это активирует вывод трассировочных сообщений (уведомляющих о текущих действиях в системе) в Journal Window (журнальное окно) каждый раз, когда заявка попадает в новый блок.

Режим отказа

Заявке никогда не будет отказано в доступе в блок TRACE

Вспомогательные окна

- Journal Window (Журнальное окно) – Запись сессии событий.
- Blocks Window (Окно блоков) – Просмотр смены Блоков в реальном времени.
- Transaction Snapshot (Текущее состояние заявки) – Изображение состояния заявки в моделировании.
- CEC Snapshot (Текущее состояние CEC) – Изображение состояния цепи текущих событий в моделировании.

- FEC Snapshot (Текущее состояние FEC) – Изображение состояния цепи будущих событий в моделировании.

TRANSFER (ПЕРЕДАЧА)

Блок TRANSFER принуждает активную заявку перейти («перескочить») в расположение нового блока.

TRANSFER A,B,C,D

Операнды

A – Режим блока передачи. Описывается ниже. Необязательный. В качестве операндов должны быть указаны BOTH, ALL, PICK, FN, P, SBR, SIM, *fraction*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, *SNA*Parameter*, или *Null*.

B – Номер или местоположения блока. Имя или номер параметра в режиме P. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

C – Номер или местоположения блока. Нарращивает значение в режимах FN или P. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

D – Значении инкремента для блока во BCEX режимах. Значение по умолчанию 1. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Алгоритм работы

Блок TRANSFER может работать в 9 «режимах», каждый из которых обладает разными свойствами. Когда заявка попадает в блок TRANSFER, операнд A используется для определения режима функционирования блока. Значение операндов B и C зависят от режима. Когда операнды, связанные с расположением блока, не определены, после блока TRANSFER используется следующий блок.

Безусловный режим (Unconditional Mode)

Когда операнд A опускается, блок TRANSFER работает в безусловном режиме. В безусловном режиме активная заявка всегда перескакивает в блок, положение которого описывается операндом B.

TRANSFER,New_Place

Когда заявка пребывает в блок TRANSFER, она немедленно переадресовывается в блок с местоположением New_Place

Дробный режим (Fractional Mode)

Когда операнд A не является зарезервированным словом, блок TRANSFER работает в дробном режиме. В дробном режиме активная заявка всегда перескакивает в блок, положение которого описывается операндом C с вероятностью, определенной операндом A. Если операнд A неотрицательное вещественное число, оно представляется в долях тысячи и конвертируется в дробную вероятность. Альтернативное место назначения определяется операндом B или NSB (следующим блоком), если операнд B опущен.

TRANSFER .75,New_Place

Когда заявка пребывает в блок TRANSFER, она отправляется к месту назначения с именем NEW_PLACE с вероятностью 0,75. В оставшееся время она направится в следующий блок. Вы можете задать начальные значения для генератора случайных чисел. Это задается в пункте «Random» настроек модели (Model Settings Notebook).

Выберите Edit / Settings (Правка / Настройки)

Затем выберите пункт **Random**. Заполните требуемое случайное значение в поле с пометкой «TRANSFER». Значение по умолчанию 1.

Двусторонний режим (Both Mode)

Когда на операнде A устанавливается значение BOTH, блок TRANSFER работает в двустороннем режиме. В двустороннем режиме, блок, определенный операндом B, проверяется. Если он отказывается принять активную заявку, проверяется блок,

определенный в операнде С. Первый блок, принявший заявку, и будет новым блоком назначения. Если ни один блок не согласился принять заявку, она остается в блоке TRANSFER до тех пор, пока она не сможет попасть в один из этих блоков.

TRANSFER BOTH,First_Place,Second_Place

Когда заявка пребывает в блок TRANSFER, проверяется блок с местоположением First_Place. Если заявке разрешено войти, она поступает на этот блок. Если нет, проверяется блок с местоположением Second_Place. Если заявке разрешено войти, она поступает на этот блок. В противном случае, заявка остается в блоке TRANSFER, пока ей не разрешено будет поступить в один из требуемых блоков.

Общий режим (All Mode)

Когда на операнде А устанавливается значение ALL, блок TRANSFER работает в общем режиме. В общем режиме, блок, определенный операндом В, проверяется. Если он отказывается принять активную заявку, блоки проверяются последовательно, до тех пор, пока не достигается блок, определенный операндом С, если не один из ранее проверяемый на согласие блоков не принял заявку, она отправляется в блок, определенный операндом С. Местоположение каждого из последовательно тестируемых блоков вычисляется посредством прибавления операнда D к местоположению предыдущего тестируемого блока. Если операнд D не используется, каждый блок между теми. Которые определены в В и С, включительно, проверяются. Если операнд С не используется, проверяется только один блок. Ни один блок, расположенный выше блока, определенного операндом С, не проверяется. Первый блок, разрешивший поступление заявки, становится ее новым местом назначения. Если ни один блок не согласился принять заявку, она остается в блоке TRANSFER до тех пор, пока она не сможет попасть в один из этих блоков.

TRANSFER ALL,First_Place,Last_Place,2

Когда заявка пребывает в блок TRANSFER, проверяется блок с местоположением First_Place. Если заявке разрешено войти, она поступает на этот блок. Если нет, проверяется каждый второй расположенный выше блок. Если заявке разрешено войти, она поступает на этот блок. Если все тестируемые блоки отказали, проверка заканчивается на блоке, местоположение которого определено как Last_Place или на блоке, идущем непосредственно перед ним, – это зависит от того, сколько блоков разделяют First_Place и Last_Place. Если ни один блок не согласился принять заявку, она остается в блоке TRANSFER до тех пор, пока она не сможет его покинуть.

Случайный режим (Pick Mode)

Когда на операнде А устанавливается значение PICK, блок TRANSFER работает в случайном режиме. В случайном режиме, место назначения выбирается случайным образом.

TRANSFER PICK,First_Place,Last_Place

Когда заявка пребывает в блок TRANSFER, местоположение выбирается случайным образом из последовательности чисел между First_Place и Last_Place,включительно. Выбранное местоположение становится новым местом назначения для активной заявки. Вы можете задать начальные значения для генератора случайных чисел. Это задается в пункте «Random» Настроек Модели (Model SettingsNotebook).

ВЫБЕРИТЕ Edit / Settings (Правка / Настройки)

Затем выберите пункт **Random**. Заполните требуемое случайное значение в поле с пометкой «TRANSFER». Значение по умолчанию 1.

Функциональный режим (Function Mode)

Когда на операнде А устанавливается значение FN, блок TRANSFER работает в функциональном режиме. В функциональном режиме, место назначения выбирается посредством вычисления значения функции, определенной в операнде В, и прибавлением необязательного инкремента, определенного в операнде С.

TRANSFER FN,Func1,5

Когда Заявка пребывает в Блок TRANSFER, для определения местоположения места назначения, вычисляется функция FUNC1, и к найденному значению прибавляется 5.

Режим параметра (Parameter Mode)

Когда на операнде А устанавливается значение Р, блок TRANSFER работает в режиме параметра. В режиме параметра, активная заявка перемещается к местоположению, высчитанному как сумма значения параметра и операнда С. Если С не определен, значение параметра является местонахождением нового места назначения.

TRANSFER P,Placemark,1

Когда заявка пребывает в блок TRANSFER, она немедленно передается в блок, следующий непосредственно за блоком, местоположение которого указано в параметре заявки Placemark.

Режим подпрограммы (Subroutine Mode)

Когда на операнде А устанавливается значение SBR, блок TRANSFER работает в режиме подпрограммы. В Режиме подпрограммы, активная заявка всегда переходит к местоположению, определенному операндом В. Местоположение блока TRANSFER помещается в параметр, определенный операндом С.

TRANSFER SBR,New_Place,Placemark

Когда заявка пребывает в блок TRANSFER, она немедленно передается в блок с местоположением New_Place. Местоположение блока DISPLACE помещается в параметр Placemark. Если такого параметра не существует, он создается.

Для возвращения из подпрограммы, используется блок TRANSFER. Работающий в режиме параметра, описанном выше.

Синхронный режим (Simultaneous Mode)

Когда на операнде А устанавливается значение SIM, блок TRANSFER работает в синхронном режиме. В синхронном режиме, активная заявка перемещается в одно из двух мест назначения, в зависимости от индикатора задержки заявки. Если индикатор задержки включен, заявка перемещается к местоположению, определенному операндом С и индикатор задержки выключается. Если индикатор задержки выключен, заявка перемещается к местоположению, определенному операндом В.

Индикатор задержки заявки включается, когда заявке отказывается в доступе в блок. Индикатор задержки остается включенным до тех пор, пока не войдет в синхронный режим блока TRANSFER.

TRANSFER SIM,Nodelay_Place,Delay_Place

Когда заявка пребывает в блок TRANSFER, она немедленно передается в блок с местоположением DELAY_PLACE, если ее индикатор задержки включен, или в блок с местоположением NODELAY_PLACE, если он выключен. После перемещения индикатор задержки всегда выключается.

Синхронный режим используется редко. Гораздо более эффективно использовать булевы переменные в режиме отказа блока TEST, при необходимости скоординировать состояние объектов. Синхронный режим развивался до появления в языке GPSS булевых переменных.

Особые ограничения

- Во всех режимах, операнд С должен быть больше операнда В, и, если используется D, (C–B) должно быть четно кратно D.
- Все высчитанные места назначения для заявок должны быть допустимыми значениями местоположений блоков
- В Двустороннем и Общем Режиме можно потратить очень много компьютерного времени на обработку неудачных результатов проверок. Существует возможность поместить заявку в накопительную систему до тех пор, пока не появится возможность успешного прохождения проверки. Это может быть сделано при помощи блоков LINK и UNLINK.

Режим отказа

Заявке никогда не будет отказано в доступе в блок TRANSFER. Если заявка оказалась заблокирована из-за отказов блоков назначения, она оставляется в блоке TRANSFER.

Вспомогательные окна

- Blocks Window (Окно блоков) – Просмотр смены блоков в реальном времени.
- Transaction Snapshot (Текущее состояние заявки) – Изображение состояния заявки в моделировании.
- CEC Snapshot (Текущее состояние CEC) – Изображение состояния цепи текущих событий в моделировании.
- FEC Snapshot (Текущее состояние FEC) – Изображение состояния цепи будущих событий в моделировании.

UNLINK (РАЗЪЕДИНЕНИЕ)

Блок UNLINK удаляет заявки из накопительной системы

UNLINK O A,B,C,D,E,F

Операнды

O – Оператор отношения. Отношение D к E для активации удаления. Выбор обуславливается ниже. Необязательный. В качестве операндов должны быть указаны *Null*, *E*, *G*, *GE*, *L*, *LE* или *NE*.

A – Номер накопительной системы. Накопительная система, из которой будут удалены одна или более заявок. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

B – Номер блока. Блок назначения для удаляемых заявок. Обязательный. В качестве операндов должны быть указаны *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

C – Предел удаления. Максимальное количество заявок, которое может быть удалено. Если не определено, используется параметр *ALL*. Необязательный. В качестве операндов должны быть указаны *ALL*, *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

D – Тестируемое значение. Имя или номер параметра заявки, используемое для проверки на необходимость удаления, или параметр *BACK* для удаления из конца очереди. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, *SNA*Parameter* или *BACK*.

E – Контрольное значение. Значение, с которым сравнивается значение операнда *D*. Необязательный. В качестве операндов должны быть указаны *Null*, *Name*, *Number*, *String*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*. операнд *E* не используется, если операнд *D* содержит значение булевой переменной или *BACK*.

F – Номер блока. Альтернативный блок назначения для поступившей заявки. Необязательный. В качестве операндов должны быть указаны *e Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, или *SNA*Parameter*.

Пример

UNLINK OnHold,Reentry,1

Это простейший способ использования блока UNLINK, первая заявка во главе очереди *OnHold*, если такая существует, изымается из очереди и направляется в блок с пометкой *Reentry*. Она помещается в CEC за заявками, с таким же приоритетом. Заявка, поступившая в блок UNLINK, направляется в следующий блок.

Алгоритм работы

Блока UNLINK удаляет заявку из накопительной системы и направляет ее в новый блок. Поступающие на удаление заявки могут выбираться, и предел удаления может фиксировать количество удаленных заявок. Если в накопительной системе не окажется заявок при обращении к ней блока UNLINK, индикатор очереди накопительной системы

выключается. Также, заявка, поступающая в блок UNLINK, может быть перенаправлена в соответствии со значением необязательного операнда F.

Чтобы ограничить количество заявок, удаляемых из пользовательской цепочки, укажите операнд C. Если операнд C опускается, подразумевается значение BCE.

Каждую транзакцию можно проверить на удаление с помощью оператора отношения или операндов D и E, оба из которых оцениваются в числовом виде.

Операнды D и E и условный оператор не обязательны. Если они опускаются, то из начала цепочки убираются все заявки до тех пор, пока не кончается цепочка или не достигается лимит (операнд C).

Операнд D может быть булевой переменной, номером параметра или словом BACK. Если операнд D является булевой переменной, он оценивается в соответствии с заявкой в цепи. Если результат ненулевой, то заявка удаляется. Если операнд D имеет значение BACK, заявки удаляются из «хвоста» пользовательской цепочки до достижения лимита. В противном случае операнд оценивается в соответствии с заявкой члена пользовательской цепочки и используется в качестве номера параметра, значение которого возвращается из пользовательской цепочки как итоговый результат. Это итоговое значение сравнивается с результатом оценивающего операнда E.

Если D указывает на параметр, а E не используется, то параметр заявки пользовательской цепочки сравнивается с тем же параметром активной заявки. Если они равны, то связанная заявка удаляется из цепочки.

Операнд E используется только в том случае, если используется относительный оператор. В этом случае требуется также операнд D. Пользовательская цепочка сканируется с начала. Каждая заявка, до достижения лимита (операнд C) удаляется, если операнд D так же относится к операнду E, как указано в относительном операторе. Если операнд E является транзакционным SNA, он оценивается в соответствии с активной заявкой.

Вы можете использовать относительный оператор для указания отношения между атрибутом заявки (операнд D) и значением ссылки (операнд E), что запустит процедуру удаления заявки. Значением по умолчанию для относительного оператора E является равенство. Если оператор отношения не используется, но используются операнды D и E, значения должны быть равны, чтобы заявка была удалена из цепочки.

Операнд F используется для указания альтернативного назначения, вводимого заявкой при невозможности достичь уровня лимита (операнд C) или при невозможности удаления заявок. Если операнд F не используется, вводимая заявка всегда переходит к следующему последовательному блоку.

Режим отказа

Заявка никогда не отказывает в доступе блоку UNLINK.

Связанные SNA

- *CAEntnum* – Среднее содержимое пользовательской цепочки.
- *CCEntnum* - Общее количество записей пользовательской цепочки. Количество всех заявок, связанных с пользовательской цепочкой *Entnum*.
- *CHEntnum* - Текущее содержимое пользовательской цепочки. Текущее количество заявок, связанных с пользовательской цепочкой *Entnum*.
- *CMEntnum* – Максимальное содержимое пользовательской цепочки. Максимальное количество заявок, связанных с пользовательской цепочкой *Entnum*. "Линия высокой воды".
- *CTEntnum* – Среднее время нахождения в пользовательской цепочке. Средняя продолжительность заявок в цепочке *Entnum*.

Связанное окно

Userchains Snapshot – изображение состояния пользовательских цепочек в симуляции.

UNTRACE

Блок UNTRACE отключает индикатор слежения активной заявки.

UNTRACE

Операнды

Нет

Пример

UNTRACE

В этом примере все заявки, проходящие через блок UNTRACE проходят сброс индикаторов слежения. Эти заявки больше не будут возвращать сообщения слежения при перемещении от блока к блоку.

Действие

Когда заявка входит в блок UNTRACE, её индикатор слежения отключается. Таким образом, от заявки не будут поступать сообщения слежения, пока она не войдёт в блок TRACR.

Сообщения слежения посылаются во все открытые окна журнала.

Режим отказа

Заявки никогда не отказывают в доступе блоку UNTRACE.

Связанные окна

- Journal Window – Запись событий сеанса.
- Transaction Snapshot – отображение состояния заявок в имитации.
- CEC Snapshot – отображение состояния текущих пользовательских цепочек в имитации.
- FEC Snapshot – отображение состояния будущих пользовательских цепочек в имитации.

WRITE

Блок WRITE передает текстовую строку в поток данных.

WRITE A,B,C,D

Операнды

- **A** – текстовая строка. Обязательный. Оценивается как строка. Операнд должен быть в формате *Имя, Номер, строка, выражениеВСкобках, SNA* или *SNA*параметр*.
- **B** – Номер потока данных. Необязательный. Операнд должен быть в формате *Ноль, Имя, Целое, выражениеВСкобках, SNA* или *SNA*параметр*. Значение по умолчанию – 1.
- **C** – Имя или номер альтернативного блока назначения. Необязательный. Операнд должен быть в формате *Ноль, Имя, Целое, выражениеВСкобках, SNA* или *SNA*параметр*.
- **D** – режим вставки. Необязательный. Операнд должен быть в формате *Ноль, ВКЛ* или *ВЫКЛ*. Значение по умолчанию – ВКЛ.

Пример

WRITE "New Line 20",1,Done

В этом примере блок WRITE посылает текстовую строку потоку данных под номером 1. Если возникает ошибка, то активная заявка обращается к блоку с именем Done. В противном случае она перемещается на следующий последовательный блок. В этом случае, если поток данных является потоком I/O или потоком In-Memory, команда WRITE выполняется в режиме вставки.

Действие

Когда заявка входит в блок WRITE, операнд A оценивается как строка. Числовые значения преобразуются в строковый эквивалент ASCII. Затем из операнда B определяется принадлежность потока данных.

Если используется операнд B, он немедленно оценивается, округляется и используется, как номер объекта потока данных. Это должно быть положительное целое число. Если операнд B не используется, то предполагается номер потока 1.

Если используется операнд C, то любая ошибка, возникшая в процессе команды WRITE, вызывает переход заявки к блоку с этим номером.

Операнд D задает режим записи, как описано ниже.

В любом случае, при обнаружении ошибки, код ошибки сохраняется в программе. Для извлечения ошибки используется блок CLOSE. Глава 4 (4.16) описывает потоки данных и коды ошибок.

Связанные блоки

- OPEN – Создание потока данных.
- CLOSE – Закрытие потока данных.
- READ – Извлечение текстовой строки из потока данных.
- SEEK – Задание текущей позиции строки потока данных.

Глава 8 - PLUS: Язык программирования моделирования

Модель GPSS World – это последовательность выражений модели. Выражение модели может быть выражением GPSS или процедурой PLUS.

PLUS выражения может существовать как операнды в выражениях GPSS так же как в пределах PLUS процедуры. Выражения могут содержать запросы процедуры, которые вызывают или встроенную или пользовательскую PLUS процедуру. Кроме того, все операции Входа/Продукции потока доступны как встроенные процедуры библиотеки. Это означает, что Вы можете читать из фаулов и посылать данные файлам, используя блоки GPSS или указывая соответствующие процедуры библиотеки, или при использовании обоих методов в комплексе. Не только, что, но и теперь Вы можете вызвать внешние программы, которые существуют на вашей системе в EXE или файлах DLL при использовании новых динамических процедур запроса. Эта глава содержит информацию относительно PLUS языка, и относительно встроенных процедурных библиотек.

Как эта глава описывает синтаксис

Действительная форма PLUS-выражения и запроса процедуры обозначена в строке синтаксиса в начале описания выражения. Выполнение PLUS-выражения подобно заполнению бланков формы. Строка синтаксиса описывает постоянные и переменные части выражения. Вам решать, что "иллюстрировать в примерах" переменной части выражения, чтобы заставить сделать то, что требуется. Пункты в линии синтаксиса сообщают Вам, как закодировать выражение. При этом должны соблюдаться следующие правила:

1. Ключевые слова вводятся в строку синтаксиса заглавными буквами. Эти слова, и невыделенные круглые скобки, должны быть введены по умолчанию. Однако, когда Вы кодируете выражение, написание ключевых слов с большой буквы является необязательным. Например,

ReturnString = LEFT(SourceString, MaxCount)

содержит название процедуры "Left" заглавными буквами. Хотя Вы можете изменить написание любым требуемым способом, Вы в любом случае должны заключать слово "Left" в круглые скобки, при вызове процедуры. Когда используется PLUS оператор присваивания, в конце строки должна быть добавлена точка с запятой.

2. Тип данных результата вызова процедуры дается в строке синтаксиса слева от знака равенства (=). Процедуры могут быть вызваны PLUS-выражением запроса процедуры. В PLUS оператор присваивания, "lvalue" (назначающий значение или матричный элемент) требуется вводить значение слева от оператора присваивания. Это действие возвращает результат запроса процедуры.

3. Аргументы задаются в строке синтаксиса полужирным курсивом заглавными буквами в виде семантических переменных. Они указывают, что Вы должны сделать выбор из класса возможностей. Определение каждого аргумента следует производить синтаксически грамотно. Аргументы должны быть определены, как и возможные варианты, задаваемые Вами одной или несколькими синтаксическими переменными в нежирном курсиве, например в "Должен быть ParenthesizedExpression", означает, что при вводе выражения Вы должны использовать действительное PLUS-выражение, заключенное в круглые скобки. Синтаксические переменные определены формально в приложении. Действительные формы, которые Вы должны использовать в экземпляре синтаксических переменных, описаны в нескольких следующих секциях. выражение и описания процедуры следуют далее.

8.1 Определение процедур PLUS

Процедуры обычно используются двумя различными способами. Они могут использоваться как глобальные переменные и называться переменными или возвращать значение выражению. Если Вы просто желаете обновить глобальную переменную или

изменить присвоенное значение, выражение возврата не требуется. Во всех случаях, где будет использоваться результат PLUS-процедуры в месте вызова процедуры (например в операнде или другом PLUS-выражении), выражение возврата необходимо. Определить PLUS-процедуру легко. Все, что Вы должны сделать, это поместить действительное выражение PROCEDURE в файл модели и оттранслировать его с моделью, или послать выражение PROCEDURE в существующую симуляцию. После этого, Вы можете вызвать вашу процедуру для оценки выражения или PLUS-оператора присваивания точно так же, как и любую другую процедуру библиотеки. Как простой пример, рассмотрите следующее выражение PROCEDURE:

PROCEDURE SetPop(Pop_Level) Foxes = Pop_Level ;

Хотя большинство процедур более сложно, это - все, что необходимо, чтобы определить её. В этой процедуре даже не объявлено значение возвращения, таким образом по умолчанию используется значение 0. Единственная цель этой процедуры состоит в том, чтобы использовать значение, Pop_Level, который передается процедуре и устанавливает определенную переменную, **Foxes**, равной этому значению. Это, или любое другое выражение PROCEDURE можно было бы даже послать в существующую имитацию, чтобы определить или пересмотреть процедуру "SetPop". Тогда, любое PLUS-выражение в моделировании может включать в себя запрос процедуры, например

SetPop(Rabbits/10);

которая назначает значение пользовательским переменным **Foxes**. Конечно, здесь мы предполагаем, что пользовательской переменной **Rabbits** уже присвоили значение. Вы можете определить временные матричные объекты и временные пользовательские переменные, которые существуют только по запросу процедуры. Это делается при помощи выражения TEMP в спецификации процедуры. Когда PLUS-процедура вызвана запросом процедуры, в вызванной процедуре выражения "выполняются" последовательно. Большинство процедур определено как внешнее комплексное выражение, содержащее список выражений. Выполнение списка выражений происходит последовательно, причем выполняется каждое выражение. Составные выражения, IF-выражения, IF-ELSE выражения, WHILE-выражения и выражения GOTO могут изменить нормальную последовательность выполнения. Если выражение RETURN выполнено, вызов процедуры закончен, и вся память, используемая этим процессом будет освобождена. В PLUS-процедур выражение RETURN необязательно. Если в течение запроса вашей процедуры, обработка когда-либо достигнет конца списка выражений процедуры, то запрос процедуры завершается, и возвращается значение 0. Что-то всегда используется, как значение возвращения, в случае если оно необходимо при оценке выражения. Если Вы не определяете значение в вашем выражении RETURN, или если ваше последнее выражение в списке выражений закончено, используется нулевой результат.

8.2 Язык

Вы можете включить в ваши модели GPSS World PLUS выражения или PLUS процедуры и эксперименты, построенные с использованием PLUS языка. Для этого ознакомьтесь со стандартными блоками языка, и способами их соединения. Им посвящены оставшиеся разделы главы.

8.2.1 Набор символов

Набор символов GPSS состоит из стандартных обозначающих и специальных символов. Символы обозначения включают заглавные буквы A-Z, строчные букв a-z, цифры 0-9, и подчеркивание (_). Специальные символы используются для обозначения операторов и пунктуации. В их число входят: «*» и «+» «-» «/» «\»; символ «^» также рассматривают как оператор. Обычно, символ «*» используется как оператор умножения и как GPSS SNA

косвенный оператор адресации. Если Вы предпочитаете, Вы можете полностью изменить их, щёлкнув "Переключить * и #" на странице 1 вкладки параметров настройки. Использование «/» редактируется в параметрах настройки на главном меню. Мы рекомендуем использовать последовательный стиль при написании прописных или строчных букв. Ваша модель будет намного более удобочитаемой, если Вы будете следовать этому совету. Например, Вы можете выделить ваши пользовательские переменные заглавными буквами, или выделить таким образом ключевые слова. С другой стороны, слова, где первая буква в верхнем регистре, остальные - в нижнем, имеют тенденцию быть самыми читабельными.

8.2.2 Имена

Названия - последовательности символов, созданные Вами, чтобы идентифицировать объекты, переменные, и местоположения программы. Символы обозначения - буквы, цифры, и подчеркивания. Есть несколько правил, которым Вы должны следовать, когда Вы создаете название. Вы должны использовать от 1 до 250 символов, и Вы должны начать название с буквы. Кроме того, ваше название не должно совпадать с ключевым словом GPSS, системным числовым признаком, или классом SNA. Ключевые слова перечислены в приложении. GPSS World – нечувствителен к регистру. Различие прописных/строчных букв не имеет значения. Только символы в константах строки и комментарии сохраняют строчные буквы. Все другие строчные буквы программно приводятся к прописным. Это устраняет опасность одинакового написания двух переменных, которые относятся к различным значениям. Первая задача при создании названия - избежать ключевых слов. При желании Вы можете обратиться к списку ключевых слов в приложении, но есть более легкий путь. Все, что Вы должны сделать, это включить во все имена подчеркивание после первого символа, который должен быть буквой. Это гарантирует, что Вы не будете сталкиваться с ключевыми словами GPSS.

Именнованные значения

Именнованные значения - названия, которые Вы поместили в область Label или в оператор присваивания PLUS. Вы можете использовать их, чтобы идентифицировать объект или для хранения значения. Если Вы используете их, чтобы маркировать выражение GPSS, определяющее объект, их называют ярлыками объекта. Если Вы создаете их, назначая значение, как в команде EQU или операторе присваивания PLUS, они называются пользовательскими переменными. Именованные переменные обычно являются глобальными. Вы можете обратиться к ним в любом месте модели. Есть два вида именованных переменных: ярлыки объекта и пользовательские переменные.

Пользовательские переменные

Пользовательская переменная – именованная переменная, не используемая как ярлык объекта. Вы можете дать ей числовое значение или значение строки, и Вы можете объединить её в непрерывную переменную. Пользовательские переменные могут быть глобальными для употребления в любом месте модели, или местными, применяемыми только в пределах единственной PLUS-процедуры. Последние объявляются во временной декларации PLUS-процедуры, в которой они определены. Все другие пользовательские переменные глобальны. Вы можете назначить значение пользовательской переменной через команды EQU, через PLUS-операторы присваивания, или через числовую интеграцию, с помощью команды INTEGRATE. Пользовательские переменные должны калиброваться перед использованием. Вы должны назначить им значения прежде, чем Вы можете использовать их в выражениях или при интеграции. Вы можете наблюдать значения пользовательских переменных в окнах плана и окнах выражений, которые были открыты в вашей имитации.

Имена процедур

Вы должны назвать PLUS-процедуру или эксперимент, когда Вы определяете его. После этого, Вы можете вызвать ту же самую процедуру запросом процедуры, используя то же самое название. По области использования названия процедуры глобальны. Любая процедура может быть вызвана из любого выражения в модели. Если Вы определяете процедуру с данным названием, любая существующая определенная PLUS процедура с тем названием пересматривается. Эксперименты - специальные виды процедур. Они идентичны процедурам в синтаксисе за исключением того, что ключевое слово EXPERIMENT заменяет PROCEDURE. Эксперименты могут быть вызваны только командой CONDUCT.

Ярлыки матриц

Подобно пользовательским переменным, матричные объекты могут быть местными или глобальными. Матрица, определенная во временной матрице PLUS-декларации является местной, известной только в процедуре, в которой она была объявлена. Такая матрица создается при инициации запроса процедуры и разрушается при его завершении. Глобальные матричные объекты объявлены в команде GPSS MATRIX. Ярлык, который Вы используете, становится матричным ярлыком объекта. Глобальные матричные объекты постоянны и на них можно ссылаться в любом месте модели. Все матрицы созданы с некалиброванными элементами. Вы должны назначить значения на них до их использования в выражениях. Чтобы рассмотреть динамику любого поперечного сечения матрицы, Вы можете использовать окно матрицы. Кроме того, Вы можете наблюдать значения матричных элементов в окнах плана и окнах выражений, которые были открыты для вашего моделирования.

8.2.3 Выражения

PLUS-выражение - комбинация одного или более элементов, названных факторами. Выражения построены при использовании операторов и запросов процедуры. Правила построения выражений описаны в следующих секциях руководства. выражения могут использоваться в PLUS-процедурах и в операндах выражений GPSS. Обычно, когда выражение используется в операнде блока, он должно быть заключено в круглых скобках. Список приемлемых синтаксических переменных тогда будет включать *ParenthesizedExpression* как одну из альтернативных форм операнда. Для совместимости некоторые команды не должны использовать внешние круглые скобки, но если Вы всегда будете вводить выражения, используемые в выражениях GPSS, то Вы не ошибётесь.

Типы данных

Любая пользовательская переменная, матричный элемент, Savevalue, или операционный параметр могут иметь значение на любой из нескольких форм - типов данных. Более того, каждый из них имеет неинициализированную форму, которая предотвращает их использование прежде, чем им было назначено значение. Три главных типа данных - целое число, вещественное и строка. Первые два могут определяться числовыми типами данных. Целые числа – это числа на 32 бита. Если в течение арифметических операций, целое число выходит за пределы диапазона, оно преобразуется к вещественному числу. Вещественные типы данных имеют числа с плавающей запятой двойной точности. Они имеют 15 десятичных цифр после запятой и диапазон экспонентов от -306 до 306. Строки – это последовательности символов ASCII. Они могут быть иметь любой размер, вплоть до максимального запроса памяти, объявленного на странице "Моделирование" вкладки настройки. Целый класс строковых процедур находится в библиотеке процедуры для того, чтобы создать и управлять строками. Типы данных явно преобразуются из одного в другой запросами процедуры, или неявно в течение оценки выражений. Это обсуждено ниже.

Факторы

Факторы - основные стандартные блоки выражений. Вы комбинируете их в выражениях, которые, в свою очередь, могут использоваться в операндах выражений GPSS и PLUS процедурах.

Факторы выражения GPSS:

1. Строковые константы, например "A stitch in time ...".
2. Действительные константы, например 201.6.
3. Целые константы, например 17.
4. Имена, например Water_Level.
5. PLUS-элементы матрицы, например Array1[P\$Part, X\$Order_Index+20].

Матричные элементы должны быть *Именем [ExpressionList]*. Список выражения может содержать от 1 до 6 выражений, соответствуя измерениям матрицы. Каждое выражение может измениться по сложности от простого целого числа до очень сложного PLUS-выражения.

6. Вызовы процедуры, например Word(X\$Quote,2) or Myproc(X\$Arg1,X\$Arg2).
Вызовы процедуры должны быть *Именем(ExpressionList)*.
7. Системные числовые атрибуты, например AC1, F\$My_Facility, MX\$Mat1(2,1), и SR*My_Parm.

Операторы и запросы процедуры используются, чтобы объединить факторы в выражения, следуя правилам, связанным с каждым оператором или процедурой. Они обсуждены ниже.

Операторы

Арифметические операторы выражений GPSS World описаны здесь в порядке убывания значимости. Все арифметические операторы приводят строковые операнды к числовым значениям.

Операнд, окруженный левыми ассоциативными операторами равного предшествования обрабатывается оператором с его левой стороны. Выражения – это комбинации одного или более факторов, связанных операторами, и оцененные согласно четкому набору правил.

Оценка

Когда выражение оценивается, значения определяются и объединяются, чтобы вычислить заключительный результат. При этом производятся следующие действия:

- Строки и числовые константы приводятся к копии себя.
- Именованные значения приводятся к связанным значениям.
- SNAs оценивают с помощью моделирования или значения статуса.
- Запросы процедуры оценивают каждый аргумент, затем возвращают результат запроса процедуры. Встроенные процедуры дают команду аргументам.
- Операторы оценивают один или два операнда, приводят промежуточные результаты к числовым значениям, и вычисляют итоговое значение операции.

Операторы в GPSS имеют порядок приоритетности, который затрагивает выражения, которые Вы создаете. Например, оператор возведения в степень «^» имеет более высокий приоритет чем оператор умножения «*». Это означает, что выражение

$$4 * 3 ^ 2$$

будет приравнено к результату 36, а не 144, потому что первый промежуточный результат сформирован, оценивая оператор с более высоким приоритетом. Если Вы хотите установить порядок оценки, отличный от подразумеваемого иерархией приоритетов оператора, Вы должны использовать круглые скобки для управления оценкой. В продолжение последнего примера,

$$(4 \# 3) ^ 2$$

Будет равно 144.

Операторы GPSS World "перегружены" в смысле, что их операнды принуждены к надлежащему типу данных непосредственно перед тем, как операция применена. Поэтому, Вы не должны волноваться о типах данных, создавая PLUS выражения.

Оценка выражений

Выражения – это комбинации математических операторов, функций библиотеки, SNAs, и констант, которые подчиняются правилам элементарной алгебры. Выражение оценивается согласно иерархии операторов, упомянутых выше, или слева направо. Порядком оценки можно управлять, вставляя соединенные круглые скобки. Выражения могут быть *оценены*, *оценены в цифровой форме*, или *оценены, как строка*. Последние два способа отражают дополнительный шаг, примененный после нормальной оценки.

Когда выражение *оценено в цифровой форме*, результат строки преобразуется к его числовому эквиваленту, основанному на числовых символах, которые начинают строку. Строка, начинающаяся с нечисловых символов преобразуется к числовому нулю.

Точно так же, когда выражение *оценено как строка*, любой числовой результат преобразуется к эквивалентной строке. В сообщениях и потоках данных представлением большого количества можно управлять, подавая научный формат представления, как сделано во вкладке блокноте параметров настройки модели: Это устанавливается на странице "сообщения" вкладки параметров модели.

Выберите **Edit / Settings**

И выберите страницу **сообщения**. После этого установите флажок, отмеченный «Научный формат представления». В научном формате представления используется основное значение, сопровождаемое степенью десяти. Например, число одна тысяча сто было бы представлено в научном формате как 1.1e3.

Специальные правила применяются, когда моделированием управляют в режиме совместимости GPSS/PC. Это обсуждено более полно в главе 1. В режиме совместимости SNA округляются, а промежуточные результаты обрезаются при оценке объектов Bvariable и Variable.

Преобразование данных

Вы не должны волноваться о типах данных и преобразованиях. Все преобразования совершаются автоматически.

Типы данных по необходимости преобразуются, когда значения используются или обрабатываются как аргументы процедуры. Например, строки, объединенные арифметическим оператором «+», преобразуются в числовые значения, а результаты складываются, вместе приводя к числовому результату. Точно так же процедуры строки, которые берут строку как аргумент, преобразуют числовое значение в его эквивалентную строку, в случае необходимости.

Элементы данных не изменяются в ходе оценки, если они не обрабатываются или не используются как аргумент процедуры. Арифметические операторы изменяют аргументы на числовые значения перед выполнением операции, и все процедуры преобразуют каждый аргумент к подходящей форме. Это делается автоматически. Преобразование происходит только с операндом или оператором, или с аргументами встроенной процедуры в течение оценки.

Аргументы пользовательской PLUS-процедуры не преобразуются до запроса процедуры. Однако, преобразование операнда и аргументов может произойти в пределах тела пользовательской процедуры, если требуется другим PLUS выражением. Когда операнд или аргумент преобразованы к строке, любое числовое значение должно быть преобразовано к его эквивалентному ASCII-значению.

Глава 9 – Дополнительные разделы

9.1 Цепи заявок

Заявки временно привязаны к другим объектам, занимая связанные списки, названные цепями. Некоторые объекты, типа средств обслуживания, имеют несколько цепей. Другие объекты имеют только единственную цепь повторения. Каждая заявка может быть на любом числе цепей. Однако, занятие одного вида цепи иногда устраняет занятие той же самой заявкой другого. Например, заявка на одной или более цепях прерываний не может быть на будущей цепи событий. Одна заявка может длиться не больше, чем одна из следующих цепей:

- цепь будущих событий
- цепь текущих событий
- цепь задержки оборудования или хранения
- цепь ожидания оборудования
- пользовательская цепь

Заявка может ждать совершения любого числа условий, может быть в любом числе операционных групп, и может быть выгружена из любого числа средств обслуживания в любой момент. Это означает, что любая заявка может быть на любом числе цепей прерыва и любого числа цепей группы и любого числа цепей повторения в то же самое время.

Цепь текущих событий

Цепь текущих событий (СЕС) - связанный список готовых заявок, которые имеют блоки, чтобы быть введенными перед моделируемыми интервалами времени. Хотя СЕС сохранена в порядке приоритета, активная заявка обычно возвращается СЕС перед ее одноранговыми узлами. Поэтому, как только заявка начинает перемещаться в имитации, она продолжает движение, если не активирована заявка более высокого приоритета. Когда активная заявка останавливается в одной из операционных цепей, активной заявкой становится самая высокая заявка приоритета, остающаяся на СЕС. Если СЕС пуста, ближайшая заявка на цепи будущих событий перемещается в СЕС.

Цепь будущих событий

Цепь будущих событий (FEC) - цепь, которая содержит заявки, которые ожидаются в течение более позднего моделируемого времени. Когда вся деятельность моделирования в течение текущего времени часов максимальна, следующая заявка берется от FEC. Именно это действие заставляет часы системы обновляться. Значение системных часов всегда равно запланированному времени последней заявки, которая будет получена от FEC. Блоки ADVANCE и GENERATE - единственный способ поместить транзакцию в FEC. Эти блоки берут приращение времени в качестве операнда и вычисляют абсолютное время перед размещением заявки на FEC. Когда часы системы достигают этого абсолютного времени, заявка перемещается в СЕС так, чтобы она могла возобновить свое движение в имитации. Таким образом могут моделироваться интервалы времени между началом работы заявок.

Блоки PREEMPT и DISPLACE могут использоваться, чтобы удалить заявки из FEC. Такие заявки могут быть запланированы заново при входе в другой блок ADVANCE.

Когда планировщик должен выбрать новую активную транзакцию, если он не может найти её в цепи СЕС, берётся заявка из FEC. Такое удаление одной или более заявок всегда заставляет системные часы продвигаться. Когда больше чем одна заявка постоянно находится в FEC, в следующий раз возникает временной затор. Если параметры модели определяют перемещение заявок временной связи из СЕС в FEC, их порядок делается случайным в пределах приоритетов. Это сделано, чтобы предотвратить развитие случайных циклов обработки. Временной затор – это возникновение больше чем одной заявки с тем же самым значением времени впереди FEC. При обнаружении временного

затора конфликтующие заявки удаляются из FEC в случайном порядке. С этой целью GPSS World получает псевдо случайные числа от генератора случайных чисел, определенного на странице «Моделирование» вкладки настроек модели. Все удаленные заявки помещаются в порядок приоритета относительно цепи текущих событий (CEC).

Цепь повторения

Заявки, которые подводят все тесты, требуемые для элемента блока, помещены в цепь повторения. Эти тесты происходят, когда заявка пытается войти в блоки GATE, TEST, TRANSFER ALL или TRANSFER BOTH. Каждый объект имеет цепь заблокированных заявок, называемых цепью повторения. Любая заявка на цепи повторения ждет значения SNA, чтобы измениться. Когда значение SNA изменяется, повторно активируются все заявки в цепи повторения объекта. Это приводит к замене на CEC. Когда заявка становится активной, определенное тестовое условие повторяется. Так как этот процесс часто использует компьютерное время, не продвигая заявки в модели, неразумный выбор условий может привести к неэффективному моделированию. Возможности блоков GATE и TEST нужно использовать с осторожностью, так как заявки в цепях повторения заменены CEC процессом повторной активации. Это обсуждено ниже. Если при повторной попытке одна из проверок завершается успешно, заявка входит в следующий блок. Когда заявка входит в блок, она автоматически удаляется из всех цепей повторения.

Цепи оборудования

Каждая цепь оборудования имеет четыре цепи заявок:

- PENDING CHAIN – список заявок, ожидающих выполнения команды PREEMPT для оборудования в "режиме прерывания".
- INTERRUPT CHAIN – список заявок, которые переняли владение этим оборудованием.
- DELAY CHAIN – приоритетная цепь заявок, ожидающих владения этим оборудованием.
- RETRY CHAIN – список заявок, ожидающих изменения состояния оборудования.

Цепи объектов хранения

Каждый объект хранения имеет две цепи заявок. Эти цепи представляют собой связанные списки заявок:

- DELAY CHAIN – приоритетная цепь заявок, ожидающих сохранения.
- RETRY CHAIN – список заявок, ожидающих изменения статуса объекта хранения.

Цепь ожидания содержит заявки, ожидающие входа в блок PREEMPT режима прерывания. Транзакция, которая пытается войти в блок PREEMPT режима прерывания от имени оборудования, не получает доступа, если в цепи прерывания данного оборудования находятся резервированные заявки. Транзакция, которой отказано в доступе, переводится в цепь ожидания оборудования. Это останавливает транзакцию в процессе симуляции. Когда заявка перестает занимать оборудование, первая заявка из цепи ожидания получает контроль над оборудованием и доступ в блок PREEMPT.

Цепь прерывания - это список зарезервированных заявок. Если заявка входит в блок PREEMPT, а оборудованием распоряжается другая заявка, то контроль над оборудованием получает новая заявка. Старая заявка перемещается в цепь прерывания, чтобы восстановить контроль позднее. Заявки одной или нескольких цепей прерывания могут перемещаться в моделировании, но их движение ограничено. Такая заявка не может входить в FEC и не может покинуть блоки ASSEMBLE, GATHER, или MATCH Block. Когда заявка отдает контроль над оборудованием, если цепь ожидания пуста, первая заявка из цепи прерывания получает контроль над оборудованием.

Цепь задержки содержит заявки, ожидающие единиц хранения. Когда заявка пытается войти в блок SEIZE от имени используемого объекта, ей отказывается в доступе в блоке SEIZE и она перемещается в цепь задержки в порядке приоритета. Точно так же, если заявка пытается войти в блок PREEMPT в режиме приоритета от имени оборудования

(используемого заявкой равного или большего приоритета), ей отказывается в доступе в блок PREEMPT и она перемещается в цепь задержки оборудования в порядке приоритета. Это останавливает транзакцию в активной модели и выбирает новую активную транзакцию. Далее, когда заявка отдает контроль над оборудованием, и цепи ожидания и прерывания пусты, контроль над оборудованием отдаётся заявке высшего приоритета из цепи задержки.

Цепь повторения – это список заявок, ожидающих изменения статуса оборудования. Эти заявки активируются повторно, когда оборудование переходит из одного статуса в другой. заявки, ожидающие в цепи задержки, ожидания или прерывания, или имеющие контроль над оборудованием "конфликтуют" из-за оборудования. Так как конфликтующая заявка станет владельцем оборудования, условием разрешения конфликта является передача контроля над оборудованием. Если заявка, владеющая оборудованием, попытается выйти из симуляции входом в блок TERMINATE или ASSEMBLE, возникает ошибка «Stop». В то же время, резервированная заявка может покинуть симуляцию. Обычно каждая заявка остается в конфликте до момента, пока она самостоятельно не войдёт в блок RELEASE или RETURN от имени оборудования. Тем не менее, блоки PREEMPT и FUNAVAIL имеют параметры, удаляющие другие заявки из конфликта за оборудование. Это также освобождает от обязательства возвращения контроля. Действительно, не конфликтующая заявка, пытающаяся войти в блок RETURN или RELEASE вызовет ошибку «Stop».

Таким образом, если оборудование освобождается заявкой, заявки блока PREEMPT в режиме прерывания первыми получают контроль над оборудованием. Далее идут предварительно зарезервированные заявки цепи и заявки, ожидающие в цепи ожидания в порядке приоритета. Когда новая заявка выбирается из цепи задержки или ожидания, она немедленно входит в блоки SEIZE или PREEMPT, а затем размещается в CEC позади приоритетных одноранговых узлов.

Пользовательские цепочки

Каждый объект Userchain содержит операционную цепь, названную пользовательской цепью. Для более детального объяснения типа объекта Userchain, пожалуйста обратитесь к главе 4. Здесь мы обсуждаем операционную цепь, названную пользовательской цепью, которая содержится в каждом юридическом лице Userchain. Пользовательские цепи связаны списками заявок, которые были удалены от цепи текущих событий блоком связи. Традиционно, было два применения для пользовательских цепей. Сначала возможно осуществить чрезвычайно сложную транзакцию, намечающую дисциплины с пользовательскими цепями. Это может быть сделано, назначая числовой порядок на операционный параметр перед выполнением команды CONNECT заявки на пользовательской цепи. Во-вторых, более ранняя версия GPSS предполагает, что пользовательские цепи используются, чтобы избежать неэффективности в процессоре GPSS. Это менее верно в GPSS World, потому что заблокированные заявки не остаются на CEC в GPSS World. Однако, еще более эффективно можно этого избежать проверяя условия, которые не могут привести к успешному тесту. В этом случае, Вы можете поместить заблокированную транзакцию в пользовательскую цепь, пока она не будет обработана.

9.2 Планировщик заявок

Удобно представить моделирование GPSS как ряд заявок, которые занимают блоки в блок-схеме. И окно входа блока и окно блоков - по существу блок-схемы. В любой момент, каждая заявка находится точно в одном блоке, но большинство блоков может содержать любое число заявок. Каждая заявка, в свою очередь, получает возможность переместиться согласно предписанному пути через блок-схему. Когда заявкам отказывают во входе в блок, она должна ждать его в текущем блоке, пока условия не станут благоприятными для его перемещения. Часть GPSS World, которая является ответственной за это перемещение, называют операционным планировщиком. Каждый

тип блока имеет его собственную задачу, которая выполняется, когда заявка пытается войти в определенный тип блока. Работа планировщика заявок и состоит в том, чтобы выбрать соответствующую задачу. Первая вещь, которую делает операционный планировщик – идентификация "активной заявки". Если СЕС не пуста, заявка с самым высоким приоритетом на СЕС становится активной заявкой. Если СЕС пуста, операционный планировщик оперирует с СЕС заявками из FЕС с самым низким значением времени. Это действие также обновляет часы системы.

Операционный планировщик пробует переместить активную транзакцию в симуляцию, насколько это возможно. В действительности, операционный планировщик удаляет активную транзакцию из СЕС, определяет задачу для следующего последовательного блока (NSB), и если происходит неожиданные события, заменяет транзакцию перед ее одноранговыми узлами (то есть тот же самый приоритет) на СЕС. Это дает шанс заявкам с более высоким приоритетом шанс переместиться в имитацию. Замена СЕС может быть изменена приоритетом и буферными блоками. После того, как заявка входит в буферный блок, она ставится позади своих одноранговых узлов в СЕС. Буферные блоки полезны, если активированная заявка должна обогнать транзакцию, которая активировала ее. Другие блоки могут столкнуться с заменой заявки на СЕС. Например, ADVANCE (+) (то есть положительное приращение времени) вычисляет запланированное время и помещает транзакцию в FЕС. Другие блоки, типа LINK, ENTER, SEIZE и PREEMPT, могут запустить активную транзакцию, чтобы опираться на операционную цепь. Удаление или замена СЕС не имеет никакого влияния на часы системы. Моделируемое время остается тем же самым, пока нет никаких заявок, оставленных на СЕС. Непрерывная замена активной заявки на СЕС дает недавно активированным заявкам с более высоким приоритетом в СЕС шанс стать активной заявкой. Когда активная заявка запускается для перемещения в цепь задержки или не может переместиться из-за некоторого другого условия, операционный планировщик выбирает другую активную транзакцию и пытается переместить её в симуляцию.

Движение заявок

Заявки должны быть в СЕС, чтобы переместиться. Даже RESERVERD или REPLACED заявка должна стать активной заявкой прежде, чем сделать попытку входа в их новый блок предназначения. Так как заявки можно отказать во входе в блок, оперативный планировщик, возможно, не приводит к элементу блока. Поэтому большинство моделирований имеет меньше блоков, чем операционные планирования. С другой стороны, блоки EXECUTE могут потребовать дополнительные элементы блока. Когда активная заявка пытается войти в блок, операционный планировщик называет задачу блока связанной со следующим типом блока. Это - задача блока, который решает, действительно ли заявка может войти в блок. Несколько типов блоков могут отказать заявки в доступе. Этими типами блоков являются: ENTER, SEIZE, UNLOAD, GATE, TEST. Кроме того, если заявка не очистила все ее выгрузки, ей будет отказано в доступе в блоки ADVANCE (+) и не будет позволяться покинуть блоки ASSEMBLE, GATHER или MATCH.

Когда активная заявка не может войти ни в какой блок, это называют "приостановкой" в пределах симуляции. Тогда она удаляется из СЕС и помещается в одну из операционных цепей, обсужденных выше. Активной заявкой тогда назначается другая заявка по выбору.

Блокировка и восстановление

Активная заявка "блокирована", когда она ожидает изменения состояния одного или более объектов. Блоки GATE, TEST и TRANSFER, ALL могут требовать, чтобы определенные условия были выполнены в одном или более объектах прежде, чем активной заявки позволяют продолжить движение в модели. Каждый объект имеет цепь повторения для заявок, которые были блокированы, пробуя войти в один из вышеупомянутых блоков GPSS. Когда статус объекта будет изменен некоторой другой заявкой, все заявки на связанной цепи повторения будут помещены в цепи текущих событий позади их

приоритетных одноранговых узлов. Активация - движение заблокированных заявок СЕС. Если активная заявка изменяет статус объекта, возможно, что одна или более заявок будут повторно активированы прежде, чем активная заявка попытается войти в ее следующий блок. Если заявка более высокого приоритета будет повторно активирована, то станет активной заявкой. Если Вы желаете, чтобы недавно активированная заявка выполнялась немедленно, Вы должны или поместить активную транзакцию в СЕС позади ее узлов приоритета (BUFFER или блок PRIORITY, настройка BU), или Вы должны назначить активированной заявке более высокий приоритет, чем у старой активной заявки. Когда повторно активированная заявка становится активной заявкой, исходный блокирующий тест повторяется. Транзакции нельзя блокировать однократным тестом. Это приведет к ошибке **«Stop»**. Если объект не раз изменял статус перед обновлением часов системы, некоторые статусы не могут быть обнаружены. Это может случиться, при измене статус объекта дважды прежде, чем приостановленная заблокированная заявка проверяет условие. Обычно, эта возможность может быть исключена осторожным использованием блока BUFFER. Не используйте BUFFER или блоки GATE в режиме отказа, или TRANSFER (BOTH или ALL), чтобы блокировать пользовательские переменные. Транзакции не могут быть заблокированы на названных значениях, потому что последние не имеют цепи повторения.

9.3 Синхронизация

Наборы сборки

Набор сборки – это собранные заявки. Транзакции одного набора называются связанными. При создании каждой заявки ей присваивается целое число, определяющее её набор сборки. заявки, созданные блоком GENERATE, получают целые числа начиная с 1. заявки, созданные блоком SPLIT, получают набор сборки от родительского элемента.

Транзакция может изменить ее набор, устанавливаемый при входе в блок ADOPT. Наборы сборки полезны для того, чтобы вызвать синхронизацию среди заявок. Легко создать, ожидать, и удалять связанные заявки в моделировании. Это облегчит представление процессов, которые в некоторый момент должны ждать определенных условий, чтобы произойти. Следующие блоки GPSS используются с этой целью:

- ADOPT – установка набора сборки заявки.
- ASSEMBLE – ожидание и разрушение связанных заявок.
- GATHER – ожидания связанных заявок.
- MATCH – ожидание, пока связанные заявки не достигнут блока MATCH.
- SPLIT – создание связанных заявок.

Глава 10 – Советы по увеличению производительности

Эта глава содержит несколько советов, которые могут улучшить производительность вашей модели. Отладка и устранение неполадок обсуждаются в главе 13.

Имитация всегда происходит медленно, если в ней открыто несколько окон. Если скорость важна, закрывайте динамические окна сразу, как только это станет возможно.

GPSS World разработан для многозадачной работы. В то же время переключение между задачами иногда может вызвать задержки в обновлении окон. Этого можно избежать, открыв меньше окон, приостановив имитацию перед открытием различных видов окон или запуская только одну имитацию.

10.1 Выделение памяти

Никакие распределения объектов не требуются в GPSS, а являются автоматическими. По этой причине выражение **REALLOCATE** отсутствует. Так как GPSS использует виртуальную память, Вы можете использовать больше памяти для ваших моделирований чем существует в вашем компьютере. Это заставляет дополнительно обращаться к диску. Если Вы находите, что ваши моделирования из-за этого занимают больше времени, Вы можете исправить ситуацию, выделяя больше физической памяти вашему компьютеру. Для безопасности, количество запросов к виртуальной памяти, которые происходят в течение моделирования, ограничено. Это предотвращает случайный доступ к большому количеству виртуальной памяти, вызывающей замедление работы. Для изменения этого предела перейдите на страницу «Симуляция» вкладки настроек модели.

10.2 Определение точек перегруженности

Общая причина проблем работы - неограниченное создание заявок в вашем моделировании. Может быть полезно рассмотреть моделирование через каждое из графических окон. Это будет показывать Вам причину замедления вашего моделирования. Изучите окно «Blocks» на предмет наличия пунктов накопления заявок. Смотрите на представление деталей каждого из других окон объекта и проверьте размер цепи повторения. Это определяет число заявок, заблокированных на изменении статуса каждого объекта. Вы можете вывести стандартный отчет для изучения статуса моделирования. Чтобы вывести GPSS сообщение, напечатайте: **REPORT**

10.3 Советы по работе

1. Моделирования проходят НАМНОГО медленнее, когда открыто одно или несколько динамических окон. Для обновления динамических окон требуется отправка большого количества сообщений. Вы можете сильно ускорить работу, закрывая все окна онлайн.
2. Моделирования без ввода / вывода потока данных выполняются быстрее, когда управляют одновременно только одной симуляцией. Это позволяет избежать переключения работы.
3. Если работа важна, убедитесь, что компьютер имеет аппаратный математический сопроцессор. Попробуйте найти сопроцессор, который работает на больших частотах чем тот, что существует на материнской плате вашего персонального компьютера.
4. Если все эти методы не в состоянии улучшить работу, Вы должны рассмотреть возможность модернизации центрального процессора. Убедитесь, что поддерживаются возможности сопроцессора. Для всех продуктов GPSS выгодна симметрическая мультиобработка. Варианты Intel оптимизированы дляэ этого.

10.4 Советы по построению эксперимента

Следующие изменения ваших моделей могут ускорить управление ваших моделей.

1. Если ваше моделирование ведет себя неожиданно, Вы должны **ОСТАНОВИТЬ** моделирование и определить, растут ли заявки где-нибудь. Начните с окна блоков и ищите красные блоки, которые указывают условие блокирования. Больше информации доступно в детальном представлении. Начните пошагово проходить моделирование, таким образом Вы сможете видеть динамику операционного потока. Откройте детальное представление каждого типа окон и ищите большие цепи повторения. Это укажет неудовлетворенное условие блокирования.
2. Проблема работы может быть вызвана повторным тестированием заблокированных заявок. Много компьютерного времени будет потрачено впустую при неудачном испытании. Когда заявка заблокирована, объект моделирования помещает ее в одну или более цепей повторения так, чтобы «перетест» мог произойти, когда условия изменяются. Если Вы используете блоки GATE, TEST, TRANSFER BOTH или TRANSFER ALL и есть большие цепи повторения, есть все показания к тому, что много компьютерного времени будет проведено в неудачных тестах. Вы должны устроить вашу модель таким образом, чтобы минимизировать неудачное испытание в этих блоках.
3. Если Вы имеете заявки, ждущие определенного условия и нет никакого шанса на успешный тест, более эффективно поместить заблокированные заявки в пользовательскую цепь, пока нет возможности успешного исполнения условия.
4. GPSS - мощный язык моделирования в отличие от ФОРТРАНА и других языков программирования. заявки не должны быть **СГЕНЕРИРОВАНЫ** каждую секунду. GPSS намечает будущие события и открывает заявки автоматически.
5. Замените блоки режима отказа GATE и TEST на блоки с дополнительными конструкциями. Если это невозможно, измените условия, которые изменяются нечасто.
6. Используйте выражение **ФУНКЦИИ** GPSS, чтобы определить распределения вероятности вместо использования сложных функций библиотеки выражения.
7. Блоки COUNT и SELECT могут произвести большое количество тестов. Пробуйте заменить их менее мощными блоками, близко подобранными для вашей модели.
8. Вы можете вызвать объект моделирования, используя календарь «одного дня» для внедрения будущей цепи событий. Эта настройка может улучшить работу моделирования с FEC
9. Вы можете отложить блок смежных операционных чисел параметра и вызвать объект моделирования сразу определить их и получить доступ к ним прямой адресацией. Это устраняет последовательный поиск операционных параметров. Такие распределения называют блоками параметра и определяются во вкладке «моделирование» настройки модели, и обсуждаются в главе 2. Вы можете использовать команду EQU, чтобы заставить именованные параметры существовать в блоке параметра.
10. Блоки потока данных могут быть дорогими в вычислительном отношении. Вы можете ускорить обработку при использовании большого количества данных относительно небольшого количества строк текста. Например, используйте строки текста, которые содержат много пунктов данных. Вы можете разобрать эти структуры, используя процедуры строки в библиотеке процедуры.
11. Вы должны держать операционные приоритеты смежными. Используйте цифры 1, 2, 3, а не 12789,-30977, 22. Это ускоряет опережение, связанное с очередями приоритета. Ноль – отсутствие приоритета.
12. За счет некоторой интерактивности, счет опроса моделирования может быть увеличен, чтобы уменьшить опрос наверху в объекте моделирования. Для этого введите большее значение на странице «моделирование» вкладки настройки.
13. Интеграция происходит медленно. Если Вы знаете аналитическое решение обычного отличительного уравнения, используйте это в FVARIABLE, или PLUS-процедуре. Это

намного быстрее чем фактическое завершение интеграции в цифровой форме. Используйте одно или более интегрированных выражений.

Глава 11 – Стандартные отчеты

GPSS World предоставляет возможности богатых статистических отчетов, позволяя вам получать результаты без особых усилий. Основные состояния всех традиционных объектов GPSS отражаются в стандартном отчете, создаваемом автоматически по завершении моделирования.

Если вам требуются дополнительные возможности отчетов, ознакомьтесь с обсуждением потоков данных в главе 4.

11.1 Управление отчетами

Обычно GPSS создает детальный стандартный отчет для каждой имитации. Каждый объект сообщения автоматически создается, если Вы не определяете именную группу, как операнд В команды START. Вы можете также подавить создание стандартных сообщений, используя параметры настройки. Чтобы сделать так, убедитесь, что флажок **Create Standard Reports** не отмечен на странице «сообщения» вкладки параметров настройки. Команды REPORT и START обсуждены в главе 6.

11.1.1 Окна отчетов

Вы можете настроить создание объекта отчета при каждом появлении отчета. Для этого установите флажок **Create Standard Reports** и **In Windows** на странице «Отчет» блокнота настроек модели.

11.1.2 Управление содержимым отчета

Вы можете выбрать содержание стандартных сообщений модели, используя ряд флажков на странице «Отчет» блокнота параметров настройки модели. Для этого выберите меню **редактирование / параметры настройки** из главного меню. Это обсуждено в главе 2. окно отчета имеет возможности редактирования текста, изменения шрифта, цвета, и печати. Это обсуждено в главе 2 в разделе «Текстовые окна».

11.2 Пример отчета

11.2.1 Модель

Файл модели, который использовался для получения типового сообщения в следующем разделе. Когда этот образцовый файл, названный SAMPLE9. GP, транслирован и управляется, это производит автоматическое сообщение по имени SAMPLE9.001. Единственная цель модели состоит в том, чтобы показать всю различную графику и окна текста в руководстве обучающей программы в главе 1, урок 11 и продемонстрировать расположение сообщения и значений различных частей сообщения в этой главе.

```
GPSS World Sample File - SAMPLE9.GPS Model to demo graphics windows
Pool STORAGE 400 ;Define Storage
Matrix1 MATRIX ,5,5 ;Define Matrix
Transit TABLE M1,200,200,20 ;Transit time in wait ; and process
```

```
GENERATE (Exponential(1,0,100))
JOIN   Maingrp   ;Xact joins grp called ; Maingrp
JOIN   Numgrp,9999 ;Add 9999 to Numeric
SAVEVALUE Addup+,1 ;Total of Transactions
```

```

ASSIGN Param_1,232 ;Assign Xact parameter
JOIN Numgrp,P$Param_1 ;Put value in Param1
LOGIC S Switch_1 ;Turn on a logic switch
MSAVEVALUE Matrix1,2,2,QA$Tot_Process
QUEUE Tot_Process ;Queue for process time
SEIZE Facility1 ;Own first Facility
LINK Chain1,FIFO,Nxtblk ;Put on Userchain if busy
Nxtblk SEIZE Facility2 ;Own a second Facility
SEIZE Facility3 ;Own a third Facility
QUEUE Process_Time ;Keep track of process
ADVANCE 100,(Exponential(1,0,100))
DEPART Process_Time ;Record length
TABULATE Transit ;Add wait + process
RELEASE Facility1 ;Give up 1st Facility
ADVANCE 20 ;Delay time for Fac 2&3
RELEASE Facility2 ;Give up 2nd Facility
ADVANCE 10 ;Extra delay time-Fac 3
RELEASE Facility3 ;Give up 3rd Facility
DEPART Tot_Process ;Leave Queue
UNLINK Chain1,Nxtblk ;Take off waiting Xacts
ENTER Pool,100 ;Place 100 units in the Storage
LOGIC R Switch_1 ;Turn off logic switch
LEAVE Pool,50 ;Take 50 units from Storage
SAVEVALUE Collect-,1 ;Show negative Savevalue
REMOVE Maingrp ;Remove Xact from group
ADVANCE 50,1 ;Wait 50(+/-1) time units
LEAVE Pool,50 ;Take 50 units from Storage
Finis TERMINATE 1 ;Destroy Xact

```

11.2.2 Отчет

Этот раздел содержит стандартный отчет, сгенерированный программой, описанной в разделе 11.2.1. Каждый элемент отдельно объясняется в разделе 11.3. Рассмотрим элементы в составе стандартного отчета GPSS World. Этот раздел включает в себя полный отчет симуляции раздела 11.2.1.

GPSS World Simulation Report - SAMPLE9.1.1

Tuesday June 6, 2000 14:00:59

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	5187.692	32	3	1

NAME	VALUE
ADDUP	10007.000
CHAIN1	10012.000
COLLECT	10017.000
FACILITY1	10011.000
FACILITY2	10014.000
FACILITY3	10015.000
FINIS	32.000
MAINGRP	10005.000
MATRIX1	10003.000

NUMGRP	10006.000					
NXTBLK	12.000					
PARAM_1	10008.000					
POOL	10002.000					
PROCESS_TIME	10016.000					
SWITCH_1	10009.000					
TOT_PROCESS	10010.000					
TRANSIT	10004.000					
LABEL LOC BLOCK TYPE ENTRY COUNT CURRENT COUNT RETRY						
1	GENERATE	61	1	0		
2	JOIN	60	0	0		
3	JOIN	60	0	0		
4	SAVEVALUE	60	0	0		
5	ASSIGN	60	0	0		
6	JOIN	60	0	0		
7	LOGIC	60	0	0		
8	MSAVEVALUE	60	0	0		
9	QUEUE	60	9	0		
10	SEIZE	51	0	0		
11	LINK	51	0	0		
NXTBLK 12	SEIZE	51	0	0		
13	SEIZE	51	0	0		
14	QUEUE	51	0	0		
15	ADVANCE	51	1	0		
16	DEPART	50	0	0		
17	TABULATE	50	0	0		
18	RELEASE	50	0	0		
19	ADVANCE	50	0	0		
20	RELEASE	50	0	0		
21	ADVANCE	50	0	0		
22	RELEASE	50	0	0		
23	DEPART	50	0	0		
24	UNLINK	50	0	0		
25	ENTER	50	0	0		
26	LOGIC	50	0	0		
27	LEAVE	50	0	0		
28	SAVEVALUE	50	0	0		
29	REMOVE	50	0	0		
30	ADVANCE	50	0	0		
31	LEAVE	50	0	0		
FINIS 32	TERMINATE	50	0	0		

FACILITY ENTRIES UTIL. AVE. TIME AVAIL. OWNER PEND INTER RETRY DELAY									
FACILITY1	51	0.937	95.278	1	51	0	0	0	9
FACILITY2	51	0.853	86.719	1	51	0	0	0	0
FACILITY3	51	0.949	96.523	1	51	0	0	0	0

QUEUE MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME AVE.(-0) RETRY									
TOT_PROCESS	11	10	60	0	6.439	556.700	556.700	0	
PROCESS_TIME	1	1	51	0	0.660	67.112	67.112	0	

STORAGE CAP. REM. MIN. MAX. ENTRIES AVL. AVE.C. UTIL. RETRY DELAY
 POOL 400 400 0 150 5000 1 23.628 0.059 0 0

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM.%
TRANSIT	553.184	307.992				
			0- 200.000	8	16.00	
	200.000 - 400.000	10	36.00			
	400.000 - 600.000	7	50.00			
	600.000 - 800.000	12	74.00			
	800.000 - 1000.000	11	96.00			
	1000.000 - 1200.000	2	100.00			

USER CHAIN	SIZE	RETRY	AVE.CONT	ENTRIES	MAX	AVE.TIME
CHAIN1	0	0	0.277	48	1	29.927

XACT GROUP	GROUP	SIZE	RETRY
MAINGRP	10	0	

NUMERIC GROUP	GROUP	SIZE	RETRY
NUMGRP	2	0	

LOGICSWITCH VALUE RETRY
 SWITCH_1 1 0

SAVEVALUE RETRY VALUE
 ADDUP 0 60.000
 COLLECT 0 -50.000

MATRIX RETRY INDICES NUMERIC VALUE
 MATRIX1 0

Intermediate Values are Zero.
 (2,2) 6.413
 Intermediate Values are Zero.

CEC XN	PRI	M1	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
61	0	5187.692	61	1	2		

FEC XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
51	0	5389.554	51	15	16		
						PARAM_1	232.000
62	0	5523.253	62	0	1		

Этот раздел объясняет элементы стандартного отчета GPSS World.

Заголовок

GPSS World Simulation Report - SAMPLE9.1.1

Tuesday June 6, 2000 14:00:59

Срока заголовка стандартного отчета формируется из имени файла модели, создавшего отчет. Также включается дата и время запуска модели.

Общая информация

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	5187.692	32	3	1

- **START TIME.** Абсолютное значение системного времени в начале периода измерения. Значение **START TIME** задается равным системным часам через выражение **RESET** или **CLEAR**.
- **END TIME.** Абсолютное значение системного времени при достижении значения прерывания 0.
- **BLOCKS.** Количество блочных объектов в симуляции в конце симуляции.
- **FACILITIES.** Количество объектов оборудования в симуляции в конце симуляции.
- **STORAGES.** Количество объектов хранения в симуляции в конце симуляции.

Имена

NAME	VALUE
ADDUP	10007.000
CHAIN1	10012.000
COLLECT	10017.000
FACILITY1	10011.000

- **NAME.** пользовательские имена, использованные в модели GPSS World со времени последней трансляции.
- **VALUE.** Числовое значение, присвоенное имени. Системные значения начинаются с 10000.

Блоки

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
1	GENERATE	61	1	0	
2	JOIN	60	0	0	
3	JOIN	60	0	0	
4	SAVEVALUE	60	0	0	
5	ASSIGN	60	0	0	

- **LABEL.** Численно-буквенное название блока (если есть).
- **LOC.** Числовая позиция блока в модели. «местоположение».
- **BLOCK TYPE.** Имя блока GPSS.
- **ENTRY COUNT.** Количество заявок, вошедших в этот блок с момента последнего выражения **RESET** и **CLEAR** или с момента последней трансляции.
- **CURRENT COUNT.** Количество заявок в этом блоке на конец симуляции.
- **RETRY.** Количество заявок, ожидающих определенного условия в зависимости от статуса блочного объекта.

Оборудование

FACILITY	ENTRIES	UTIL.	AVE.	TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
FACILITY1	51	0.937	95.278	1	51	0	0	0	9	
FACILITY2	51	0.853	86.719	1	51	0	0	0	0	
FACILITY3	51	0.949	96.523	1	51	0	0	0	0	

- **FACILITY.** Имя или номер записи оборудования.
- **ENTRIES.** Количество раз использования оборудования с момента последнего выражения **RESET** и **CLEAR** или с момента последней трансляции.
- **UTIL.** Единица симулированного времени с момента последнего измерения владения оборудованием.
- **AVE. TIME** Среднее время владения индивидуальными заявками во время периода измерения. Период измерения начинается с трансляции модели или при выполнении команды **RESET** или **CLEAR**.
- **AVAIL.** Состояние доступности записи оборудования в конце симуляции. 1 – доступно, 0 – недоступно.
- **OWNER.** Номер заявки, владеющей оборудованием. 0 – оборудование не имеет владельца.

- PEND. Номер заявок, ожидающих резервирования оборудования путем входа в режим прерывания блоков PREEMPT.
- INTER. Количество резервированных заявок для этого оборудования. Число заявок в цепочке прерывания.
- RETRY. Количество заявок, ожидающих определенного условия в зависимости от статуса объекта оборудования.
- DELAY. Количество заявок, ожидающих захвата оборудования. Эта цепочка также содержит заявки, ожидающие резервирования оборудования в режиме приоритета блоков PREEMPT.

Очереди

```

QUEUE MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME AVE.(-0) RETRY
TOT_PROCESS 11 10 60 0 6.439 556.700 556.700 0
PROCESS_TIME 1 1 51 0 0.660 67.112 67.112 0

```

- QUEUE. Имя или номер записи очереди.
- MAX. Максимальное содержимое объекта очереди в течение периода измерения. Период измерения начинается с трансляции модели или при выполнении команды RESET или CLEAR.
- CONT. Текущее содержимое объекта очереди в конце имитации.
- ENTRY. Количество записей. Общее число записей очереди во время периода измерения.
- ENTRY(0). Количество нулевых записей. Общее число записей очереди со временем жизни 0.
- AVE.CONT. Средневзвешенное среднее единицы очереди во время периода измерения. результат «пространство-время», разделенный на продолжительность измерения.
- AVE.TIME. Среднее время на единицу содержимого очереди, использованного при измерении. результат «пространство-время», разделенный на общее количество записей.
- AVE.(-0). Среднее время на единицу очереди, использованное за период измерения, с поправкой на «нулевые записи». Продукт пространство-время, разделенный на (общее число записей минус число нулевых записей).
- RETRY. Количество заявок, ожидающих определенного условия в зависимости от статуса объекта очереди.

Объекты хранения

```

STORAGE CAP. REM. MIN. MAX. ENTRIES AVL. AVE.C. UTIL. RETRY DELAY
POOL 400 400 0 150 5000 1 23.628 0.059 0 0

```

- STORAGE. Имя или номер объекта хранения.
- CAP. Ёмкость хранения для объекта хранения, определенного в выражении STORAGE.
- REM. Количество неиспользованных единиц хранения в конце имитации.
- MIN. Минимальное количество единиц хранения, использованных за период измерения. Период измерения начинается с трансляции модели или при выполнении команды RESET или CLEAR.
- MAX. Максимальное количество единиц хранения, использованных за период измерения.
- ENTRIES. Количество «записей» в объекте хранения за период измерения. Общее накопление операнда В выражения ENTER.
- AVL. Состояние доступности записи хранения в конце симуляции. 1 – доступно, 0 – недоступно.

- AVE.CONT. Средневзвешенное среднее единицы хранения во время периода измерения. результат «пространство-время», разделенный на продолжительность измерения.
- UTIL. Доля всего продукта пространство-время для объекта хранения во время периода измерения.
- RETRY. Количество заявок, ожидающих определенного условия в зависимости от статуса объекта хранения.
- DELAY. Количество заявок, ожидающих входа в блоки ENTER от данной единицы хранения.

Таблицы и Q-таблицы

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM.%
TRANSIT	553.184	307.992				
		200.000 - 400.000	10	36.00		
		400.000 - 600.000	7	50.00		
		600.000 - 800.000	12	74.00		
		800.000 - 1000.000	11	96.00		
		1000.000 - 1200.000	2	100.00		

- TABLE. Имя или номер единицы Table или Qtable.
- MEAN. Взвешенное арифметическое среднее значений через табуляцию.
- STD.DEV. Взвешенное стандартное примерное отклонение значений. $S.D. = \sqrt{SOS / (COUNT - 1) - (SUM2 / (COUNT)(COUNT - 1))}$) Где SOS – накопленная сумма квадратов.
- RANGE. Нижний и верхний пределы описываемого класса частотности. Значения аргумента таблицы, большие, чем нижний предел и меньшие или равные верхнему лимиту обновляют этот класс. Операнд В выражения TABULATE используется для предоставления взвешенного фактора. Классы частотности с величинами, равными нулю, в отчёт не включаются.
- RETRY. Количество заявок, ожидающих определенного условия в зависимости от статуса табличного объекта.
- FREQUENCY. Общее взвешенное количество значений через табуляцию, попадающих в этот диапазон. Общее количество операндов В TABULATE.
- CUM.% Кумулятивное количество частотности, выраженное в процентах от общего количества..

Пользовательские цепочки

USER CHAIN	SIZE	RETRY	AVE.CONT	ENTRIES	MAX	AVE.TIME
CHAIN1	0	0	0.277	48	1	29.927

- USER CHAIN. Имя или номер записи пользовательской цепочки.
- SIZE. Количество заявок в пользовательской цепочке на конец периода измерения.
- RETRY. Количество заявок, ожидающих определенного условия в зависимости от статуса объекта пользовательской цепочки.
- AVE.CONT. Средневзвешенное среднее значение занятости пользовательской цепочки во время периода измерения. результат пространство-время, разделенный на продолжительность измерения.
- ENTRIES. Общее число заявок, сделанных в пользовательской цепочке за весь период измерения.
- MAX. Максимальное число заявок, сделанных в пользовательской цепочке за весь период измерения.
- AVE.TIME. Среднее время заявки в пользовательской цепочке за период измерения. результат пространство-время, разделенный на общее количество записей.

Группы заявок

XACT GROUP GROUP SIZE RETRY
MAINGRP 10 0

- XACT GROUP. Имя или номер объекта группы заявок.
- GROUP SIZE. Количество заявок, входящий в состав группы в конце имитации.
- RETRY. Количество заявок, ожидающих определенного условия в зависимости от статуса объекта группы заявок.

Числовые группы

NUMERIC GROUP GROUP SIZE RETRY
NUMGRP 2 0

- NUMERIC GROUP. Имя или номер объекта числовой группы.
- GROUP SIZE. Количество числовых значений, входящий в состав группы в конце имитации.
- RETRY. Количество заявок, ожидающих определенного условия в зависимости от статуса объекта числовой группы.

Логические переключатели

LOGICSWITCH VALUE RETRY
SWITCH_1 1 0

- LOGICSWITCH. Имя или номер объекта логического переключателя.
- VALUE. Значение объекта логического переключателя в конце симуляции. 1 – «установлено» или «истина», 0 – «сброшено» или «ложь».
- RETRY. Количество заявок, ожидающих определенного условия в зависимости от статуса объекта логического переключателя.

Значения сохранения

SAVEVALUE RETRY VALUE
ADDUP 0 60.000
COLLECT 0 -50.000

- SAVEVALUE. Имя или номер записи сохранения.
- VALUE. Значение объекта сохранения в конце симуляции.
- RETRY. Количество заявок, ожидающих определенного условия в зависимости от статуса объекта сохранения.

Объекты матрицы

MATRIX RETRY INDICES NUMERIC VALUE
MATRIX1 0

Intermediate Values are Zero.
(2,2) 6.413

Intermediate Values are Zero.

- MATRIX. Имя или номер объекта матрицы.
- RETRY. Количество заявок, ожидающих определенного условия в зависимости от статуса объекта матрицы.
- INDICES. До 6 целых чисел, определяющих элемент матрицы.
- VALUE. Значение элемента объекта матрицы в конце симуляции. Элементы со значением 0 отображаются в группах.

Цепочка текущих событий

CEC XN PRI M1 ASSEM CURRENT NEXT PARAMETER VALUE
61 0 5187.692 61 1 2

- XN. Номер заявки каждого элемента цепочки текущих событий.
- PRI. Приоритет планирования заявки.
- M1. Время отметки. Время генерации заявки или самого раннего родителя, или время заявки по вводу блока MARK без операндов.
- ASSEM. Номер набора сборки для заявки.

- CURRENT. Номер блока, в котором заявка существовала до конца симуляции или на момент отчета.
- NEXT. Номер следующего блока, запланированного ко вводу заявкой.
- PARAMETER. Имена или номера параметров заявки.
- VALUE. Значение параметра.

Цепочка будущих событий

FEC XN PRI BDT ASSEM CURRENT NEXT PARAMETER VALUE

51 0 5389.554 51 15 16

PARAM_1 232.000

62 0 5523.253 62 0 1

- XN. Номер заявки каждого элемента цепочки будущих событий.
- PRI. Приоритет планирования заявки.
- BDT. Время отправки блока. Время абсолютных системных часов, на которое назначена отправка блока из цепочки будущих событий.
- ASSEM. Номер набора сборки для заявки.
- CURRENT. Номер блока, в котором заявка существовала до конца имитации или на момент отчета.
- NEXT. Номер следующего блока, запланированного ко вводу заявкой.
- PARAMETER. Имена или номера параметров заявки.
- VALUE. Значение параметра.

Глава 12 – Статистика GPSS World

12.1 Введение

GPSS World имеет широкий инструментарий сбора статистики. Для несложных симуляций статистика стандартного отчёта и статистика библиотеки ANOVA удовлетворит большинство пользователей. В этих случаях GPSS World собирает и отчитывается о статистике автоматически. Это рассмотрено в главах 11 и 8.

Если требуется использование более детальной статистики, используйте переменные объекты и процедуры PLUS. В этих случаях вся сила математических функций библиотеки в вашем распоряжении. Статистические таблицы определяются в выражениях TABLE или QTABLE. Собственно запись информации вызывается одним или несколькими блоками имитации TABULATE. Каждый раз, когда вводится команда TABULATE, данная величина регистрируется в таблице. Команды QTABLE основываются на объектах очереди и представляют собой легкий путь для сбора статистики. В этом случае табуляции возникают путем входа в блок DEPART вместо блоков TABULATE.

Обычно, необходимо производить моделирование таким образом, чтобы значения управления были взяты от пользовательских переменных, и при окончании моделирования результаты находились в пользовательских переменных. Это происходит из-за особенностей работы созданных автоматическими генераторами экспериментов PLUS. Если вам все же придется использовать их, необходимо будет учесть эти ограничения.

Для всестороннего анализа можно накопить результаты моделирования при использовании команд Open, Close, Read, Write, Seek в форме блоков GPSS или вызовов процедуры. Статистика моделирований приводит к созданию базы данных, которая может использоваться для анализа. Для анализа данных в матрице результата, доступна встроенная многоканальная процедура ANOVA.

12.2 ANOVA (дисперсионный анализ)

Если Вы плохо знакомы с моделированием, то Вы заметите, что, когда Вы повторяете моделирование на той же самой сессии, Вы можете получить различные результаты из-за случайных эффектов. Такие эффекты нужно отличать от реальных эффектов, вызванных в соответствии с новыми проектами в ваших моделируемых системах. Процедура библиотеки ANOVA предоставляет Вам простое и разумное решение для того, чтобы установить, корректны ли ваши результаты или содержат вызванные отклонениями помехи.

Чтобы измерить количество случайного шума, Вы должны повторить моделирования, изменяя только определители случайных чисел. Эти повторные прогоны называются ответами. Они важны для измерения статистического шума, то есть стандартной ошибки в вашем эксперименте. В самой простой схеме каждый такой моделируемый проект должен повторяться несколько раз. Тогда процедура библиотеки ANOVA будет использовать данные, для того чтобы обнаружить присутствие эффектов, в значительной степени влияющих на изменение факторов эксперимента.

Процедура ANOVA предусматривает анализ данных нажатием одной кнопки. Она вычисляет доверительные интервалы и дисперсионный анализ результатов моделирования. Обычно, в процессе эксперимента, заполняется специальный глобальный объект матрицы, который называется матрицей результата с результатами моделирований, и затем передаете это как аргумент процедуре ANOVA. Детали использования многоканальной процедуры библиотеки ANOVA среды GPSS обсуждены подробно в следующей главе в разделе 13.5.3.

12.3 RESET

Если требуется исключить эффекты начальных условий из конечного моделирования, необходимо использовать команду RESET, чтобы начать период измерения после того, как исчезнут переходные эффекты. Команда RESET описана в главе 6.

12.4. Пространственно-временные продукты

Пространственно-временные продукты – это продолжительность времени, умноженная на инкремент. Пространственно-временные продукты используются в вычислении нескольких видов полезной статистики. Например, чтобы вычислить заработные платы рабочих, можно использовать число людей, которые работали каждое данное число часов. Для вычисления общее количество рабочих часов, можно умножить часы на количество людей, чтобы сформировать пространственно-временной продукт для каждого часа работы. Можно сложить все пространственно-временные продукты, для вычисления общее количество рабочих часов для каждого человека. Точно так же большинство статистических данных, о которых сообщают в стандартных сообщениях, являются накоплениями пространственно-временных продуктов. Устройство, очередь, память, и объекты Userchain имеют протоколы SNA, которым для оценки необходимы пространственно-временные продукты. Например, среднее количество используемой памяти – это среднее значение времен самой длительной продолжительности ее использования. Накопленный пространственно-временной продукт делится на времени моделирования для вычисления среднего количества используемой памяти. Устройство, очередь и объекты Userchain также имеют пространственно-временные продукты. Они используются, для вычисления средней занятости через соответствующий протокол SNA. Другие пространственно-временные продукты равны области под графом содержания очереди или содержания Userchain.

Команда RESET может использоваться, чтобы начать новый период измерения. Команда RESET заставляет пространственно-временные продукты обнуляться. Это позволяет накапливаться новым пространственно-временным продуктам. В протоколах SNA, основанных на вычислениях пространственно-временных продуктов, этот метод инициализации может внести небольшое смещение на низких уровнях.

Таблица 12—5. Системные числовые атрибуты пространство-время

SNA ENTITY CALCULATION

FR UTILIZATION (ppt)

FT AVERAGE HOLDING TIME PER CAPTURE

QA AVERAGE CONTENT

QT AVERAGE RESIDENCE TIME

QX AVERAGE RESIDENCE TIME EXCLUDING ZEROS

SA AVERAGE CONTENT

SR FRACTIONAL UTILIZATION

ST AVERAGE HOLDING TIME

CA AVERAGE CONTENT

CT AVERAGE RESIDENCE TIME

SNA ENTITY CALCULATION

Facility 1000 times (Space-time product) divided by (Total Simulated Time)

Facility (Space-time-product) divided by (Total capture count)

Queue (Space-time-product) divided by (Elapsed time)

Queue (Space-time-product) divided by (Total count)

Queue (Space-time product) divided by (Total Count)-(Count finding zero)

Storage (Space-time-product) divided by (Elapsed time)

Storage 1000 times (Space-time product) divided by (Capacity) times (Elapsed time)

Storage (Space-time-product) divided by (Sum of storage requirements)

Userchain (Space-time-product) divided by (Elapsed time)

Userchain (Space-time-product) divided by (Total count)

Глава 13 - Экспериментирование

Эта глава рассказывает об особенностях экспериментирования в среде GPSS, и их использовании. Общий краткий обзор представлен во введении главы, сопровождаемом фоновым материалом, затем изложено детализированное представление всех особенностей экспериментирования. Главу завершает список практических руководств по разрешению возможных страниц.

Однако, чтобы ознакомиться со всеми возможностями среды GPSS, ознакомьтесь с остальными частями главы. Также полезными окажутся несколько новых типовых моделей, включая OneWay.gps, Multiway.gps, ExperEther.gps, LatinSquare.gps, GraecoLatin.gps и уроки 19 и 20 в Обучающем руководстве к программе.

13.1 Введение

Экспериментальная фаза проекта моделирования основана на существовании полностью разработанной, проверенной системы моделирования в среде GPSS, которая отображает влияние каждого из факторов, эффекты которых должны быть измерены. Только в этом случае вы можете продолжать экспериментальную фазу вашего проекта. Концептуально, среда GPSS поддерживает три различных подхода к экспериментированию: эксперименты отбора, пользовательские эксперименты и эксперименты оптимизации. Есть особенности, которые отличают каждый из них.

Эксперименты, сгенерированные GPSS, экспериментируют с неподвижным числом уровней обработки для каждого фактора. Когда Вы создаете собственные эксперименты, Вы не имеете этого ограничения. Вы ограничены 6 факторами, каждый из которых может иметь любое число уровней обработки. Ваша задача состоит в определении эксперимента, заполнении матрицы результата результатом каждого прогона и передаче ее процедуре библиотеки ANOVA.

13.1.1 Эксперименты отбора

Эксперимент отбора обычно используется, чтобы идентифицировать самые важные факторы, затрагивающие моделируемую систему. Эта информация является критической для того, чтобы направить остальную часть исследования самым эффективным способом. Результаты отбора определяют, какие факторы не эффективны и должны получить низкий приоритет относительно дальнейшего исследования. Кроме того, чувствительность результатов эксперимента к одному или несколькими скрытым факторам даёт также может иметь значение. Среда GPSS содержит автоматический генератор эксперимента, который может создать эксперименты отбора. Для его использования необходимо заполнить параметры в окне меню редактирования основного окна. Это генерирует PLUS-код, вставляемый в объект модели. Этот процесс также может загрузить функциональную клавишу соответствующей командой CONDUCT для моделирования. После этого, вам остаётся создать объект модели (Ctrl+Alt+S), запустить эксперимент (обычно F11), и затем проанализировать результаты.

13.1.2 Пользовательские эксперименты

Пользовательские эксперименты более гибки, но их необходимо создавать и выполнять их напрямую. Даже в этом случае, когда среда GPSS обеспечивает большую поддержку в процессе работы. Необходимо задать входные параметры библиотечной процедуры ANOVA, которая анализирует эксперименты с 6 факторами и их взаимодействия. Основное требование ANOVA GPSS состоит в передаче названия матрицы GPSS, в которой были сохранены результаты эксперимента.

Есть две важные вещи, которые необходимо знать о матрицах результата. Перед началом эксперимента, необходимо инициализировать элементы матрицы в состояние UNSPECIFIED. Ниже приведено используемое для этого типовое выражение:

INITIAL MyResultMatrix,UNSPECIFIED

Оно поясняет подпрограмме ANOVA, что выполнение эксперимента не было закончено. Вторая вещь то, что, если эксперимент содержит более одного фактор, необходимо сделать их симметричными, чтобы подпрограмма среды GPSS ANOVA могла точно проанализировать дисперсию. Такая специальная симметрия обозначается техническим термином "ортогональность". Все, что необходимо сделать, для достижения ортогональности, это посмотреть на каждую пару факторов в эксперименте. Если, в пределах каждой пары, каждый уровень обработки первого фактора появится то же самое число раз, что и в пределах каждого уровня обработки второго фактора, то эксперимент будет ортогонален. Если все пары факторов будут симметричны, ортогональность будет достигнута. В случае ошибки, GPSS выведет сообщение. GPSS содержит язык PLUS, который можно использовать для написания программируемых экспериментов.

13.1.3 Оптимизация экспериментов

Оптимизация и количественное предсказание поведения системы - часто первичные цели проекта моделирования. Оба из них непосредственно поддерживаются в среде GPSS. Поверхность отклика – это уравнение, которое предсказывает результаты моделирования. Часто желательно установить поверхность отклика для того, чтобы сократить результаты, обеспечивая прогнозирующую методологию, определяя количество чувствительности результатов к числовым входам и другим факторам, и определяя оптимальные уровни обработки. GPSS обеспечивает основу для разнообразных методов нахождения поверхности откликов. PLUS эксперимент, который вычисляет поверхность отклика, Результирующий эксперимент может использовать метод наискорейшего подъёма и метод местного исследования для нахождения оптимальную значение. В случае успеха, выводится математическое описание наилучшей пригодной поверхности отклика и предсказанных оптимальных условий.

13.2 Экспериментирование и анализ вариантности

В этом разделе мы рассмотрим некоторые из технических вопросов, связанных с экспериментированием в среде GPSS. Здесь представлено обсуждение главных проблем, связанных с вызовом процедуры ANOVA или при проведении эксперимента.

13.2.1 Мотивация

Системы моделирования не полностью отвечают условиям реальных систем, которые они имитируют. Они - просто представления, предназначенные, чтобы для определения самых важных поведенческих особенности целевой системы. Главное отличие от реального миром заключается в том, что системы моделирования являются совершенно повторимыми. Таким образом, если мы повторяем моделирование много раз, мы каждый раз получаем одинаковые самые результаты. Такой результат крайне маловероятен в естественных науках. Фактически, это настолько неестественно, что мы преднамеренно вводим изменчивость в наши системы моделирования, чтобы сделать их более реалистичными. Практически, мы часто находим системы моделирования, в которых случайное изменение в смоделированных процессах, искусственно введенных нами, существенно влияет на поведение целевой системы. Другим источником изменчивости в системах моделирования являются непосредственные измерения. Принимая во внимание, что в реальном мире некоторые инструменты измерения являются очень поменеустойчивыми и неточными, в моделируемой среде, мы наслаждаемся высокой

точностью измерений, без влияния окружающей среды. Дело в том, что наблюдения от моделирований являются противостоестественно чистыми, и могут иногда показывать эффекты, которые существуют в реальном мире, но чрезвычайно трудны для наблюдения в нем. В отличие от экспериментов, проводимых в реальном мире, в моделировании мы имеем намного лучший контроль над изменчивостью, введенной, чтобы подражать имеющейся в реальном мире хаотичности. В общем, мы вводим один или более потоков псевдослучайных чисел, которые являются в состоянии передать определенные статистические параметры хаотичности. Более того, когда мы выбираем способ распределений вероятности, чтобы точно отразить целевое поведение системы. Среда GPSS предоставляет более двадцати способов, каждый из которых может быть отобран и использоваться аналитиком моделирования. В нашем количественном анализе результатов моделирования, мы предположим, что мы успешно смоделировали случайную значения, имеющиеся в реальном мире.

Дисперсионный анализ или ANOVA, являются высоко развитой методологией для того, чтобы извлечь информацию из результатов эксперимента. В основном, метод состоит в разбиении всех наблюдаемых отклонений на части, и связывании этих частей с экспериментальными факторами и их взаимодействиями. Одна часть, названная ошибочным сроком, связана с неизбежно свойственной наблюдениям хаотичностью. ANOVA пытается определить все отклонения наблюдений от их среднего числа. "Ошибками" является изменение, которое не удовлетворило условиями эксперимента. Такие ошибки свойственны оценке наблюдаемых отклонений, которая называется стандартной ошибкой. Обычно, только те эффекты, которые превышают величину стандартной ошибки, являются реальными, а не вызваны случайными колебаниями. Для установление этого факта обычно используется метод проверки "F Statistic". Необходимо всеми возможными способами отделять отклонения, введенные для реализма, от отклонений статистических моделей - стандартной ошибки. В общем случае, нельзя искажать прошлое, чтобы не вносить изменения в реальное поведение модели. С другой стороны, цель проекта экспериментов состоит в том, чтобы уменьшить неучтенные в нашем статистическом анализе отклонения. Наша цель состоит в том, чтобы моделировать хаотичность реального мира точно, не преуменьшая его. Мы, однако, с другой стороны хотим найти способы уменьшить изменчивость наблюдений в дисперсионном анализе, не изменяя способ, с помощью которого моделируется естественная хаотичность. Сокращение оценки стандартной ошибки в дисперсионном анализе позволяет существенным эффектам подниматься над статистическим уровнем шума. Два метода выполнения, которые мы обсудим ниже, улучшают остаточные данные, используемые в оценке стандартной ошибки: первый, увеличивая число наблюдений, второй - изменяя основную статистическую модель. Они обсуждены в Секции 13.2.3

13.2.2 Номенклатура

При проведении эксперимента, выбирается одна или более метрики, которые определяют количество состояний системы или некоторого другого интересующего результата. Эти наборы называют "наблюдениями" или "выходом", и набор всех наблюдений включает "результаты" эксперимента. Исследованию подлежат одно или несколько условий, называемых "факторами", которые, влияют на значение некоторых из наблюдений. Мы назначаем значения, называемые "уровнями обработки" к факторам, когда мы определяем условия для каждого прогона моделирования. При наличии нескольких факторов, мы говорим, что условия определены "комбинациями обработки", так как должны быть определены несколько уровней обработки. Если влияние некоторого фактора отличается, при изменении уровня обработки некоторого другого фактора, мы говорим, что есть "взаимодействие" между этими двумя факторами. Так как мы будем использовать совокупную модель, когда эффекты двух факторов вместе не являются суммой их

отдельных эффектов, мы говорим, что между ними есть взаимодействие. Могут быть отличные взаимодействия, вовлекающие любое число факторов. Каждая итерация моделирования целевой среды называется "выполнение". Комбинация обработки определяет условия выполнения, и одно или более наблюдений формируют результат выполнения. Когда условия ряда выполнений одинаковы, за исключением рандомизации, мы называем выполнения в этом наборе - "прогонами", а формы - "ячейками" эксперимента.

В ходе дисперсионного анализа, сделанного средой GPSS, наблюдения разделяются на компоненты, называемые "эффектами", которые являются результатом влияния факторов и их взаимодействий. То, как это разделение сделано, зависит от основной совокупной модели, названной "статистической моделью". Случайное отклонение заставляет наблюдения немного отличаться от суммы эффектов в статистической модели. "Вектор ошибок" наблюдения получается вычитанием оценки эффектов, соответствующих элементам статистической модели. Так как ошибка является количественным остатком, её часто называют "остаточной". В среде GPSS для завершения дисперсионного анализа многоканальных экспериментов, т.е. экспериментов с более чем одним фактором, требуется, чтобы эксперимент был ортогонален. Это означает, что параметр оценки в пределах анализа должны быть некоррелированными. Практически, обеспечение определенного числа запусков в пределах каждой комбинации обработки сбалансированного плана гарантирует ортогональность.

Эффекты имеют важное значение для расчета дробных факторных планов, которые будут обсуждаться ниже. Для определения количества экспериментов (выполнений) в дробном факторном плане, проектировщиком выбирается небольшое количество эффектов в качестве. К сожалению, вследствие этого, некоторые эффекты становятся неразличимы с другими ("aliased"). В GPSS набор генераторов, отделенных знаком равенства (=), называют "отношением определения".

GPSS поддерживает экспериментирование через внутренние процедуры библиотеки и через PLUS, встроенный в моделирование язык программирования. PLUS является и процедурным языком низкого уровня, доступным изнутри моделирования, и языком управления высокого уровня, который может определять условия и последовательность запусков в эксперименте.

13.2.3 ANOVA

Дисперсионный анализ - инструмент, созданный сэром Рональдом Фишером, который является в состоянии извлечь большую часть информации, доступной в ряде замеров. Мы определяем количество изменений наблюдений от полного среднего числа, и затем ломаем это на кусочки, каждый из которых имеет отдельную причину. Если какой-нибудь экспериментальный фактор не может быть найден, чтобы вызвать изменчивость в измерении, мы говорим, что это не имеет существенного эффекта на эксперимент. С другой стороны, если фактор действительно вызывает изменчивость, мы сравниваем количество этого с оценкой свойственной изменчивости наблюдения, стандартной ошибки. Мы делаем это, чтобы исключить очевидные эффекты, которые являются ни чем иным, как случайным колебанием. Наш стандарт сравнения то, что изменение из любого источника должно быть намного большим чем стандартная ошибка, чтобы считаться существенным эффектом. Тест F, названный по имени Фишера, используется с этой целью. Мы используем тест F, как критерий, в соответствии с которым мы объявляем, что эффекты экспериментальных факторов и их взаимодействий статистически существенные. Неявный в использовании ANOVA – это совокупность математических моделей, для описания изменяющихся компонентов в наблюдениях. Мы назовем это "статистической моделью".

Рассмотрим статистическую модель, используемую, чтобы проанализировать данные от эксперимента с единственным фактором, а именно, фактором А.

$$\psi_{i,\varphi} = \mu + \alpha_i + \varepsilon_{i,\varphi}$$

Рисунок 13.1 – Однофакторная статистическая модель

На рисунке 13.1 заметьте введение альфа-термина, который обозначает эффект i-уровня обработки. Только фактор, который рассматривают в модели, эксперимент включил бы в один или более запусков, в каждый из уровней обработки фактора А. Все наблюдения на данном уровне обработки проанализированы. Отсюда в этом эксперименте есть только один фактор, при этом число комбинаций обработки - только число уровней обработки того фактора. Дисперсионный анализ, основанный на этой статистической модели приведет к таблице ANOVA, которая делит изменение наблюдений на изменения, возникшие из-за обработки А, и из-за случайного изменения.

$$\psi_{i,\varphi,k} = \mu + \alpha_i + \beta_{\varphi} + \gamma(\alpha\beta)_{i,\varphi} + \varepsilon_{i,\varphi,k}$$

Рисунок 13.2 – Двухфакторная статистическая модель

На рисунке 13.2 мы рассматриваем модель, где участвуют два фактора. Заметьте, что мы теперь включаем срок для взаимодействия между фактором А и фактором В, обозначенный (ab). Этот срок может отличаться в каждой комбинации обработки, и поэтому подготовлен дважды, отсюда два фактора. Отличие от вышеупомянутого состоит в том, что, когда эффекты двух факторов вместе не сумма их отдельных эффектов, наблюдается взаимодействие между ними. Сила взаимодействия между факторами А и В обозначена "AB" и будет отражена в строках таблицы ANOVA. В полном факторном эксперименте были бы включены все сроки взаимодействия. Так же, как полная модель с 2 факторами на рисунке 13.2 имеет 5 уровней, полная модель с 3 факторами имеет 8 уровней, и полная модель с 4 факторами имеет 16 уровней. Вся эта информация обычно представляется в итоговой таблице ANOVA. GPSS может работать с моделями с 6 факторами, включая взаимодействия с 3 путями. Рисунок 13.3, ниже, представляет окно ANOVA, о котором сообщает GPSS. Как объяснено выше, заглавные буквы обозначают факторы, а комбинации заглавных букв обозначают взаимодействия. Сначала, обратимся к нижней части таблицы. Целая сумма квадратов должна быть разделена, и компоненты должны быть связаны с эффектами факторов и их взаимодействиями. То, что перенесено, то есть, остаточная сумма площадей, показывается в предыдущей строке, помеченной "ошибка". Средняя сумма квадратов ошибочного уровня используется, чтобы оценить стандартную ошибку эксперимента. К каждой сумме площадей относятся степени свободы. Исходя из статистических рассмотрений, степени свободы – это тот делитель, который должен использоваться, чтобы создать достоверную оценку стандартной ошибки, в отсутствии других эффектов. В нашем случае, достаточно представлять степени свободы как надлежащий делитель, связанный с суммой квадратов в таблице ANOVA. GPSS будет всегда вычислять степени свободы.

ANOVA					
Source of Variance	Sum of Squares	Degrees of Freedom	Mean Square	F	Critical Value of F (p=.05)
A	28.000	2	14.000	5.600	3.89
B	21.000	3	7.000	2.800	3.49
AB	11.000	6	1.833	0.733	3.00
Error	30.000	12	2.500		
Total	90.000	23			

Treatment Level A B	Count	Mean	Minimum	Maximum	95% C.I. (SE)
1 1	2	3.000	2.000	4.000	[0.564, 5.436]
1 2	2	1.000	1.000	1.000	[-1.436, 3.436]

01/27/01 13:26:44 1.5811388

Рисунок 13.3 –Таблица ANOVA

Каждый фактор и взаимодействие в статистической модели представлены разделительной линией в таблице ANOVA. В каждой линии мы имеем сначала сумму квадратов и степеней свободы, связанной с той оценкой. Они - основы, из которых получены другие числа. Простое деление приводит к среднему квадрату, и деля то частное на средний квадрат ошибки, от нижнего ряда таблицы, мы получаем статистическое F для того эффекта.

Теперь мы готовы сделать некоторые выводы. Мы должны решить, является ли значение F достаточно большим, чтобы объявить, что эффект существенен. Пороговое значение, которое мы будем использовать, чтобы сделать сравнение, называют "Критическим Значением F". Оно помещается только справа от нашего статистического F на той же самой строке. Если наше значение F превышает «Критическое Значение», мы заключаем, что мы имеем дело с существенным эффектом. В противном случае мы заключаем, что эффект не существен, и мы игнорируем любое связанное изменение в наблюдениях только из-за случайного шума. Большее значение F приводит к более сильному эффекту. Таблица ANOVA на рисунке 13.3 показывает существенность эффекта фактора A, и несущественность эффектов от фактора B и взаимодействия AB.

Иногда эксперимент будет не в состоянии обнаружить эффект даже при том, что μ фактически существует. Одна из наших целей состоит в том, чтобы сделать это настолько, насколько это возможно. Согласно таблице ANOVA, есть два способа сделать эксперимент более приспособленным обнаруживать реальные эффекты. Чтобы получить

лучшие результаты, мы нуждались бы или в большем статистическом F или в меньшем «Критическом Значении F ».

Желательно удалить любую часть суммы квадратов ошибки, которая происходит из-за любого важного эффекта, который мы не включили в анализ. Если мы можем сделать это, наше статистическое F вообще будет большим. В экспериментировании в естественных науках это применяется, делая сравнения в гомогенной среде насколько это возможно. Эта методика известна как "блокирование".

Однако, в учебном моделировании к этой цели лучше всего приблизиться, идентифицируя дополнительные факторы, которые должны быть включены в эксперимент. Два других подхода направлены на увеличение степеней свободы ошибочного уровня. Первое должно просто увеличить число прогонов эксперимента. Это - обычно самый дорогой подход, но это может быть весьма эффективным. Вторая возможность имеет отношение с дизайном эксперимента и статистической модели анализа.

13.2.4 Выбор факторов

Когда Вы используете процедуру библиотеки ANOVA, Вы должны определить матричный объект GPSS, названный матрицей результата, хранить индивидуальные результаты каждого выполнения. Если бы Вы анализировали больше чем один показатель, то Вы имели бы больше чем одну матрицу результата. Начиная с GPSS World матричные объекты могут иметь до 6 измерений, матрица результата также ограничена 6 измерениями. Когда Вы пишете ваш собственный эксперимент GPSS World, самое важное решение, которое Вы сделаете, состоит в том, чтобы выбрать факторы эксперимента.

Они - величины, которыми Вы можете управлять, чтобы оптимизировать вашу систему. Каждый фактор должен быть представлен пользовательской переменной, которая берет уровни обработки как значения. После того, как Вы выполняете каждое моделирование в эксперименте, Вы должны поместить получающееся значение в матрицу результатов так, чтобы данные могли быть проанализированы процедурой ANOVA.

Позиция в матрице результата для каждого результата определена комбинацией обработки. Например, если эксперимент рассматривает 4 машинных типа и две скорости каждого, результат моделирования третьей машины в высокой скорости вошел бы в матрицу результата в позиции [3,2]. На конец, первое выполнение в этой комбинации обработки вошло бы в элемент [3,2,1] из матрицы результата.

Каждое измерение матрицы результата проанализировано как фактор, процедурой библиотеки ANOVA. Не существует никакого произвольного предела числу уровней обработки в пределах фактора, только наложенный виртуальной памятью вашего компьютера.

Мы можем исследовать любой эффект, давая свое собственное измерение в матрице результата. Если мы так хотим, мы могли бы даже отделить потоки случайного числа и обработать их индивидуально, как отличные факторы. Каждое начальное число случайного числа представило бы отличный уровень обработки. Это обеспечило бы информацию относительно относительной важности каждого случайного входного процесса.

Обычно, тем не менее, мы используем единственное измерение для копий, в следующее в пределах единственной комбинации обработки. Тогда мы сообщаем процедуре ANOVA, для которой используется измерение. Это сделано во втором параметре. Не все эксперименты нуждаются в повторном измерении. ANOVA часто в состоянии извлечь оценку стандартной ошибки и без этого. Однако, в зависимости от плана эксперимента, иногда недостаточно остаточных данных, чтобы оценить стандартную ошибку. Тогда дисперсионный анализ не будет в состоянии вычислить статистику F для таблицы ANOVA. В таких случаях, Вы должны будете или добавить повторные выполнения в эксперимент или удалить взаимодействия высокого уровня из статистической модели,

чтобы обеспечить достаточные остаточные данные для оценки стандартной ошибки. Оба из этих методов требуют изменения в вызове процедуры библиотеки ANOVA.

13.2.5 Потоки случайных чисел

Рекомендация, часто используемая в создании экспериментов: управляйте всеми факторами, что Вы можете; блокируйте то, чем Вы не можете управлять; делайте случайным то, что Вы не можете блокировать. Это относится к использованию случайных потоков в дискретном случайном моделировании.

Обрабатывать потоки случайного числа как факторы означает распределить измерение для каждого потока, и использовать каждое отличное начальное число как уровень обработки. Хотя возможно рассмотреть потоки случайного числа как факторы таким образом, и исследовать их относительные эффекты, это может быть лучше всего сделано как отдельное исследование.

Поскольку господствующее исследование главного производит и взаимодействия, мы больше интересуемся использованием потоков случайного числа, чтобы представить источники естественной и неизбежной изменчивости, резервируя измерения матрицы результата для изучаемых факторов. Самый общий подход комбинирует все потоки случайного числа в единственный псевдофактор, где каждый уровень обработки - ряд отличных начальных чисел для всех потоков случайного числа.

Это имеет некоторые подобия с блокированием, где все сравнения выполнены в пределах каждой гомогенной среды. процедура библиотеки ANOVA тогда может обработать это, копируя измерение отдельно, чтобы лучше оценить стандартную ошибку эксперимента. Эксперимент тогда состоит из экспериментальной "ячейки" для каждой комбинации обработки, с несколькими копиями выполнения в пределах каждой ячейки.

Если эти подходы слишком дороги, или нет времени, чтобы наполнять матрицу результата, выполнения должны быть "полностью рандомизированы". Это означает, что рандомизация не должна быть ограничена никогда, и что начальные значения случайного числа не должны быть повторены в пределах единственного потока случайного числа. Если эта процедура не в состоянии обеспечить достаточно многие степени свободы для анализа, Вы, возможно, должны также ограничить уровень взаимодействий, которые будут включены в статистическую модель.

13.3 Функции GPSS World

Теперь мы обратимся к определенным особенностям GPSS, который может использоваться в любой из фаз экспериментирования проекта моделирования. Сначала, мы полагаем, что особенности предусмотрели анализ пользовательских экспериментов. GPSS предусматривает два вида автоматически сгенерированных экспериментов, которые проектированы для Вас, и пользовательских экспериментов, которые спроектированы Вами.

13.3.1 Пользовательские эксперименты

Непосредственная цель пользовательского эксперимента состоит в том, чтобы обеспечить информацию, требовавшуюся встроенной процедурой библиотеки ANOVA. Это означает, что Вы должны заполнить матрицу результата всеми результатами эксперимента. Если Вы предпочитаете, Вы можете сделать это непосредственно с помощью списков команд и ручного ввода. Однако, язык PLUS, который позволяет вам написать программируемые эксперименты, может сделать вашу жизнь намного проще, если ваш эксперимент сложен. Вы должны начать с выбора факторов, эффекты которых Вы будете изучать. Например, Вы можете хотеть измерить эффект числа телефонных линий по среднему времени ожидания в телефонной станции. Фактором вашего эксперимента было бы число

телефонных линий, и Вы назначите это на первое измерение вашей матрицы результата. Затем мы решаем, сколько уровней обработки (то есть число телефонных линий) будет изучено. Скажем, что мы хотим моделировать обмены с 4, 8, и 16 линий. Мы тогда нуждались бы в 3 слотах в матрице результата для этих трех условий. Поэтому первое измерение в матрице результата имело бы размер 3.

Мы продолжаем добавлять факторы к эксперименту таким образом, используя число уровней обработки как размер соответствующего измерения в матрице результата. Обычно, мы используем последнее измерение для копирования. Размер копирующегося измерения должен очевидно быть достаточно большим, чтобы содержать следствия любой ячейки (то есть комбинация обработки) в эксперименте. Единственное ограничение то, что общее количество измерений не может превысить 6. То, когда Вы анализируете данные, Вы должны явно сказать подпрограмму ANOVA, которая используется для повторных измерений.

Это - общая схема создания пользовательских экспериментов. В продолжении этого раздела мы рассмотрим шаги более подробно. Мы рассмотрим внедренный язык программирования PLUS, требования и ограничения матрицы результата, и использования процедуры библиотеки ANOVA. Урок 19 из руководства оОбучающей программы GPSS приводит для Вас пример.

Эксперименты, созданные при помощи автоматических генераторов эксперимента, не считаются пользовательскими экспериментами, что обсуждается позже. Однако, Вам может показаться полезным изучить программирование сгенерированного эксперимента прежде чем Вы попытаетесь сделать это самостоятельно.

13.3.2 Эксперименты PLUS

PLUS - простой, но мощный язык программирования, который является важной частью среды моделирования GPSS. Полное описание PLUS находится в главе 8 этого руководства.

Урок 17 из руководства обучающей программы GPSS World также описывает его. эксперимент – это PLUS-процедура, где ключевое слово PROCEDURE заменено на EXPERIMENT. Эксперименты могут сделать все, что выполняют нормальные процедуры, и даже намного больше. Эксперимент начинается командой CONDUCT, которая называет вызванный эксперимент и передает ему любые параметры, которых могут потребоваться. Преимущество PLUS-эксперимента основано на способности вызвать процедуру библиотеки "DoCommand". Почти все команды GPSS могут выполнить запрос DoCommand со строкой команды как параметр. Помимо того, что эксперимент может вызвать DoCommand, но это может сделать и любая нормальная PLUS-процедура, которую назвали в течение выполнения эксперимента. Это добавляет дополнительную гибкость в программирование сложных экспериментов.

Пример запроса DoCommand:

DoCommand("START 1000");

Библиотечная функция DoCommand является специфической в том, что строка его параметра компилируется в глобальной области. Это означает, что Вы не можете сослаться на местные переменные в параметре запроса DoCommand. Напротив, если Вы должны передать местные значения к DoCommand, Вы должны использовать строковые команды PLUS, чтобы создать строку параметра, включая оцененные результаты.

Примером этого служит процедура "", созданная автоматическими генераторами эксперимента для регистрации выполненной информации в журнале:

* The Run Execution Procedure *

```

PROCEDURE SEM_GetResult() BEGIN
  /* Run Simulation and Log Results. */
  /* Treatments have already been set for this run. */
  TEMPORARY CurrentYield,ShowString,CommandString;

  /* Run Procedure Call */
  RunProc(SEM_NextRunNumber);

  /* Place Run Results in Journal. */
  ShowString = PolyCatenate("Run ",String(SEM_NextRunNumber),". ", "" );
  ShowString = PolyCatenate(ShowString," Yield=",String(CurrentYield),". ");
  ShowString = PolyCatenate(ShowString," Factor1=",String(Factor1), ";" );
  ShowString = PolyCatenate(ShowString," Factor2=",String(Factor2), ";" );
  ShowString = PolyCatenate(ShowString," Factor3=",String(Factor3), ";" );
  CommandString = PolyCatenate("SHOW """,ShowString,"""", "" );
  DoCommand(CommandString);

  SEM_NextRunNumber = SEM_NextRunNumber + 1;
  RETURN CurrentYield;

END;
```

Рисунок 13.4 –Использование DoCommand в процедуре PLUS

Эта процедура выпускает команду SHOW в 3 шагах. Это делает запись результата и условий выполнения в журнале. Сначала команда создает временную строку, ShowString, которая содержит переменные, оцененные в местном масштабе. Во вторых, это строит полную строку с законченной командой SHOW во временном переменном CommandString. Наконец, это вызывает процедуру библиотеки DoCommand. Нормальные процедуры подобного рода могут только вызвать DoCommand. Другими словами, такие процедуры должны вызываться напрямую или через PLUS-эксперимент.

В экспериментах со многими выполнениями, удобно использовать PLUS-процедуры, чтобы сделать набор из каждого моделирования, выполненного в эксперименте. Такая процедура инициализирует генераторы случайного числа и команды управления моделированием. Тогда Вы сможете установить высокий уровень вашего PLUS-эксперимента, чтобы вызывать процедуру выполнения всякий раз, когда моделирование должно быть выполнено.

Пример процедуры выполнения по умолчанию, созданной генераторами эксперимента, находится в разделе 13.4.2. Даже когда GPSS создает эксперимент для Вас, Вы вольны изменять процедуру выполнения, чтобы она соответствовала условиям выполнения вашего собственного проекта моделирования.

13.3.3 Матрица результатов

Первичный источник информации эксперимента сохранен в матрице результата. Для анализа результатов эксперимента в библиотеке ANOVA Вы должны убедиться, что матрица результата была установлена правильно.

В GPSS World, матричный объект может иметь до 6 измерений. Нет никакого произвольного предела на размер каждого измерения кроме ограничений, наложенных виртуальной памятью компьютера. Однако, одно из самых важных заданий в установке эксперимента GPSS - выбор факторов. Каждый фактор, который будет включен в эксперимент назначен на одно из измерений матрицы результата.

Кроме того, в зависимости от дизайна эксперимента, Вы будете вероятно хотеть использовать одно измерение для повторных измерений, которые должны быть достаточными для того, чтобы обработать выполнения в наибольшей ячейке. Для каждого измерения, используемого для фактора, по крайней мере такого размера, как число уровни обработки того фактора. Это – та причина, по которой каждое выполнение должно быть отражено в матрице результата. Если ваш эксперимент имеет больше чем один показатель результата, Вы должны определить отдельную матрицу результата, а затем использовать процедуру ANOVA для каждого.

Просмотр матрицы результата

Матрицы GPSS могут отображаться в окне матрицы или в стандартном отчете. Так как GPSS не может рассмотреть все 6 измерений одновременно, отчет представляет собой только двухмерный массив в окне матрицы. Маленькое диалоговое окно используется для выбора пласта матрицы, которая должна появиться.

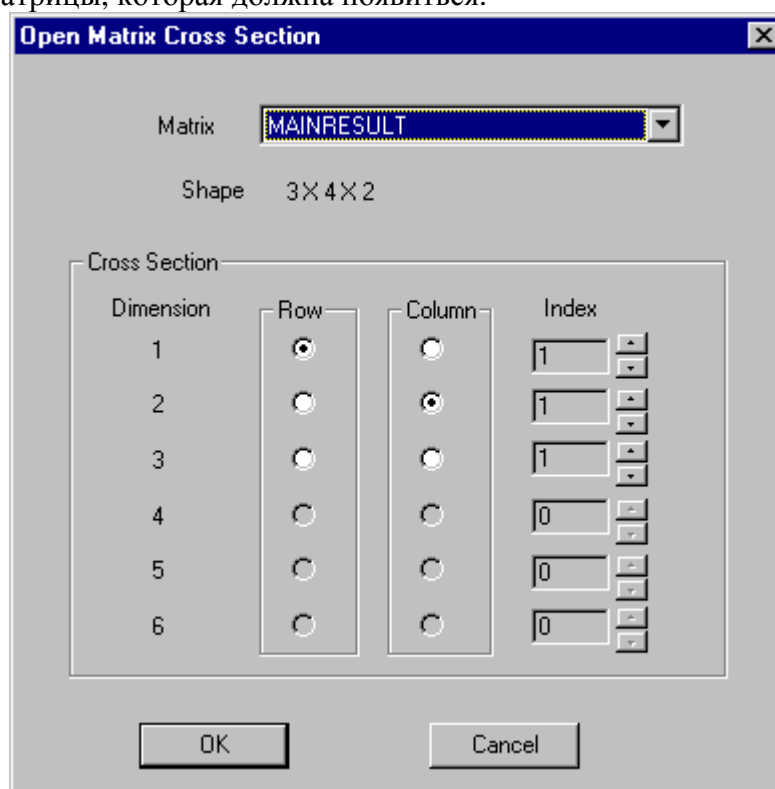


Рисунок 13.5 –диалоговое окно выбора матрицы

Рисунок 3.5 представляет диалог, который позволяет нам выбрать пласт через объект матрицы более высокого ранга. В этом случае мы имеем дело с 3-мерными матрицами результата 3 на 4 на 2. Чтобы использовать её, сначала выберите эти два измерения, чтобы сформировать строки и столбцы матричного окна, затем используйте индекс остающихся измерений, чтобы выбрать более глубокие пласты.

Номера строки и столбца указывают, где на пласте мы находимся, а индекс третьего измерения выбирает пласт. В матрицах с 4 или больше измерениями индексное значение было бы необходимо для каждого измерения, не участвующего как строка или столбец.

Как только матричное окно открыто, Вы можете просмотреть любое местоположения в пределах матрицы. Окна матрицы обновляются динамически, и пользователь может одновременно открыть любое число окон матриц. Другой способ отображения элементов матричного объекта состоит в том, чтобы включить результаты в стандартный отчет. Страница 2 параметров настройки объекта модели или моделирования (Редактирование / параметры настройки) имеет переключатель, который, будучи включен, помещает линию для каждого элемента матрицы в стандартном отчете.

Инициализация матрицы результатов

Переменные в пределах GPSS могут принимать разнообразные типов значений без пользовательского вмешательства. Значения преобразовываются от одной формы к другой, когда это необходимо. Например, строка преобразуется к вещественному числу, если оператор требует такого параметра. Переменные и матричные элементы могут принять значение UNSPECIFIED, чтобы указать, что никакое значение не было назначено. Значение UNSPECIFIED не является строкой и вызовет ошибку «Stop», если оператор встретит его при обработке выражения.

13.3.4 Библиотечная процедура ANOVA

Заключительный шаг в пользовательском эксперименте - обычно анализ результатов. Процедура ANOVA может обращаться с многоканальным дисперсионным анализом, рассматривающим до 6 факторов и до взаимодействий с 3 путями всех комбинаций главных факторов. Есть 3 аргумента процедуры ANOVA. Первое - название матрицы результата, которая описана более подробно, ниже. Второй аргумент - дополнительное измерение матрицы результата, которая используется для копирования. Каждый уровень в этом измерении представляет запуск с отличными случайными начальными значениями. Процедура ANOVA использует всю информацию, связанную с копирующимся измерением, как часть оценки стандартной ошибки. Это предоставляет оценке больше информации, и больше степеней свободы чем имело копирующееся измерение. Некоторые экспериментальные планы не используют копирующееся измерение. В том случае, используйте 0 как второй аргумент. Третий аргумент управляет взаимодействиями фактора, которые будут включены в статистическую модель. Если аргумент - 2, только взаимодействия с 2 путями включены в анализ. Если аргумент - 1, никакие сроки взаимодействия не включены. Значение 3 или больше - главные эффекты взаимодействия с 2 путями и взаимодействия с 3 путями, которые будут включены в статистическую модель. Четыре пути взаимодействия не поддерживаны GPSS World. Когда уровни взаимодействия удалены от модели, дополнительная информация и степени свободы становятся доступными для лучшей оценки стандартной ошибки. Главная цель процедуры ANOVA состоит в том, чтобы создать стандартную таблицу ANOVA в журнале, который указывает F статистически и его критическое значение. Кроме того, когда Вы вызываете процедуру ANOVA из процесса PLUS, при завершении работы процедуры ANOVA возвращается стандартная ошибка.

Таблица ANOVA

Ниже приведена таблица ANOVA, автоматически сгенерированная процедурой библиотеки.

ANOVA					
Source of Variance	Sum of Squares	Degrees of Freedom	Mean Square	F	Critical Value of F (p=.05)
A	28.000	2	14.000	5.600	3.89
B	21.000	3	7.000	2.800	3.49
AB	11.000	6	1.833	0.733	3.00
Error	30.000	12	2.500		
Total	90.000	23			

Treatment Level A B	Count	Mean	Minimum	Maximum	95% C.I. [SE]
1 1	2	3.000	2.000	4.000	[0.564, 5.436]
1 2	2	1.000	1.000	1.000	[-1.436, 3.436]

01/27/01 13:26:44 1.5811388

Рисунок13.6 – Таблица ANOVA

Каждый фактор и взаимодействие в статистической модели представлены на отдельной строке в таблице ANOVA. В каждой строке мы имеем основные вычисления и статистическое F для эффекта. Большие F оценивают более сильные эффекты. Их нужно считать существенным, если значение превышает то, что называют "критическим значением F", которое вычисляется для Вас в GPSS World. Таблица ANOVA выше показывает существенность эффекта фактора A, несущественность эффектов от фактора B и взаимодействия AB. Важно использовать методы сокращения разницы, используя дисперсионный анализ. Они искажают изменчивость, введенную, чтобы моделировать хаотичность реального мира. Сокращение собственной изменчивости наблюдений заставляет статистику F быть завышенной. Когда Вы делаете дисперсионный анализ, Вы должны исследовать предположения позади анализа, типа нормальности и неизменной разницы, которую статистики называют 'homoscedasticity'. GPSS World обеспечивает другую особенность с этой целью. Если Вы создадите объект таблицы GPSS с определенным названием, то процедура ANOVA заполнит его методом дисперсионного анализа. Это обсуждено в следующей секции. Сообщение о статистике ячейки следует за таблицей ANOVA, и это дает диапазоны и 95%-ые доверительные интервалы для каждой комбинации факторов в эксперименте, основанном на объединенной оценке стандартной ошибки. Хотя доверительные интервалы, связанные с комбинациями обработки весьма зависимы, диапазоны могут все еще использоваться как проверка предположений ANOVA. Когда они резко различаются, однородность остаточной разницы вызывает

сомнение. В том случае, Вы, возможно, должны исследовать данные более близко перед тем, чтобы делать ваши заключительные выводы. Один путь состоит в том, чтобы определить таблицу GPSS, чтобы собрать значения разницы, вычисленные в дисперсионном анализе. Следующая секция описывает, как сделать это.

13.3.5 Таблица остатков

Когда процедура ANOVA выполняет анализ, она ищет объект таблицы GPSS специального названия. Процедура ищет название таблицы, которое совпадает с названием результирующей матрицы за исключением того, что к нему прилагается строка "_Residuals". Если она находит объект таблицы полученного названия, процедура ANOVA сводит в таблицу остаток анализа ради проверки предположений, используемых ANOVA. Например, если Вы будете иметь объект таблицы GPSS по имени MyResults_Residuals определенным, когда Вы называете ANOVA с матрицей результата по имени MyResults, то GPSS World очистит таблицу и заполнит её значениями остатка от дисперсионного анализа. Отклонения и остатки могут иногда обнаруживаться в таблице Residuals, а нехватка гомогенной разницы (heteroscedasticity), когда она имеет место, является часто очевидной как здесь, так и в описательном сообщении о статистике, которое следует за таблицей ANOVA.

13.4 Автоматические генераторы экспериментов

GPSS World может создать эксперименты фильтрации или оптимизации. Их можно использовать для исключения нерелевантных факторов или для нахождения лучшей комбинации обработки. Процесс вставляет PLUS-эксперимент в объект модели и по желанию назначает функциональной клавише F11 соответствующую команду CONDUCT. Вся процедура занимает 4 шага:

1. Заполните диалог (меню Edit) и щелкните OK.
2. Отредактируйте процедуру запуска и нажмите OK.
3. Оттранслируйте модель (Ctrl+Alt+S).
4. Нажмите функциональную клавишу.

Вот и всё. Эксперимент запустится и запишет статус в журнал. После завершения будет также создан полный отчёт. В случае эксперимента оптимизации в отчет будет включено математическое описание поверхности отклика вокруг оптимума. Анализ отчета входит в ваши задачи.

Оба типа сгенерированных PLUS-экспериментов содержат включение процедуры запуска PLUS. Проследите за тем, чтобы эта процедура корректно задала условия запуска. Процедура запуска вызывается каждый раз при начале новой имитации. Вы также можете настроить генератор экспериментов на создание шаблонной процедуры запуска. Даже в этом случае её можно приспособить под свои требования. Технология этого описана в следующем разделе.

13.4.1 Процедура запуска

Процедура запуска – это гибкая связь между произведенным экспериментом и имитацией пользователя. И эксперименты отбора, и эксперименты оптимизации используют их. Произведенные эксперименты неоднократно вызывают процедуру запуска, по заданным в диалоге создания пользовательским именам, чтобы выполнить каждый запуск в эксперименте. Каждый эксперимент должен вызвать вашу процедуру запуска или процедуру запуска GPSS World. Процедура запуска используется, чтобы установить период измерения в течение моделирования, которое должно быть соблюдено. Это настраивает каждый запуск в эксперименте согласно специфическим требованиям вашего моделирования и вызывается один раз для каждого запуска в эксперименте. Процедуры,

которыми управляют PLUS процедуры вызывают эксперименты и поэтому могут содержать запросы DoCommand. Обычно процедура запуска содержит команды установки для моделирования, типа команд RMULT, чтобы управлять генераторами случайных чисел, и START и RELOAD команды, чтобы настроить период измерения запуска. Когда эксперимент генерируется автоматически, генератор по желанию создаёт шаблонную процедуру запуска, которая может быть изменена как необходимо. В качестве примера приведена процедура запуска по умолчанию, созданную генератором экспериментов:

```
PROCEDURE RunProc(Run_Number) BEGIN

    DoCommand("CLEAR OFF"); /* Must use OFF to preserve results. */

    /* EXPAND THIS RMULT IF YOU HAVE MORE RNGs. */
    /* All Random Number Streams must have new seeds. */
    TEMPORARY CommandString;

    /* Evaluate before passing to DoCommand. */
    CommandString = Catenate("RMULT ",Run_Number#111);

    /* DoCommand compiles the string in Global Context. */
    DoCommand(CommandString);

    /* SET UP YOUR OWN RUN CONDITIONS. */
    DoCommand("START 100,NP"); /* Get past the Startup Period. */
    DoCommand("RESET"); /* Begin the Measurement Period. */
    DoCommand("START 1000,NP"); /* Run the Simulation. */

END;
```

Рисунок 13.7 –Стандартная процедура запуска

Если процедура запуска создаётся в GPSS World, вы можете редактировать её под свои нужды. Процедуру можно редактировать как до помещения в объект модели, так и после этого.

13.4.2 Фильтрующие эксперименты

Фильтрующие эксперименты используются, чтобы искать факторы, которые должны быть исследованы более тщательно или те, которые подозреваются в несущественности. Однако, комбинаторная природа измерения главных эффектов и всех возможных взаимодействий часто делает полный набор из запусков, названных полным факторным планом экспериментов, слишком длинным и/или дорогим для запуска. Чтобы сделать процедуру более управляемой, мы ограничиваем число уровней обработки двумя факторами, и ищем пути управления только частью возможных запусков, не жертвуя самой важной информацией. Генератор фильтрующего эксперимента GPSS World делает большинство работы за Вас. Вы должны решить, какую информацию Вы можете опустить, чтобы уменьшить число запусков в эксперименте. Обычно это были бы высокоуровневые взаимодействия фактора, которые являются обычно незначимыми и могут игнорироваться. Даже если Вы считаете иначе, помните, что цель эксперимента отбора состоит в том, чтобы идентифицировать важные факторы, которые будут подробнее изучены позже. Когда Вы требуете, чтобы GPSS уменьшил полный факторный план, показывающий на экране эксперимент исходя из набора запусков, некоторые

эффекты смешиваются с другими, и Вы не можете отличить их в дисперсионном анализе. GPSS World разделит набор всех возможных множеств эффектов на подмножества называемые альтернативными группами. Только объединенный результат всех эффектов в группе показателен в дисперсионном анализе. Это означает, что, если фактор А и фактор В находится в альтернативной группе, Вы потеряете возможность определить, какой из них является ответственным за наблюдаемый эффект. Возможно еще худшим событием является наличие двух сильных эффектов в той же самой альтернативной группе, которые отменяют друг друга, таким образом скрывая фактические эффекты каждого фактора. Однако, обычно возможно устроить альтернативные группы так, чтобы каждый главный эффект и самые важные взаимодействия с 2 путями находились в собственных альтернативных группах, только с более высокими порядками взаимодействия. GPSS World предоставляет альтернативные группы по умолчанию, которые изолируют главные эффекты, если это возможно. Это облегчит для Вас изменение альтернативных групп, если настройки по умолчанию не подходят. В любом случае, когда Вы запускаете эксперимент отбора в GPSS World, Вы должны всегда исследовать альтернативные группы прежде, чем Вы начинаете запуски. Если Вы не отделили самые важные эффекты в отличные группы, интерпретация результатов будет трудна.

Генерация PLUS-эксперимента отбора

Первый шаг в создании эксперимента отбора - открыть диалог. Для этого щелкните "Insert Experiment" в меню «Edit» главного окна, и нажмите "эксперимент отбора ...". Для этого Вы должны иметь открытый активный объект модели. Теперь заполните поля и нажмите ОК, когда Вы готовы.

Рисунок 13.8 показывает диалоговое окно эксперимента отбора

Screening Experiment Generator

Experiment Name:

'Run Procedure' Name:

	Name (User Variable)	Value 1	Value 2
A	<input type="text" value="Node_Count"/>	<input type="text" value="100"/>	<input type="text" value="200"/>
B	<input type="text" value="Min_Msg"/>	<input type="text" value="512"/>	<input type="text" value="1024"/>
C	<input type="text" value="Max_Msg"/>	<input type="text" value="12144"/>	<input type="text" value="24288"/>
D	<input type="text" value="Fraction_Short_Msgs"/>	<input type="text" value="600"/>	<input type="text" value="300"/>
E	<input type="text" value="Intermessage_Time"/>	<input type="text" value="1.0"/>	<input type="text" value="0.5"/>
F	<input type="text"/>	<input type="text"/>	<input type="text"/>

Fraction: ☐ Full ☐ Half ☒ Quarter ☐ Eighth ☐ Sixteenth Run Count:

Results: Expression Matrix

☒ Generate Run Procedure ☒ Load F12 with CONDUCT Command

Рисунок 13.8 –Генератор эксперимента отбора

Чтобы определить производимый PLUS-эксперимент, мы просто заполняем поля диалога. "Название эксперимента" и «области 'Name'» - процедуры, которыми управляют, используются как названия ПРОЦЕДУРЫ в произведенных экспериментах. ЭКСПЕРИМЕНТ переименовывает процедуру запуска неоднократно. Факторы

эксперимента прибывают следом. Каждое название фактора - фактически название пользовательской переменной и должно подчиняться правилам GPSS World. То есть именование должно начинаться с алфавитного символа, и это не должно совпадать с ключевым словом, SNA, или классом SNA. Начиная с экспериментов отбора в GPSS World введены полные или дробные факторные эксперименты 2 КБ. Есть два уровня обработки, которые будут определены для каждого фактора. Вы должны определить названия и два уровня обработки для между 1 и 6 факторами, включительно. Факторы должны быть определены, последовательно начинаясь с первого фактора. Выбор уровней обработки является критическим. Вы можете хотеть сделать некоторое предварительное экспериментирование так, чтобы выборы, которые Вы делаете, были хорошие. Выберите уровни, которые должны быть далеко друг от друга, чтобы выявить изменения поведения и пробовать избежать уровней, где эффекты скрыты другими факторами. Группа "Дробь" является следующей. Это позволяет нам определять, какой дробью полного эксперимента 2 КБ нужно управлять. Счет запуска, который закончится, определен направо. Выберите меньшую дробь, чтобы уменьшить счет запуска. "Выражение результата" требуется. Вы должны определить выражение, которое будет оценено как метрическое из моделирования. Затем мы имеем два флажка, которые позволяют нам выбирать дополнительные варианты. Вы только должны нажать функциональную клавишу F11, чтобы начать эксперимент.

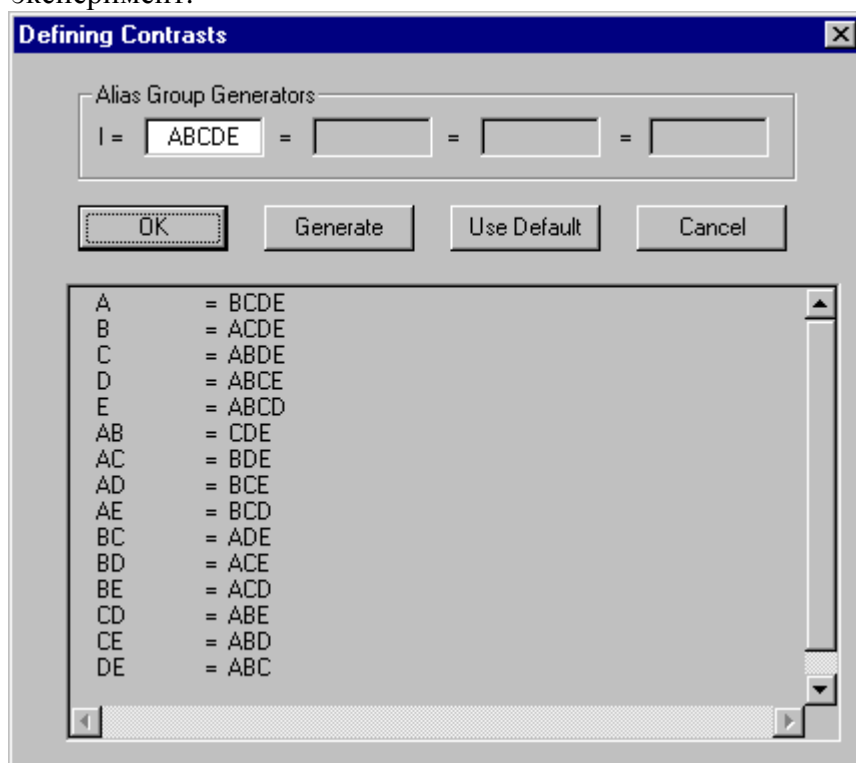


Рисунок 13.9 –Диалоговое окно псевдонима групп. Улучшенная модель эксперимента.

Из диалога «альтернативные группы» мы можем видеть, что этот экспериментальный дизайн изолирует все главные факторы, не путает их со взаимодействиями, и это даже изолирует взаимодействия с 2 путями первых немногих главных эффектов. Бросьте более близкий взгляд на эффекты, используемые как "генераторы" в диалоге «альтернативные группы». Мы можем изменить разделение эффектов в различный набор альтернативных групп, просто используя различные взаимодействия высокого уровня сгенерированного набора. Чтобы сделать это, измените генераторы и нажмите на кнопку GENERATE. Прежде, чем Вы делаете это, есть несколько вещей, которые необходимо знать о генераторах. Сначала, Вы должны знать то, что когда GPSS World создает альтернативу, это сначала расширяет набор генераторов, формируя весь возможный модуль на 2

продукта и добавляя их к набору генераторов, теперь названных "Расширенным Набором" генераторов.

Например, если ABCD и BCDE будут генераторами, то их продукт, выполнение приложения, будет в «Расширенном Наборе» генераторов (ABCDE * BCDE = ABBCDDDE = (BB) (машинописная копия) (DD) E = выполнение приложения). Отметьте, что любой эффект в «Расширенном Наборе» генераторов не будет появляться в сообщении эффектов. Это означает, что Вы не должны выбирать генераторы, которые приводят к важному эффекту, происходящему в «Расширенном Наборе» генераторов. Вы можете возвратиться к генераторам по умолчанию, предложенным GPSS World, нажимая на Кнопку «Use defaults». Нажмите Enter, чтобы создать новый список альтернативных групп.

На следующий шаг должен определиться PLUS-кодекс, включающий эксперимент в ваш объект модели. Когда Вы пошли на компромисс между числом запусков и совмещением эффектов, щелкните кнопку ОК в диалоге альтернативных групп и кнопку «Вставка эксперимента» в диалоге генератора эксперимента отбора. Если Вы потребовали, чтобы GPSS World произвел процедуру запуска, откроется соответствующий диалог. Детали об определении процедуры запуска находятся в предыдущей секции. В любом случае, Вы можете редактировать произведенный эксперимент непосредственно в вашем окне модели.

Сгенерированный эксперимент

*

* ScreenEthernet

* Fractional Factorial Screening Experiment

*

Global_Delays MATRIX ,2,2,2,2,2

INITIAL Global_Delays,UNSPECIFIED

EXPERIMENT ScreenEthernet() BEGIN

/* Run 1 */

Node_Count = 100;

Min_Msg = 512;

Max_Msg = 12144;

Fraction_Short_Msgs = 600;

Intermessage_Time = 1.0;

IF (StringCompare(DataType(Global_Delays[1,1,1,1,1]),"UNSPECIFIED")'E'0)

THEN BEGIN

/* Run Procedure Call */

DoTheRun(1);

Global_Delays[1,1,1,1,1] = QT\$Global_Delays;

END;

/****** Runs 2-15 would go here *****/

/* Run 16 */

Node_Count = 200;

Min_Msg = 1024;

Max_Msg = 24288;

Fraction_Short_Msgs = 300;

```

Intermessage_Time = 1.0;
IF (StringCompare(DataType(Global_Delays[2,2,2,2,1]),"UNSPECIFIED")'E'0)
THEN BEGIN
    /* Run Procedure Call */
    DoTheRun(16);
    Global_Delays[2,2,2,2,1] = QT$Global_Delays;
END;

/* Aliased Effects in Fractional Factorial Experiment */
EFFECTS(Global_Delays,"I=ABCDE");

END;
*****
* Run Procedure *
*****
PROCEDURE DoTheRun(Run_Number) BEGIN

    DoCommand("CLEAR OFF"); /* Must use OFF to preserve results. */

    /* EXPAND THIS RMULT IF YOU HAVE MORE RNGs. */
    /* All Random Number Streams must have new seeds. */
    TEMPORARY CommandString;
    /* Evaluate before passing to DoCommand. */
    CommandString = Catenate("RMULT ",Run_Number#111);
    /* DoCommand compiles the string in Global Context. */
    DoCommand(CommandString);

    /* SET UP YOUR OWN RUN CONDITIONS. */
    DoCommand("START 100,NP"); /* Get past the Startup Period. */
    DoCommand("RESET"); /* Begin the Measurement Period. */
    DoCommand("START 1000,NP"); /* Run the Simulation. */

END;
*****

```

Запуск эксперимента

Команда CONDUCT используется в GPSS World для начала эксперимента. Синтаксис схож вызовом процедуры. Однако, есть способ лучше. В диалоге создания эксперимента можно настроить GPSS World на использование клавиши F11 с соответствующей командой CONDUCT. Это можно увидеть в настройках объекта модели. (см. **Edit / Settings / Function Keys**).

Чтобы начать выполнение эксперимента, необходимо сделать две вещи: во-первых, необходимо оттранслировать модель, создав объект имитации. Для этого щелкните **Command / Create Simulation** или нажмите **Ctrl+Alt+S**.

Затем нажмите F11. Это вводит команду CONDUCT. эксперимент начинает работу, сообщая статус и выводя информацию в окно журнала созданного объекта имитации. На рисунке 13.10 показан результат выполнения эксперимента по отбору.

Alias Group	Effect	Sum of Squares	Degrees of Freedom	F - for Only Main Effects	Critical Value of F (p=.05)
A = BCD	0.030	0.002	1	0.000	10.13
B = ACD	-0.058	0.007	1	0.000	10.13
AB = CD	-0.178	0.063	1		
C = ABD	0.063	0.008	1	0.000	10.13
AC = BD	-0.014	0.000	1		
AD = BC	-0.109	0.024	1		
D = ABC	0.177	0.062	1	0.000	10.13
Grand Mean	11.456				
Total		1050.158	7		
Error		1050.079	3		

01/16/01 13:45:33 Experiment Ended.

Рисунок 13.10 – Представление генератора экспериментов

Отчет об эксперименте показывает силу каждого эффекта и, если возможно вычисляет статистическую величину F для групп переменных с главными эффектами в качестве участников. Так Вы можете запланировать будущие действия, игнорируя незначачие факторы и более полно исследуя важные.

13.4.3 Оптимизация экспериментов и поверхностная характеристика

Оптимизация и количественный прогноз поведения системы часто - главные цели проекта моделирования. Оба из них непосредственно поддерживаются языком GPSS World's Optimizing Experiment Generator.

Методология поверхностной характеристики - набор статистических методик для эмпирической разработки модели и образов исследования. Хотя чаще всего используют оценку оптимальных уровней обработки, это также может предоставить важную информацию для общих прогнозов объема Выпуска, ожидаемого при непроверенных условиях. GPSS поддерживает несколько важных методологий поверхностной характеристики, в известном смысле, предназначенный для того, чтобы сделать их удобными. Анализ моделирования устанавливает начальные условия в нескольких диалоговых окнах, и тогда GPSS автоматически создает отклик поверхностного эксперимента, который ищет оптимальное значение. Идеально, Вы можете запустить эксперимент и позволить ему работать автоматически до тех пор, пока он не найдет оптимальное значение Выбранного показателя. Когда Вы заполняете диалоги в windows, Вы, в действительности, определяете блок (фактически "гиперблок") названный локальной экспериментальной областью, которая блуждает по набору допустимых факторных уровней. Если Вы сделаете этот блок слишком маленьким, то эксперимент будет выполняться больше, чем оптимально необходимо. Если Вы сделаете его слишком большим, то оценки оптимальных уровней обработки будут менее точными, чем они могли быть.

GPSS World применяет несколько различных методологий, поскольку он исследует поверхностные характеристики объекта моделирования GPSS WOULD. Первичное понятие - понятие локальной экспериментальной области, которая определена Вами в диалоге установки. Чтобы определить его, каждый из 5 факторов, которые будут

получены, связаны с двумя значениями, определяющими степень локальной экспериментальной области. Тогда, это - целая область, которая перемещает факторное пространство. Цель, которая не всегда достижима, состоит в том, чтобы содержать оптимум в пределах локальной области, осуществить подтверждение выполнения, и затем сообщить об оптимальных условиях и математической модели, которая использовалась.

GPSS использует несколько главных экспериментов различной сложности, чтобы получить необходимую информацию. Поскольку результаты доступны, они должны соответствовать или линейной модели или второй модели (включая двухсторонние взаимодействия) данных. Это начинается при использовании метода Steepest Ascent, поскольку он оставляет свою отправную точку. Директория определена градиентом первой модели, которая соответствует данным. Если происходят сбои теста для первой модели, используется вторая.

Как только градиент вычислен, центр локальной экспериментальной области перемещает его до тех пор, пока он не попадает или в сортировку с убывающим шагом или с пределом, утверждаемым в локальной экспериментальной области, как определено в генераторе оптимизированного эксперимента. Если диапазон уровней обработки будет слишком узким, то выполнится много ненужных шагов, чтобы достигнуть градиента. Каждый шаг перемещения требует моделирования в центре локальной экспериментальной области, и о его результатах сообщают в журнале. Если, при остановке движения, локальная экспериментальная область не содержит оптимальных значений, эксперимент попытается перейти в новую директорию. Это действие может быть подавлено перенаправлением границ в генераторе эксперимента оптимизации.

Initial Local Experimental Region			Movement Limits	
Factor Name (User Variable)	Value 1	Value 2	Low Limit	High Limit
A Node_Count	100	300	0	100
B Min_Msg	256	512	0	100
C Max_Msg	6072	24288	0	100
D Fraction_Short_Msgs	600	300	0	100
E Intermessage_Time	1.0	0.5	0	100

Result
Expression: [QT\$Global_Delays/1000]
☒ Maximize ☐ Minimize

☒ Generate Run Procedure ☒ Load F12 with CONDUCT Command

Buttons: Insert Experiment, Cancel, Help

Рисунок 13.11 – Генератор оптимизированного эксперимента

Рисунок 13.11 показывает диалог обычно используемого генератора оптимизированного эксперимента. Так же, как с генератором представления эксперимента к нему обращаются через меню правка главного окна Windows.

Чтобы определить PLUS эксперимент, который генерируется, мы просто заполняем в поля диалога. "Название Эксперимента" и 'Запуск процедуры' Имя – это поля,

использующиеся как имена процедур в сгенерированных экспериментах. Эксперимент неоднократно вызывает выполненную процедуру.

Затем появляются факторы эксперимента. Каждое факторное название - фактически название пользовательской переменной и должно удовлетворять условию соглашения об именах GPSS WOULD. Имена должны начинаться с алфавитного символа, и это не должны совпадать с ключевым словом, протоколом SNA, или классом протокола SNA. Два уровня обработки каждого фактора определяют локальную экспериментальную область, которая смещает поиск оптимальных условий. Вы должны определить имена и два уровня обработки для диапазона между 1 и 5 факторами, включительно. Факторы должны быть определены, последовательно начиная с А фактора. Выбор уровней обработки крайне важен.

Несколько важных обменов неявны в способе, которым Вы определяете локальную экспериментальную область. Во-первых, формируется отправная точка эксперимента. Вы должны сделать ее как можно ближе к оптимальной. Во-вторых, минимизируя ее, как правило, ваша математическая модель наиболее пригодна. В-третьих, когда эксперимент делает один шаг к оптимуму, точка перемещается в смежную область. Поэтому, большая экспериментальная область сделает меньше шагов. Как Вы видите, результаты конфликтуют друг с другом. Вам решать к какому компромиссу прийти. В дальнейшем, не упускайте факт того, что легко изменить сгенерированный эксперимент после того, как он был вставлен в вашу модель. Стоит проверить сгенерированный PLUS код, чтобы увидеть способы улучшения эксперимента. Возможно, Вам захочется сделать некоторое предварительное экспериментирование так, чтобы выборы, которые Вы делаете, были хорошими.

Вы должны использовать эксперименты оптимизации. Раздел диалога для перемещения границ дает Вам возможность ограничить движение локальной экспериментальной Области. Если возможно Вы должны ввести границы для каждого из факторов так, чтобы время не было потрачено впустую на неосуществленные моделирования. Перенаправление границ - другой способ ограничить поиск. Значения устанавливают границы для количества экспериментов по времени и могут передвигаться в новую директорию факторного пространства. Эти границы могут предотвратить бесконечное выполнение цикла.

Требуется "Выражение результата". Вы должны определить выражение, которое будет оценено как показатель моделирования.

В Итоге, у нас есть два переключателя, которые позволяют нам выбирать дополнительные опции. Во-первых, у вас может быть GPSS, которая производит шаблон выполнения процедуры, чтобы согласовать с данным сгенерированным экспериментом. Вы можете редактировать выполнение процедуры, чтобы удовлетворить Ваши потребности. Во-вторых, у вас может быть GPSS, устанавливающая настройки, которые загружают соответствующую команду CONDUCT в установку нажатием клавиши F12. Если Вы выбрали это, после того, как создали объект моделирования, Вы только должны нажать F12, чтобы выполнить эксперимент.

Следующий шаг должен вставить PLUS код, включающий эксперимент в ваш шаблонный объект. Когда Вы заполнили все соответствующие поля в диалоге генератора оптимизации эксперимента, нажмите Insert Experiment, чтобы создать PLUS код и вставить его в Вашу модель. Если Вы запрашиваете GPSS для того, чтобы он сгенерировал выполнение процедуры, Вы видите диалоговое окно, которое позволяет нам персонализировать его. Подробности об индивидуализации выполненной процедуры находятся в начале этого раздела.

В любом случае, Вы, возможно, захотите отредактировать сгенерированный эксперимент непосредственно в окне вашей модели. Вы можете изменить выполнение процедуры там, как Вы пожелаете. Нажмите только ОК, и все новые операторы GPSS включающие эксперимент оптимизации, помещены внизу вашей модели.

Выполнение эксперимента

Conduct - команда GPSS, которая используется, чтобы вызвать эксперимент. Параметр к команде Conduct подобна типичному вызову процедуры. Но у нас есть лучший путь. В диалоговом окне дерево построений мы также можем вызвать загрузку GPSS через клавишу F12 с соответствующей командой Conduct. Вы можете видеть это в параметрах настройки вашей модели (см. **редактирование / параметры настройки / функциональные клавиши**).

Чтобы запустить выполнение эксперимента, мы должны сделать две вещи. Сначала мы должны преобразовать модель, таким образом создавая объект моделирования. Нажмите на **команда/ создать моделирование** или сочетание Ctrl+Alt+S.

Тогда мы нажимаем функциональную клавишу F12. Это вводит команду Conduct. Эксперимент переходит в режим работы, сообщая текущее состояние и выводя его в окно журнала недавно созданного объекта моделирования. Рисунок 13.12 показывает результаты эксперимента оптимизации.

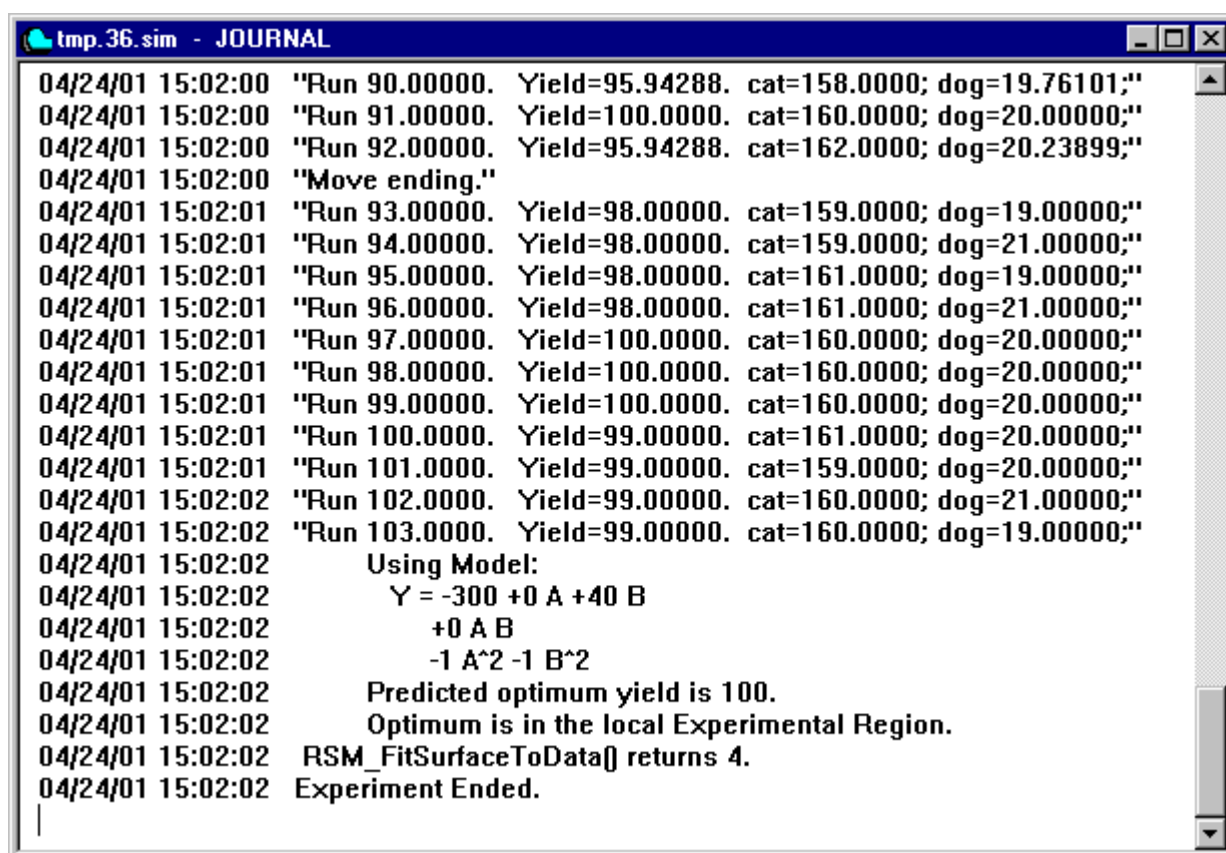


Рисунок 13.12 – Результаты оптимизации эксперимента

Когда все прошло успешно, эксперимент оптимизации сообщает результаты каждого запуска. Благодаря этому можно просмотреть результаты каждого подэксперимента и проследить движения всех локальных экспериментальных областей. Сгенерированный PLUS эксперимент пытается переместить локальную экспериментальную область так, чтобы это включить оптимальные условия. Эксперимент заканчивается, когда это происходит, или когда предел перенаправления или предел движения достигнуты.

Когда перемещение останавливается, о результатах пригодности тестирования и уравнения для соответствующих поверхностных характеристик также сообщаются в окне журнала. Тогда невыполненный PLUS эксперимент пытается проверить спрогнозированные оптимальные условия запуском моделирования.

13.5 Краткие указания по выполнению

Мы заканчиваем главу некоторыми полезными советами, которые могут помочь избежать ловушек, в то время как Вы заняты проведением экспериментов.

- GPSS использует пользовательские переменные, чтобы взять уровень обработки каждого фактора, как значения. Когда Вы создаете названия для своих переменных, они не должны совпадать с существующими ключевыми словами, протоколами SNA, или классами протокола SNA. Названия с больше, чем двумя символами, сопровождаемыми, по крайней мере одной цифрой или символом подчеркивания приемлемы. Факторные имена в экспериментах - простые пользовательские имена, принимающие обозначенные уровни обработки, как значения во время эксперимента.
- По умолчанию, GPSS использует # символ, чтобы обозначить умножение и резервы * для косвенности протокола SNA. Если Вы предпочитаете использовать * для умножения, Вы можете переключить роль * и # переключателем в первой странице шаблонов параметра настройки. К этому обращаются через меню редактирования.
- Когда Вы создаете свои собственные имена, они не должны содержать существующие ключевые слова или слова класса протокола SNA. Имена, включающие символы подчеркивания корректны.
- Параметры настройки, унаследованные от базового объекта моделирования, включая использование функциональной клавиши (см. меню редактирование). Произвольно, генераторы эксперимента загружают функциональную клавишу соответствующей командой Conduct. Визуальный генератор экспериментов произвольно загружается нажатием F11, генератор эксперимента оптимизации, F12. После создания моделирования, чтобы выполнить эксперимент, только нажимают функциональную клавишу.
- Директива DoCommand позволяет эксперименту или любую процедуру, вызванной этим же экспериментом, выполнять любую команду GPSS, кроме команды Conduct.
- DoCommand преобразует свою строку параметра в глобальном контексте. Не используйте временные имена переменной в этом контексте. Если Вы должны передать значения локальных переменных или параметров к директиве DoCommand, сначала Вы должны встроить строку, использующую PLUS строковые процедуры. Это заставит локальные переменные полностью быть оцененными прежде, чем ввести область видимости директивы DoCommand. Нижняя строка – те значения, что Вы должны использовать, не имена временных переменных в строке, которую Вы передаете к процедуре DoCommand. глобальные переменные могут быть именованы как угодно. Для примеров, см. выполнение процедуры в разделе 14.3.2.
- Вы можете поместить выражения Include в любом месте PLUS процедуры. Обычно помещают все команды после последнего Include оператора.
- Если у вас будет больше чем один показатель, описывающий ваши наблюдения, то Вы должны будете редактировать любой сгенерированный эксперимент и определить дополнительные матрицы результата и вызова ANOVA.
- Если в таблице ANOVA выяснилось, что статистика F не могла быть вычислена, попытайтесь уменьшить уровень взаимодействий в анализе, если есть возможность.
- У моделирований может быть любое число экспериментов. Используйте команду Conduct, чтобы вызвать эксперимент.
- Не передавайте параметры Procedure или временные переменные к DoCommand, не оценив их сначала. Это рассматривалось выше. Строка параметра DoCommand

откомпилирована в глобальной области видимости, которая не может обратиться к локальным переменным. Вы можете создать строку перед вызовом в DoCommand, и затем, в отдельном шаге, передать ее как параметр.

- Не забывайте, что операторы PLUS могут только появиться внутри PLUS процедуры.
- Вы можете заменить любую процедуру, включая эксперименты, в существующем объекте моделирования. Так как они длиннее, чем простые команды, Вы, вероятно, использовали бы оператора INCLUDE как заказную команду, которая указывает на текстовый объект, содержащий процедуру.
- Инициализируйте свою матрицу результата к UNSPECIFIED. Значение по умолчанию объекта матрицы языка GPSS 0. Инициализация к UNSPECIFIED препятствует тому, чтобы 0 значений использовались как результаты. Выполнение инициализации вне PLUS эксперимента позволяет вашему эксперименту быть прерываемым (см. следующий совет).
- Если Вы хотите, чтобы ваши эксперименты были прерываемы, в вашем PLUS эксперимент прежде, чем каждый будет работать, Вы должны проверить элемент матрицы результата на "UNSPECIFIED" тип данных. Тогда, когда эксперимент перезапущен, будучи остановленным (и возможно сохранен), законченный выполняется, как пропущенный. Чтобы видеть пример этого, обратитесь к любому сгенерированному эксперименту.
- Сгенерированные эксперименты вставлены внизу вашей модели. Эксперимент заменяет любое название экспериментом, зарегистрированным ранее. Вы ответственны за удаление любых старых сгенерированных экспериментов, в которых Вы больше не нуждаетесь.
- Вы должны всегда завершать полный тест основного моделирования прежде, чем начать фазу экспериментирования.
- Когда Вы получаете ошибки трансляции, используете команду "Next Error" в меню поиска. Когда Вы получаете ошибки "времени выполнения", используете "искать, Чтобы выровнять" команду в меню поиска, чтобы найти местоположение ошибки.

Глава 14 – Сообщения об ошибках

14.1 Введение

Есть много условий, которые могут предотвратить нормальное завершение операции. Они могут произойти от операции меню, во время преобразования модели, или во время выполнения моделирования.

Ошибки, которые происходят, поскольку Вы инициализируете команды и окружающие операции, немедленно появляются и препятствуют тому, чтобы операция началась. Они требуют, чтобы Вы исправили условие или выбрали другую операцию. Другие две категории ошибок вообще обусловлены изменениями в вашей модели.

Синтаксические ошибки могут произойти, когда Вы преобразуете модель или команду. Чтобы исправить ошибки, которые происходят в преобразованной модели, Вы можете использовать **поиск / следующая ошибка** или **поиск / предыдущая ошибка**, чтобы переместить курсор непосредственно в ошибочного оператора. Если ошибка происходит в файле включения, Вы можете получить лучшие диагностические сообщения от временно помещенных операторов в объект моделирования.

Ошибки, которые происходят в пределах моделирования, называют Error Stops. Они вводят моделирование в состояние ожидания, и если эксперимент происходит, он прерывается. Обычно Error Stops говорят Вам, что ошибка может быть исправлена в модели, командами **поиск / следующая ошибка**. Вместо этого число файлов включения (модель - число 0 файла) и текстовый номер строки даны в сообщении об ошибках.

Редактирование / команда меню Goto обычно используется, чтобы найти и исправить эти ошибки.

Иногда Вы должны будете провести некоторый анализ, чтобы найти ошибку. Если источник ошибки не очевиден, Вы, возможно, захотите остановить моделирование в определенном месте и затем использовать команду SHOW, Open Windows и копии экрана, чтобы видеть то, что получилось.

Для ненадежных Error Stops Вам следует использовать окно Blocks и инструментальную панель отладки. Тогда Вы можете поместить условие остановки, которое вызывает остановку как раз перед тем, когда ошибочное условие происходит. Затем используйте интерактивные команды SHOW, чтобы оценить все прогнозируемые операнды. Когда условие выполняется поздно, Вам будет удобнее сохранить объект Simulation как раз перед моделируемым временем, когда ошибка должна произойти. Это может экономить время, когда Вы должны повторно запустить последнюю часть моделирования несколько раз в диагностическом процессе. Вы можете продолжить вновь открывать файл, который сохранял моделирование вместо того, чтобы делать повторный запуск моделирование к ошибочному пункту.

Главные сообщения об ошибках перечислены далее в алфавитном порядке с дополнительным объяснением. Где Вы видите XXX в сообщении, оно означает, что сообщение об ошибках содержит фразу, которая зависит от подробностей ошибочного условия. В общем, некоторые сообщения являются результатом больше чем одного условия.

14.2 Сообщения и Объяснения

"Поток данных моделирования потерпел неудачу"

Не удалась операция Data Stream. Подробности были даны в предыдущем сообщении. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить это ошибку.

"Заявка заблокирована из-за невозможных условий"

Активная заявка пытается ждать условие, которое никогда не может произойти. Такая заявка никогда не была бы в состоянии ввести блок. Вы должны изменить значения,

операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку. Не пытайтесь заблокировать на интегрированной пользовательской переменной. Используйте операционные пороги поколения в интегрирующейся команде вместо этого.

"Заявка попыталась захватить или выгрузить собственное устройство"

Активной заявке уже принадлежит устройство, которое она снова пытается захватить или выгрузить. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Заявка, которой принадлежит устройство, пытается завершиться"

Моделирование пытается закончить заявку, которой принадлежат одно или более устройство. Заявки должны выпустить устройства прежде, чем они смогут быть разрушены. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Заявка, которая была выгружена в устройстве, попыталась захватить или выгрузить его"

Активная заявка выгружена в устройстве, которое она пытается захватить или выгрузить снова. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Доступ запрещен"

Чтение или операция записи были не разрешены, потому что другой процесс использует файл.

"Определите тип данных"

Вычисление терпело неудачу, потому что операнд взял значение, которое не может быть применено к арифметическому выражению. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Переполнение суммирования"

Результат операции превышает максимальное допустимое значение. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Не допустимо смешивать группу генераторов XXX как тождественную(то есть пустой указатель)"

Вы не можете использовать данное выражение как условное название генератора группы. См. главу 13 *Справочного описания GPSS* том, как определить их.

"Эксперимент не может вызывать процедуру"

Хотя Эксперимент - своего рода PLUS процедура, Вы должны использовать команду Conduct, чтобы вызвать его.

"Параметр В должен быть больше чем параметр А"

Когда параметры были оценены, параметр В не был больше чем параметр А. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Параметр не может использоваться как временная матрица: XXX"

Вы не можете использовать то же самое название для обоих параметров и локальных переменных.

"Параметр не может использоваться как временный: XXX"

Вы не можете использовать то же самое название для обоих параметров и локальных переменных.

"Арифметическое переполнение"

Результат операции превышает максимальное допустимое значение. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Устраните порядковую ошибку"

Элемент массива не существует. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Блок был рассчитан неположительно"

Оценка расчета блока привела к 0 или отрицательному значению. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Попытайтесь упустить неактивное устройство"

Активная заявка пытается упустить устройство, которое она ни не имеет, и была выгружена. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Попытайтесь упустить не расположенное в одном устройство"

Активная заявка пытается упустить устройство, которое ей не принадлежит. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Попытайтесь освободить больше памяти, чем имеет место"

Активная заявка пытается заставить объект памяти использовать больше полезной мощности, чем было определено в команде STORAGE. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Индекс блока является слишком большим"

Блочное местоположение, которое было вычислено, превышает число блоков в модели. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Метки блока не могут использоваться в другом месте"

Блочные метки не могут использоваться как названия других объектов.

"Предел блока является слишком маленьким"

Предел блока достижим перед начальным местоположением блока. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Переполнение стека запросов. Возможен бесконечный цикл. Размер стека увеличен в параметрах настройки"

Величина вложения вызовов процедуры превысила позволенный максимум. Максимум может быть установлен в ПК параметрах настройки. Возможно, что одна из Ваших PLUS процедур вызывает одну из процедур, формируя бесконечный цикл.

"Не может перейти на локальные страницы PLUS массива"

Запрос к памяти блокирован. Максимальная память запрашивает в моделированной странице ПК параметрах настройки. Убедитесь, что у Вас есть достаточно пространства перестановки на Вашем жестком диске.

"Не может получить память для текстовой строки"

Запрос к памяти блокирован. Максимальная память запрашивает установку в моделированной странице ПК параметрах настройки. Убедитесь, что у Вас есть достаточно свободного пространства на жестком диске.

"Невозможно получить числовой элемент группы"

Запрос к памяти блокирован. Максимальная память запрашивает установку в моделированной странице ПК параметрах настройки. Убедитесь, что у Вас есть достаточно свободного пространства на жестком диске.

"Не удается получить параметр блока"

Запрос к памяти блокирован. Максимальная память запрашивает установку в моделированной странице ПК параметрах настройки. Убедитесь, что у Вас есть достаточно свободного пространства на жестком диске.

"Не удается получить операционный параметр"

Запрос к памяти блокирован. Максимальная память запрашивает установку в моделированной странице ПК параметрах настройки. Убедитесь, что у Вас есть достаточно свободного пространства на жестком диске.

"Невозможно получить операционный элемент, стоящий в очереди"

Запрос к памяти блокирован. Максимальная память запрашивает установку в моделированной странице ПК параметрах настройки. Убедитесь, что у Вас есть достаточно свободного пространства на жестком диске.

"Не удается получить заявку".

Запрос к памяти блокирован. Максимальная память запрашивает установку в моделированной странице ПК параметрах настройки. Убедитесь, что у Вас есть достаточно свободного пространства на жестком диске.

"Не удается найти функцию в указанном модуле".

Динамический функциональный запрос нашел файл, но не нашел функцию в пределах этого модуля. Пожалуйста, ознакомьтесь с главой 8 для подробностей относительно использования Call() и связи с директивами.

"Синтаксическая ошибка в операторе".

Запрещенный символ был найден в текстовой строке. Пожалуйста, исправьте это, основываясь на возможностях редактирования текста GPSS.

"Бесконечный цикл в выражениях."

Количество вложений в оценке выражения превысило позволенный максимум. Максимум может быть установлен на странице моделирования ПК параметрах настройки. Возможно, что объекты переменной GPSS и вызовы процедуры формируют бесконечный цикл, который Вы должны предотвратить. Эта ошибка обычно вызывается бесконечным циклом во время оценки выражения. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку. Вы можете назначить промежуточное значение на Savevalue или операционный Параметр.

"Запятая необходима между координатами".

Матричные координаты элемента должны быть отделены запятыми.

"Сопряженный блок не является блоком соответствия"

Активная заявка пытается проверить условие соответствия в блоке, который не является блоком соответствия. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Общая вероятность должна быть 1.0 в последнем пункте"

Последняя общая вероятность в функциональном операторе повторителя должна быть 1.0.

"Числа потока данных должны быть строго положительными"

Вы не можете использовать отрицательное число для потока данных. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Данная величина не была инициализирована"

Вы пытаетесь использовать переменную в операции прежде, чем задали начальное значение. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Декремент не определен для этого типа данных"

Не удалось произвести вычисление, потому что операнд принял значение, которое не может быть применено к арифметическому значению. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Разделитель не определен для этого типа данных"

Не удалось произвести вычисление, потому что операнд принял значение, которое не может быть применено к арифметическому значению. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Переполнение разряда"

Результат операции превышает максимальное допустимое значение. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Ошибка диапазона разряда"

Один из операндов принял значение, для которого не определена операция. Попытка к делению на ноль может вызвать эту ошибку. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"DoCommand получил недопустимую команду"

Строка прошла к DoCommand (), директива не содержала правильную команду GPSS. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Объект не переменная"

Активная заявка попыталась оценить переменную объекта, который не существует. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Ошибка в операнде QUEUE"

Оценка операнда блока QUEUE привела к отрицательному значению. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Ошибка произошла при открытии модели"

Ошибка произошла, когда GPSS попытался прочитать файл, связанный с объектом моделирования. Повторите операцию. Если ошибка повторяется, у Вас может возникнуть аппаратная ошибка. Файл может быть поврежден.

"Ошибка произошла при сохранении модели"

Ошибка произошла, когда GPSS попытался записать файл, связанный с объектом моделирования. Повторите операцию. Если ошибка повторяется, у Вас может возникнуть аппаратная ошибка. Файл может быть поврежден.

"Ошибка данных в буфере обмена. Пожалуйста, повторите попытку"

Операция в буфере обмена не удалась. Повторите операцию. Если ошибка повторяется, Вам необходимо искать ошибку в операционной системе.

"Ошибка заполнения блока GENERATE, связанная с его первой заявкой"

GPSS не смог инициализировать блок GENERATE, связанный с его первой заявкой. Появилась ошибка в вычислении времени прибытия, или в одном из других операндов блока GENERATE.

"Ошибка при чтении открытого текстового файла - включения"

Ошибка произошла, когда GPSS попытался прочитать файл, связанный с текстовым объектом. Повторите операцию. Если ошибка повторяется, у Вас может возникнуть аппаратная ошибка. Файл может быть поврежден.

"Ошибка остановки"

Ошибочное условие остановило операцию. Дополнительные подробности Вы найдете в сопроводительных сообщениях.

"Ошибка: поврежден объект моделирования"

Ошибка произошла, когда GPSS попытался прочитать файл, связанный с объектом моделирования. Данные в файле имеют неизвестный формат.

"Ошибка: уменьшение общих вероятностей"

Общая вероятность в функции следования не может уменьшаться.

"Ожидание =."

Произошла синтаксическая ошибка. В его текущем состоянии транслятор ищет плавающую точку или период. Если это – модель преобразования, Вы можете использовать **поиск / следующая ошибка** или **поиск / предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Ожидание метки"

Синтаксическая ошибка произошла. В его текущем состоянии транслятор ищет метку. Если это - модель преобразования, Вы можете использовать **поиск / следующая ошибка** или **поиск / предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Ожидание 'DO' "

"Ожидание формального параметра"

Синтаксическая ошибка произошла. В его текущем состоянии транслятор ищет параметр в вызове процедуры. Если это - модель преобразования, Вы можете использовать **поиск / следующая ошибка** или **поиск /предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Ожидание числового значения или идентификатора"

Синтаксическая ошибка произошла. В его текущем состоянии транслятор ищет числовое значение или идентификатор. Если это - модель преобразования, Вы можете использовать **поиск / следующая ошибка** или **поиск /предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Ожидание числового значения"

Синтаксическая ошибка произошла. В его текущем состоянии транслятор ищет числовое значение. Если это - модель преобразования, Вы можете использовать **поиск / следующая ошибка** или **поиск /предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Ожидание правой круглой скобки"

Синтаксическая ошибка произошла. В его текущем состоянии транслятор ищет символ правой круглой скобки. Если это - модель преобразования, Вы можете использовать **поиск / следующая ошибка** или **поиск /предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Ожидание эксперимента, процедуры или оператора GPSS"

Синтаксическая ошибка произошла. В его текущем состоянии транслятор ищет один из элементов в сообщении об ошибках. Если это - модель преобразования, Вы можете использовать **поиск / следующая ошибка** или **поиск /предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Ожидание фактора выражения"

Синтаксическая ошибка произошла. В его текущем состоянии транслятор ищет правильный фактор выражения. Формальная грамматика дана в приложении *Справочного описания GPSS*. Если это - модель преобразования, Вы можете использовать **поиск / следующая ошибка** или **поиск /предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Ожидание левой круглой скобки"

Синтаксическая ошибка произошла. В его текущем состоянии транслятор ищет символ левой круглой скобки. Если это - модель преобразования, Вы можете использовать **поиск / следующая ошибка** или **поиск /предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Ожидание левой квадратной скобки '['"

Синтаксическая ошибка произошла. В его текущем состоянии транслятор ищет символ '[' левой квадратной скобки. Если это - модель преобразования, Вы можете использовать **поиск / следующая ошибка** или **поиск /предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Ожидание правой круглой скобки ']' "

Синтаксическая ошибка произошла. В его текущем состоянии транслятор ищет символ ']' правой круглой скобки. Если это - модель преобразования, Вы можете использовать **поиск / следующая ошибка** или **поиск /предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Ожидание правой квадратной скобки ']' "

Синтаксическая ошибка произошла. В его текущем состоянии транслятор ищет символ ']' правой квадратной скобки. Если это - модель преобразования, Вы можете использовать **поиск / следующая ошибка** или **поиск /предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Ожидание точки с запятой"

Синтаксическая ошибка произошла. В его текущем состоянии транслятор не находит ';' символ. Если это - модель преобразования, Вы можете использовать **поиск / следующая ошибка** или **поиск /предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Эксперимент прерывался"

Моделирование было прервано, в то время как PLUS эксперимент выполнялся. Обычно это означает, что у матрицы результата нет необходимых данных. Вы должны завершить сохранение. Эксперименты, сгенерированные автоматически тестом GPSS на завершение, выполняются и могут быть перезапущены командой Conduct.

"Эксперименты не могут быть вложены"

PLUS эксперименты могут вызываться непосредственно командой Conduct.

"Слишком большая экспонента"

Экспонента превышает максимальное допустимое значение. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Экспоненциальное переполнение"

Результат операции превышает максимальное допустимое значение. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Память выражения полна"

Для сохранения выражений больше нет участка памяти. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Коэффициенты должны последовательно начинаться с А"

Экспериментальные коэффициенты обозначаются последовательно, с начиная А, В, С, D, Е, и F.

"Сбой, загружающий указанный модуль"

Динамический функциональный запрос не мог найти необходимый файл. Пожалуйста, ознакомьтесь с главой 8 для подробностей относительно использования Call (), и связал библиотечные процедуры.

"Ошибка при закрытии файла"

Попытка закрыть поврежденный файл. Это может вызвать аппаратную ошибку.

"Ошибка файла"

Ошибка препятствовала тому, чтобы операция файла завершилась должным образом.

"При интеграции операций с плавающей точкой обнаружена ошибка"

Арифметическая операция с плавающей точкой столкнулась с переполнением или потерей значимости. Вы должны масштабировать значения своих переменных, таким образом, чтобы они не превышали числовые пределы.

"Глагол GPSS не может использоваться как оператор или операнд"

Синтаксическая ошибка. Транслятор не находит оператор или операнд, но вместо этого, столкнулся с глаголом GPSS.

"GPSS Студенческой версии превысил предел блока GPSS"

Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Отказ системы помощи"

Отказ произошел, инициализируя систему интерактивной справки. Файл помощи GPSS может отсутствовать.

"Идентификатор не был определен"

Ссылки на имя, которое не было определено, Вызвали ошибку. Вы должны добавить соответствующее определение объекту модели и/или моделирования.

"Запрещенная попытка переполнить отрицательными значениями объект Queue"

Заявка попыталась ввести Queue или DEPART из блока, который был переполнен отрицательными значениями. Этому недопустимо. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Запрещенный символ"

Транслятор столкнулся с запрещенным символом. У вашего оператора есть синтаксическая ошибка. В окне модели Вы можете использовать **поиск / следующая ошибка** или **поиск / предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Запрещенная комбинация операндов в блоке ALTER"

Вы использовали запрещенную комбинацию операндов в ALTER блоке. Пожалуйста, ознакомьтесь с главой 7 данного руководства для подробностей относительно использования ALTER –блоков

"Запрещенная комбинация операндов в REMOVE Блок"

Вы использовали запрещенную комбинацию операндов в REMOVE блоке. Пожалуйста, ознакомьтесь с главой 7 данного руководства для подробностей относительно использования REMOVE –блоков

"Запрещенная комбинация операндов в UNLINK Блок"

Вы использовали запрещенную комбинацию операндов в UNLINK блоке. Пожалуйста, ознакомьтесь с главой 7 данного руководства для подробностей относительно использования UNLINK-блоков

"Запрещенная комбинация операторов"

Вы использовали запрещенную комбинацию операторов. Пожалуйста, ознакомьтесь с главой 7 данного руководства для подробностей относительно использования.

"Запрещенный класс протокола SNA в блоке COUNT ИЛИ SELECT"

Вы использовали запрещенный класс протокола SNA операнда E COUNT или SELECT блока. Пожалуйста, ознакомьтесь с главой 7 этого руководства для подробностей относительно использования этих блоков.

"Файл -включения файлов должен быть текстовым объектом (.txt)"

Файлы файла-включения должны быть файлами с расширением имени .txt .

"Несовместимые форматы. Архив и эта версия программного обеспечения GPSS используют **различные форматы**"

Вы пытаетесь прочитать файл, связанный с объектом GPSS другого формата, нежели формат программного обеспечения GPSS, которое Вы используете.

Для объектов моделирования или модели Вы должны обновить их , используя новую версию GPSS. Для объектов, используйте буфер обмена Windows со старой версией GPSS , чтобы создать текстовые файлы для каждой модели. Тогда, используйте буфер обмена Windows с новой версией GPSS, чтобы создать новые объекты моделирования из текстовых файлов.

"Некорректное ключевое слово"

Вы использовали ключевое слово, которое не принадлежит той области, где Вы его использовали это. У Вашего оператора есть синтаксическая ошибка. В окне модели Вы можете использовать **поиск / следующая ошибка** или **поиск / предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Неправильная матричная форма"

GPSS столкнулся с матрицей неправильной формы, так как пытается обратиться через поверхностный отклик к данным. Если матричный оператор был сгенерирован автоматическим генератором экспериментов языка GPSS не был изменен, пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Инкремент не определен для этого типа данных"

Не удалось произвести вычисления, так как операнд принял значение, которое не может быть применено к арифметическому значению. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Косвенный протокол SNA запрещен"

Вы не можете использовать системный числовой атрибут с косвенным спецификатором объекта как этот операнд. У Вашего оператора есть синтаксическая ошибка. В окне модели Вы можете использовать **поиск / следующая ошибка** или **поиск / предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

Косвенная адресация обсуждается в разделе 3.4.2 *Справочного описания GPSS*.

"Целочисленное переполнение разряда"

Результат операции превышает максимальное допустимое значение. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Ошибка целочисленного диапазона разряда"

Один из операндов принял значение, для которого не определена операция. Попытка к делению на ноль может вызвать эту ошибку. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Допуск интеграции не может быть поддержан для: XXX"

Во время интеграции переменной, данной в сообщении об ошибках, уменьшение размера минусага интеграции не было достаточно для того, чтобы переместить местную ошибку в пределы ошибочного допуска. Вы должны или увеличить ошибочный допуск в моделированной странице ПК параметров настройки, или Вы должны создать свое собственное использование метода интеграции PLUS процедуры. Это возможно, так как производная переменных интеграции является слишком большой по величине.

"Переменная интеграции не была инициализирована: XXX"

Когда проинтегрировали переменную, упомянутую в сообщении об ошибках, как выяснилось, у неё не было никакого начального значения. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Внутренняя ошибка компилятора. Неизвестный тип источника"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Внутренняя ошибка"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Недопустимый символ \$ в названии"

Ваш оператор содержит синтаксическую ошибку. Запрещенный символ был обнаружен в идентификаторе. В окне модели Вы можете использовать **поиск / следующая ошибка** или **поиск / предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

Названия обсуждаются в разделе 3.5 *Справочного описания GPSS*.

"Недопустимая попытка включить GENERATE блок"

Заявкам не позволено включать, GENERATE блоки, за исключением блоков, созданных непосредственно ими. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Недопустимая попытка пропустить необходимый операнд"

У Вашего оператора есть синтаксическая ошибка. В окне модели Вы можете использовать **поиск / следующая ошибка** или **поиск / предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Запрещенный символ в операторе языка GPSS"

Ваш оператор содержит синтаксическую ошибку. Обнаружен запрещенный символ. В окне модели Вы можете использовать **поиск / следующая ошибка** или **поиск / предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Запрещенный символ в экспоненте"

Ваш оператор содержит синтаксическую ошибку. Обнаружен запрещенный символ. В окне модели Вы можете использовать **поиск / следующая ошибка** или **поиск / предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Запрещенный символ в числе"

Ваш оператор содержит синтаксическую ошибку. Обнаружен запрещенный символ. В окне модели Вы можете использовать **поиск / следующая ошибка** или **поиск / предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Запрещенный символ"

Ваш оператор содержит синтаксическую ошибку. Обнаружен запрещенный символ. В окне модели Вы можете использовать **поиск / следующая ошибка** или **поиск / предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимая координата столбца. Должно быть имя или целое число"

У Вашего Оператора есть синтаксическая ошибка. Матричные индексы элемента должны быть именами или целыми числами. В окне модели Вы можете использовать **поиск / следующая ошибка** или **поиск / предыдущая ошибка** переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете

получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Определите тип данных"

Не удалось произвести вычисление, потому что операнд взял значение, которое не может быть применено к арифметическому выражению. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Недопустимая десятичная точка в числе"

У Вашего оператора есть синтаксическая ошибка. Столкнулись с лишней десятичной точкой. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимый разделитель"

У Вашего Оператора есть синтаксическая ошибка. Столкнулись с недопустимым символом - разделитель. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимый разделитель в операторе языка GPSS"

У Вашего Оператора есть синтаксическая ошибка. Столкнулись с недопустимым символом - разделитель. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимый долларовый символ"

Ваш оператор содержит синтаксическую ошибку. Столкнулись с недопустимым символом '\$'. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Показательная степень числа некорректна"

Ваш оператор содержит синтаксическую ошибку. Столкнулись с недопустимым символом 'E'. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно в ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимое выражение"

Выражение было неправильно сформировано. Вы должны исправить синтаксис. Формальная грамматика дана в Приложении *Справочного описания GPSS*.

"Недопустимый функциональный счет пары"

Вы нарушили одно из правил относительно определения объектов функции языка GPSS. Пожалуйста, изучите раздел 4.4 *Справочного описания GPSS* и главы 6 Функции Следования.

"Недопустимый тип функции"

Вы нарушили одно из правил относительно определения Объектов Функции языка GPSS. Пожалуйста, изучите раздел 4.4 *Справочного описания GPSS* и главы 6 Функции Следования.

"Недопустимый флаг"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимое ключевое слово для этого операнда"

Ваш оператор содержит синтаксическую ошибку. Ключевое слово недопустимо или неуместно. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимое ключевое слово"

Ваш оператор содержит синтаксическую ошибку. Ключевое слово недопустимо или неуместно. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимый порядковый номер списка"

Список порядковых номеров Функционального следования некорректно. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимое название в операторе языка GPSS"

Ваш оператор содержит синтаксическую ошибку. Название недопустимо или неуместно. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

Названия обсуждаются в разделе 3.5 *Справочного описания GPSS*.

"Недопустимое отрицательное число в операторе GPSS"

Ваш оператор содержит синтаксическую ошибку. Отрицательное число недопустимо или неуместно. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимое количество индексов матрицы"

Размерность матрицы, подразумеваемой количеством индексов в матричном спецификаторе элемента, не соответствует размерности команды MATRIX . Эти два оператора должны иметь одну и ту же матричную размерность.

"Недопустимое число параметров процедуры"

У процедуры запроса нет числа параметров требуемых PLUS определением процедуры. Эти два оператора должны согласоваться.

"Недопустимое номер"

Ваш оператор содержит синтаксическую ошибку. Номер недопустим или неуместен. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимый оператор"

Ваш оператор содержит синтаксическую ошибку. Оператор недопустим или неуместен. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимая PLUS процедура"

Ваш оператор содержит синтаксическую ошибку в определении процедуры. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимое определение процедуры"

Ваш оператор содержит синтаксическую ошибку в определении процедуры. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимый идентификатор процедуры"

Ваш оператор содержит синтаксическую ошибку. Идентификатор процедуры является неправильным или неуместным. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимый указатель оператора"

Ваш оператор содержит синтаксическую ошибку. Указатель оператора является неправильным или неуместным. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимое вещественное число в GPSS операторе"

Ваш оператор содержит синтаксическую ошибку. Вещественное число не может использоваться в этой позиции. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимая координата строки"

Ваш оператор содержит синтаксическую ошибку. Матричные индексы элемента должны быть именами или целыми числами. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимый класс протокола SNA"

Ваш оператор содержит синтаксическую ошибку. Системный числовой атрибут класса недопустим или неуместен. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

Классы протокола SNA рассматриваются в разделе 3.4 *Справочного описания GPSS*.

"Недопустимый спецификатор объекта протокола SNA"

Ваш оператор содержит синтаксическую ошибку. Системный числовой атрибут недопустим или неуместен. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно

ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

Протоколы SNA рассматриваются в разделе 3.4 *Справочного описания GPSS*.

"Недопустимый протокол SNA в операторе GPSS"

Ваш оператор содержит синтаксическую ошибку. Системный числовой атрибут недопустим или неуместен. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

Протоколы SNA рассматриваются в разделе 3.4 *Справочного описания GPSS*.

"Недопустимый протокол SNA"

Ваш оператор содержит синтаксическую ошибку. Системный числовой атрибут недопустим или неуместен. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

Протоколы SNA рассматриваются в разделе 3.4 *Справочного описания GPSS*.

"Недопустимая строка в операторе GPSS"

Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Недопустимая строка"

Ваш оператор содержит синтаксическую ошибку. Строка недопустима. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"При интеграции найден недопустимый пороговый блок"

Ошибка произошла во время интеграции переменной, упомянутой в сообщении. Пороговое пересечение произошло, но номер блока адресата не попал в необходимый диапазон. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Недопустимое использование Спецификатора Класа протокола SNA"

Ваш оператор содержит синтаксическую ошибку. Системный числовой атрибут недопустим или неуместен. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

Протоколы SNA рассматриваются в разделе 3.4 *Справочного описания GPSS*.

"Недопустимый XXX символ в YYY"

Ваш оператор содержит синтаксическую ошибку. Символ, данный в сообщении, недопустим или неуместен в выражении. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Недопустимое XXX Выражение."

Ваш оператор содержит синтаксическую ошибку. Выражение, данное в сообщении, недопустимо или неуместно. В окне модели Вы можете использовать поиск / следующая

ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

Выражения рассматриваются в разделе 8.2 *Справочного описания GPSS..*

"Недопустимое XXX имя"

Ваш оператор содержит синтаксическую ошибку. Выражение, данное в сообщении, недопустимо или неуместно. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

Выражения рассматриваются в разделе 8.2 *Справочного описания GPSS..*

"Недопустимое значение Y"

Значение Y в функции следования недопустимо или неуместно. Функции следования обсуждаются в главе 6.

"Номер строки должен быть строго положительным целым числом"

Ваш оператор содержит синтаксическую ошибку. Номер строки недопустимо или неуместно. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Выстраивание строк должно начинаться в первом столбце"

Ваш оператор содержит синтаксическую ошибку. Номер недопустим или неуместен. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Параметр List Function является слишком большим"

Когда параметр функции списка был оценен, он превысил размер списка. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Блокировка нарушений"

Блокировка совместного использования файлов. Повторите операцию, когда файл доступен.

"Ошибка диапазона логарифма"

Операнд может быть отрицательным. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Отказ программного интерфейса почтовых приложений. Электронная почта не отправилась"

Ваша система программного интерфейса почтовых приложений не могла быть инициализирована, или ошибка произошла, когда передача сообщения была предпринята. Сообщение не отправлено.

"Не удалось инициализировать программный интерфейс почтовых приложений. Электронная почта не отправлена"

Ваша система программного интерфейса почтовых приложений не могла быть инициализирована, или ошибка произошла, когда передача сообщения была предпринята. Сообщение не отправлено.

"Оператор ручного моделирования"

Ручной оператор моделирования столкнулся с Error Stop. Дополнительные подробности следуют за этим сообщением.

"Матричный столбец не может быть назначен на систему или блочную метку"

Матричный столбец определен не корректно. Измените Матричный спецификатор элемента в Операторе.

"Отказ инверсии матрицы"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Максимальное значение превышено: возможен бесконечный цикл. Увеличьте максимальное значение (Max Depth) в параметрах настройки"

Количество вложений в оценке Выражения превысило позволенный максимум. Максимум может быть установлен на странице моделирования ПК параметрах настройки. Возможно, что объекты переменной GPSS и вызовы процедуры формируют бесконечный цикл, который Вы должны предотвратить.

Эта ошибка обычно вызывается бесконечным циклом во время оценки выражения. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку. Вы можете назначить промежуточное значение на Savevalue или операционный Параметр.

"Ошибка памяти"

Запрос к памяти блокирован. Максимальная память должна быть определена на странице модели ПК параметрах настройки. Убедитесь, что у Вас есть достаточно свободного места на Вашем жестком диске (дисках).

" Запрос к памяти блокирован"

Запрос к памяти блокирован. Максимальная память должна быть определена на странице модели ПК параметрах настройки. Убедитесь, что у Вас есть достаточно свободного места на Вашем жестком диске (дисках).

"Минимум и максимум должны быть числовыми"

У минимума или максимума должно быть числовое значение. Пожалуйста, исправьте оператора и повторите операцию.

"Максимум должен быть больше чем минимум"

У Вас не может быть минимальное значение больше, чем максимальное. Пожалуйста, исправьте Оператора и повторите операцию.

"Отсутствуют операнды GPSS"

Ваш оператор содержит синтаксическую ошибку. Транслятор не находит элемент в сообщении. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Ввести отсутствующие операнды GPSS"

Ваш оператор содержит синтаксическую ошибку. Транслятор не находит элемент в сообщении. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Отсутствует объект спецификатора протокола SNA"

Ваш оператор содержит синтаксическую ошибку. Транслятор не находит элемент в сообщении. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Отсутствуют метки оператора"

Ваш оператор содержит синтаксическую ошибку. Транслятор не находит элемент в сообщении. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если

ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Пропущены координаты для протокола SNA Класса MX"

Ваш оператор содержит синтаксическую ошибку. Транслятор не находит элемент в сообщении. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Не удалось перевести модель; за пределами памяти"

Запрос к памяти блокирован. Максимальная память должна быть определена на странице модели ПК параметрах настройки. Убедитесь, что у Вас есть достаточно свободного места на Вашем жестком диске (дисках).

"Раздел модуля не определен для этого типа данных"

Не удалось произвести вычисление, потому что операнд принял значение, которое не может быть применено к арифметическому значению, или сделана попытка деления на ноль. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Ошибка диапазона при разделе модуля"

Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Переполнение при умножения."

Результат операции превышает максимальное допустимое значение. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Множитель определил блочную метку определения"

Ваш оператор содержит синтаксическую ошибку. Блочная метка может использоваться только для одного блока. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Множитель определил метку: XXX"

Ваш оператор содержит синтаксическую ошибку. Метка, упомянутая в сообщении, не была определена. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Множественные определения параметра: XXX"

Ваш оператор содержит синтаксическую ошибку. Параметр, упомянутый в сообщении, не был определен. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Умножение не определено для этого типа данных"

Не удалось произвести Вычисление, потому что операнд принял значение, которое не может быть применено к арифметическому значению, или сделана попытка деления на ноль. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Многоканальный эксперимент не является ортогональным. Не доступна статистическая величина F"

Выполненные результаты в матрице результата не формируют ортогональный эксперимент. Вы должны или завершить эксперимент, или переопределить его, если сам эксперимент не является ортогональным.

"Значению не присвоили имя"

Столкнулись с идентификатором uninitialized. Если Вы не присваиваете имя значению в операторе EQU, GPSS назначает "системное" число для использования в определении объектов GPSS. Имена, данные системой могут использоваться чтобы назвать объекты, но, скорее всего не будут использоваться в выражениях.

Не путайте имена значений с результатом протокола SNA. Например, если Вы определяете переменный объект под названием VAR1, Вы оцениваете его справочной информацией на V\$VAR1 а не на VAR1. Здесь VAR1 служит спецификатором объекта.

"Название слишком велико"

Ваш оператор содержит синтаксическую ошибку. Названия не могут превысить 200 символов. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

Имена рассматриваются в разделе 3.5 *Справочного описания GPSS*.

"Отрицательный операнд A в блоке SPLIT"

Активная заявка пытается операнд блока SPLIT оценить с недопустимым отрицательным результатом. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Отрицательный запрос памяти"

Когда операнд B ENTER или LEAVE были оценены, количество памяти, которое они требуют, было меньше нуля. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Отрицательное приращение времени"

Активная заявка пыталась вычислить отрицательное приращение времени. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Отсутствуют данные. Ошибка произошла в пользовательском диалоговом окне"

Вы должны завершить ввод необходимых данных в поле диалогового окна.

"Отсутствует участок памяти для преобразования числа с плавающей точкой"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Отрицательный параметр функции List"

Оцененный параметр функции List отрицательный. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Отрицательный модуль"

Модуль должен быть положительным. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Недостаточно памяти для перевода модели"

Запрос к памяти блокирован. Убедитесь, что у Вас есть достаточно свободного места на Вашем жестком диске (дисках).

"Код объекта поврежден"

Пожалуйста, снова переведите Модель. Если ошибка сохраняется, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Операнд B QUEUE или DEPART отрицательный"

Активная заявка вызвала операнд B из блока QUEUE или DEPART, он будет оценен с недопустимым отрицательным значением. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Операция не определена для этого типа данных"

Не удалось произвести Вычисление, потому что операнд принял значение, которое не может быть применено к арифметическому значению. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Недостаточно памяти"

Запрос к памяти блокирован. Максимальная память должна быть определена на странице модели ПК параметрах настройки. Убедитесь, что у Вас есть достаточно свободного места на Вашем жестком диске (дисках).

"Номер параметра должен быть положительным целым числом"

Номер заявки определен как отрицательное число. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Путь не найден"

Ввод операции вывода попытался использовать путь, который не существует в вашей файловой системе. Пожалуйста, исправьте ситуацию и повторите операцию.

"Графический предел превышен"

Вы не можете больше добавить графики к этому графическому окну. Вы можете открыть дополнительное графическое (PLOT) окно.

"PLUS массив не был определен"

Справочная информация была определена на PLUS массив, который не был определен.

"PLUS процедура не найдена"

Вы должны определить PLUS процедуре прежде, чем вы вызовете ее.

"Вероятность не может быть меньше чем 0"

Вероятность не может быть меньше чем 0 или больше чем 1. Это описано в *Справочном описании GPSS*.

"Вероятность не может превысить 1.0"

Вероятность не может быть меньше чем 0 или больше чем 1. Это описано в *Справочном описании GPSS*.

"Предел оператора процедуры превышен"

Процедура является слишком большой. Вы должны разделить ее на несколько процедур.

"Процедура заменена. Повторное определение"

Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Оценка направлена на несуществующий объект блока"

Активная заявка вызвала справочную информацию на объект блока, который не существует. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Оценка направлена на несуществующий объект функции"

Активная заявка вызвала справочную информацию на объект функции, которая не существует. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку. Вы, возможно, забыли определить объект функции.

"Оценка направлена на несуществующий объект матрицы"

Активная заявка вызвала справочную информацию на объект матрицы, который не существует. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку. Вы, возможно, забыли определить объект матрицы.

"Оценка направлена на несуществующий параметр"

Активная заявка вызвала справочную информацию на параметр заявки, которая не существует. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку. Параметры заявки созданы в SUB блоках ASSIGN, MARK, SPLIT и TRANSFER.

"Оценка направлена на несуществующий объект Savevalue"

Активная заявка вызвала справочную информацию на объект **Savevalue**, который не существует. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку. Вы, возможно, забыли определить объект **Savevalue**.

"Оценка направлена на несуществующий объект памяти"

Активная заявка вызвала справочную информацию на объект памяти, которая не существует. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку. Вы, возможно, забыли определить объект памяти.

"Оценка направлена на несуществующий объект таблицы"

Активная заявка вызвала справочную информацию на объект таблицы, которая не существует. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку. Вы, возможно, забыли определить объект таблицы с TABLE или QTABLE оператором

"Оценка направлена на несуществующий объект переменной"

Активная заявка вызвала справочную информацию на объект переменной, которая не существует. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку. Вы, возможно, забыли определить объект переменной с VARIABLE, BVARIABLE или FVARIABLE оператором

"Опция REmove не определена"

Вы определили блок адресата, не используя необходимую опцию RE. Пожалуйста, ознакомьтесь с описанием блока оператора в главе 8 с описанием большого количества подробностей.

"Опция REmove использовалась без адресата"

Вы должны определить местоположение блока адресата, когда Вы используете опцию RE. Пожалуйста, ознакомьтесь с описанием блока оператора в главе 8 с описанием большого количества подробностей.

"Ошибка установки"

Ошибка произошла во время инициализации GPSS. Если ошибка сохраняется, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Share.exe не был загружен или общедоступная область была заблокирована"

Ошибка операционной системы. Если ошибка сохраняется, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Извините Текстовый файл не может быть открыт"

Ошибка произошла, когда GPSS попытался читать текстовый файл. Повторите операцию. Если ошибка сохраняется, возможно, у Вас может быть аппаратная ошибка или ошибка операционной системой. Файл может быть поврежден.

"Извините модель не может быть открыта"

Ошибка произошла, когда GPSS попытался читать файл, связанный с объектом моделирования. Повторите операцию. Если ошибка сохраняется, возможно, у Вас может быть аппаратная ошибка или ошибка операционной системой. Файл может быть поврежден.

"Извините модель не может быть сохранена"

Ошибка произошла, когда GPSS попытался записать файл, связанный с объектом моделирования. Повторите операцию. Если ошибка сохраняется, возможно, у Вас может быть аппаратная ошибка или ошибка операционной системой.

"Извините, сообщение не может быть открыто"

Ошибка произошла, когда GPSS попытался читать файл, связанный с отчетом моделирования. Повторите операцию. Если ошибка сохраняется, возможно, у Вас может быть аппаратная ошибка или ошибка операционной системой. Файл может быть поврежден.

"Извините, сообщение не может быть сохранено"

Ошибка произошла, когда GPSS попытался записать файл, связанный с отчетом моделирования. Повторите операцию. Если ошибка сохраняется, возможно, у Вас может быть аппаратная ошибка или ошибка операционной системой.

"Извините, матрицы не определены для моделирования"

Вы не можете открыть Окно матрицы моделирования, потому что матричные объекты не определены в Объекте Моделирования.

"Извините, таблицы не определены для моделирования"

Вы не можете открыть Окно таблицы моделирования, потому что таблицы или объекты Qtable не определены в Объекте Моделирования.

"Извините, моделирование не может произойти"

Ошибка произошла, когда GPSS попытался читать файл, связанный с объектом моделирования. Повторите операцию. Если ошибка сохраняется, возможно, у Вас может быть аппаратная ошибка или ошибка операционной системой. Файл может быть поврежден.

"Извините, моделирование не может быть сохранено"

Ошибка произошла, когда GPSS попытался записать файл, связанный с объектом моделирования. Повторите операцию. Если ошибка сохраняется, возможно, у Вас может быть аппаратная ошибка или ошибка операционной системой.

"Извините, текстовый документ не может быть сохранен"

Ошибка произошла, когда GPSS попытался записать текстовый файл, связанный с объектом моделирования. Повторите операцию. Если ошибка сохраняется, возможно, у Вас может быть аппаратная ошибка или ошибка операционной системой.

"Квадратный корень не определен для этого типа данных"

Не удалось произвести вычисление, потому что операнд принял значение, которое не может быть применено к арифметическому выражению или которое является отрицательным. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Ошибка диапазона квадратного корня"

Параметр не может быть отрицательным. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Стандартная ошибка 0. Не может выполнить тест F"

Вычисление для теста F не может быть произведено, потому что произошла стандартная ошибка эксперимента 0.

"Запрос к памяти превышает суммарную мощность объекта памяти"

Когда операнд В оператора ENTER был оценен, запрос к памяти превысил суммарную мощность объекта памяти. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку. Вы можете желать переопределить объект памяти.

"Вычитание не определено для этого типа данных"

Не удалось произвести вычисление, потому что операнд принял значение, которое не может быть применено к арифметическому выражению. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Переполнение результата вычитания"

Результат операции превышает максимальное допустимое значение. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Синтаксическая ошибка в метке"

Ваш оператор содержит синтаксическую ошибку. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Синтаксическая ошибка в операнде А"

Ваш оператор содержит синтаксическую ошибку. Транслятор не находит элемент в сообщении. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если

"Синтаксическая ошибка в операнде В"

"Синтаксическая ошибка в операнде С"

"Синтаксическая ошибка в операнде D"

"Синтаксическая ошибка в операнде E"

"Синтаксическая ошибка в операнде F"

"Синтаксическая ошибка в операнде G"

"Синтаксическая ошибка в операнде N"

"Синтаксическая ошибка в операторе"

"Синтаксическая ошибка"

Ваш оператор содержит синтаксическую ошибку. Транслятор не находит элемент в сообщении. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Синтаксическая ошибка. XXX"

Ваш оператор содержит синтаксическую ошибку в выражении, упомянутом в сообщении. Транслятор не находит элемент в сообщении. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Синтаксическая ошибка. Запрещенное использование директивы Procedure"

Ваш оператор содержит синтаксическую ошибку. Имя PLUS директивы недопустимым способом. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Синтаксическая ошибка. Неправильное вычисление параметра"

Ваш оператор содержит синтаксическую ошибку. Процедура PLUS содержит вычисление параметра, которое не соответствует вычислению, определенному для процедуры. Имя PLUS директивы недопустимым способом. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Синтаксическая ошибка. Недопустимый PLUS оператор"

Ваш оператор содержит синтаксическую ошибку в PLUS операторе. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Синтаксическая ошибка. Недопустимый идентификатор TEMPORARY MATRIX"

Ваш оператор содержит синтаксическую ошибку в PLUS TEMPORARY MATRIX операторе. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Синтаксическая ошибка. Определена отрицательная размерность"

Ваш оператор содержит синтаксическую ошибку. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Синтаксическая ошибка. Слишком большая размерность матрицы"

Ваш оператор содержит синтаксическую ошибку. Вы определили большую размерность в объекте матрицы, чем действительно существует. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Синтаксическая ошибка. Неожиданное окончание файла"

Транслятор ожидает продолжение языковых элементов, но неожиданно столкнулся с концом модели. Эта ошибка могла произойти, в случае если у Вас есть строка с открытыми двойными кавычками.

"Синтаксическая ошибка. Неизвестный символ"

Ваш оператор содержит синтаксическую ошибку. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Системная ошибка, переводящая взаимодействие"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Осуществимый набор для принципиального блока не мог быть найден"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Ошибка исключения произошла при сохранении моделирования"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Некорректный запрос DEQ"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Некорректный запрос"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Некорректное исправление ошибок"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Некорректная символьная строка"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Повреждено соединение в DEQ"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Повреждено соединение в UnQQE"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. QQE повисло"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Некорректное прямое исправление ошибок"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Функциональная ошибка установки"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Внутреннее исключение"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Внутреннее переполнение"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Недопустимый косвенный указатель"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Недопустимый операнд"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Неизвестный следующий блок"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Неизвестный QQE"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Системная ошибка. Отказ кэша блока управления очередью"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Позиции табуляции должны быть большими и отличными от нуля"

Позиции табуляции должны быть определены как положительные и строго возрастающие целые числа.

"У тестируемого выражения нет числового или логического значения"

Вы можете только использовать числовое значение как результат тестирования Выражения Block. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Требуется XXX поле"

Ваш оператор содержит синтаксическую ошибку. Поле, определенное в сообщении, недопустимо. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Некорректная команда"

Ваш оператор содержит синтаксическую ошибку. Команды рассматриваются в главе 6 в *Справочном описании GPS*.

"Диск полон"

Попытка была записать файл на диск, который не содержит свободного места.

"Динамический запрос должен передавать в качестве параметра 0 или 1"

У динамического запроса не содержит корректных параметров. Call () и директивы процедуры описаны в главе 8 *Справочного описания GPSS*.

"Достигнут конец файла"

Ваш оператор содержит синтаксическую ошибку. Транслятор исчерпал текст прежде, чем оператор был закончен. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

Эта ошибка может быть Вызвана, если у строки нет заключительного двойного символа каВычки.

"Метка объекта отсутствует"

Ваш оператор содержит синтаксическую ошибку. Транслятор не находит элемент в сообщении. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если

ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Код завершения должен быть -1, 0 или +1"

Команда принимает только эти значения в качестве параметра.

"Выражение некорректно"

Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Выражение должно содержать метку"

Выражение не может рассматриваться, быть изображено на графиках или сохранено без метки.

"Файл не найден"

Ошибка произошла, когда GPSS попытался прочитать файл, который операционная система не смогла найти. Местоположение файлов рассматривается в главе 2 *Справочного описания GPSS*, и также это связано с динамическими запросами, описанными в главе 8 *Справочного описания GPSS*.

"Генераторы должны быть независимым w.r.t. умножением модуля. Ни один из них не должен быть продуктом других"

Вы определили группу имен генераторов, которая может совпадать с комбинацией других генераторов. Они должны быть независимыми. Группу имен генераторов рассматриваются в главе 13 *Справочного описания GPSS*.

"J параметр должен быть больше чем i параметр"

Запрашиваемая процедура использовала параметры, которые были некорректны. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Местоположение параметра должно быть больше чем 0.0"

Запрашиваемая процедура использовала параметры, которые были некорректны. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"У матрицы содержит слишком много размерностей"

Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Матричный индекс отрицателен"

Активная заявка пыталась оценить матричный индекс, который равен 0 или отрицателен. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Матричный индекс слишком большой"

Активная заявка пыталась оценить матричный индекс, который превышает размерность матрицы. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Матричное смещение слишком большое"

Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Максимальный параметр должен быть больше чем параметр режима"

Запрашиваемая процедура использовала параметры, которые были некорректны. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку

"Математическое ожидание должно быть больше 0.0"

Запрашиваемая процедура использовала параметры, которые были некорректны. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку

"Параметр режима должен быть больше чем минимальный параметр"

Запрашиваемая процедура использовала параметры, которые были некорректны. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку

"Числовое преобразование не определено"

Вызванная операция запросила числовое преобразование, которое не может быть выполнено. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Вложение файлов файла- включения слишком велико"

Файлы файла- включения могут быть вложены максимум 5 раз.

"Оптимальный тип должен принимать значения -1 для минимума или +1 для максимума"

Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"R параметр должен быть строго между 0 и 1"

Запрашиваемая процедура использовала параметры, которые были некорректны. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку

"Случайный числовой поток должен быть положительным"

Получен отрицательный числовой поток Случайных чисел. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Предел перенаправления должен быть между 0 и 10, включительно"

Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Копируемая размерность не попадает в заданный диапазон"

Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Матриц результата имеет недопустимую форму"

Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Результаты дробного факторного плана не завершены"

Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Параметр масштаба должен быть больше чем 0.0"

Вызванная операция запросила недопустимое значение. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Параметр формирования должен быть больше чем 0.0"

Вызванная операция запросила недопустимое значение. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Моделирование еще выполняется и не может быть сохранено"

Вы должны остановить моделирование прежде, чем сохранить его.

"Моделирование было выполнено Коммерческой версией GPSS"

Студенческая версия GPSS не может открыть объекты моделирования, созданные Коммерческой версией.

"Моделирование было выполнено новой версией GPSS. Эта версия не может прочитать его"

Вы должны создать новый объект Simulation со своей текущей версией GPSS.

"Моделирование не было создано Студенческой версией"

Студенческая версия GPSS не может открыть объекты моделирования, созданные Коммерческой Версией.

"Среднеквадратичное отклонение должно быть больше 0.0"

Вызванная операция запросила недопустимое значение. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Система неспособна преобразовать часть последовательного режима моделирования"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Т параметр должен быть больше 0.0"

Вызванная операция запросила недопустимое значение. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Верхний предел вычисления слишком низок"

Активная заявка пыталась оценить недопустимый операнд. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Пользовательская клавиша не корректна для этого продукта. Пожалуйста, зарегистрируйтесь, чтобы получить новую клавишу"

Вы должны приобрести корректную пользовательскую клавишу от Minuteman Software. К регистрационному диалогу можно обратиться через файл / программное обеспечение регистра в меню основного окна GPSS .

"Отсутствуют заявки. Проверьте операционные пределы и разделение на блоки"

Или Вы ввели оператора, который требует активной заявки, а она отсутствует, или все операторы GENERATE достигли своих пределов.

Если Вы ограничиваете число заявки, созданной при использовании операнда D , GENERATE возвращает оператора, чтобы обнулить его. Вы должны обратиться к другому источнику заявок в моделировании (SPLIT или GENERATE), Вы должны увеличить конечное значение в одном или более TERMINATE блоках, или Вы должны уменьшить конечное значение в операторе START.

"В списке отсутствуют ошибки"

Список ошибок пуст. Возможно, Вы должны снова перевести модель.

"Больше нет записей каталога"

Больше нет файлов в этом каталоге (папке).

"Недостаточно данных, определенных в этом эксперименте"

У программы анализа нет достаточного количества точек на графике, чтобы продолжить анализ. Вы должны завершить выполнение эксперимента прежде, чем проанализировать результаты.

"Отсутствует остановка. Вы хотите удалить все остановки?"

В настоящее время в выбранном блоке нет условия остановки. Тем не менее, есть опция, позволяющая удалить все условия остановки.

"Отсутствует заявка для оценки параметра"

Столкнулись с оператором, который нуждался в одном или более параметрах активной заявки для ее оценки. Однако, нет никакой активной заявки.

"Отсутствует заявка для оценки протокола SNA"

В Протокол SNA отсутствует заявка для его оценки. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Отсутствие эксперимента. DoCommand () недопустим"

Директива DoCommand () не может быть вызвана, если эксперимент не в стадии выполнения. То есть он начат командой Conduct. Во время эксперимента, вызывая PLUS процедуру прямо или косвенно через PLUS эксперимент, можно вызвать директиву DoCommand ().

"Отсутствует заявка"

Определенная Заявка в настоящее время не существует в Объекте Моделирования.

"Для продолжения недостаточно машинной памяти"

Запрос к памяти блокирован. Максимальная память должна быть определена на странице модели ПК параметрах настройки. Убедитесь, что у Вас есть достаточно свободного места на Вашем жестком диске (дисках).

"Должен быть входной блок для порогового выражения"

Вы должны определить блок, который должен получить заявки, которые будут сгенерированы, когда порог скрещен.

"Произошла ошибка переадресации моделирования"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Произошла ошибка инициализации перевода моделирования"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Произошла аппаратная ошибка"

Ваша операционная система сообщила, что аппаратная ошибка препятствовала тому, чтобы GPSS завершил операцию.

"Произошла ошибка при открытии XXX"

Ошибка произошла, когда GPSS попытался прочитать файл, идентифицированный в сообщении. Если ошибка сохраняется, у Вас может возникнуть аппаратная ошибка или ошибка, связанная с операционной системой. Файл может быть поврежден

"Произошла ошибка при попытке установить указатель файла"

Операционная система сообщила, что ошибка произошла, когда GPSS попытался установить указатель файла. Повторите операцию. Если ошибка сохраняется, у Вас может возникнуть аппаратная ошибка или ошибка, связанная с операционной системой. Файл может быть поврежден

"Эта особенность не осуществима"

Функция отсутствует в этой версии GPSS.

"Диапазон времени должен быть числовым и больше 0"

Диапазон времени должен быть положительным числом.

"Установленные допустимые значения должны быть между 0.000000001 и 0.1, включительно"

Допуск интеграции должен быть в данном диапазоне.

"Слишком много блочных объектов для Студенческой версии"

Студенческая Версия поддерживает только 150 или менее блоков GPSS.

"Слишком много открытых файлов"

Ваша операционная система ограничивает число файлов, которые могут быть открытыми. Вы должны закрыть другие приложения и продолжить.

"Слишком много значащих цифр"

Вы превысили максимальное количество значащих цифр в числе. Реальные значения сохранены с точностью до 2 знака после запятой.

"Ошибка домена тригонометрической функции"

Значение операнда не попадает в диапазон, разрешенный для этой функции. Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Сокращение не определено для этого типа данных"

НЕ удалось произвести Вычисление, потому что операнд принял значение, которое не может быть применено к арифметическому значению. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку.

"Невозможно остановить моделирование"

Пожалуйста, закройте все приложения. Повторите операцию. Если эта ошибка сохраняется, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Невозможно установить выражение"

Произошла внутренняя ошибка. Пожалуйста, сообщите об этой ошибке. Для этого, будьте любезны, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Открытая двойная кавычка"

Вы должны закончить строку заключительной двойной кавычкой.

В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Неопределенная метка: XXX"

Ваш оператор содержит синтаксическую ошибку. Метка в сообщении не была определена. Проверьте орфографию. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Неизвестный разделитель"

Ваш оператор содержит синтаксическую ошибку. В текущем состоянии транслятор не находит символ - разделитель, например, запятую. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Неизвестная символ"

Транслятор столкнулся с символом, который он не смог распознать. Возможно, поврежден файл. Если эта ошибка сохраняется, отправьте отчет об этой ошибке на сайт GPSS - MinutemanSoftware.com.

"Использование отрицательного числа"

Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Не удалось инициализировать разъем Windows"

GPSS был неспособен инициализировать интерфейс Internet. Возможно, проблема в Вашей операционной системе.

"Вы не можете использовать GENERATE в ручном моделировании. Используйте SPLIT"

Чтобы создать заявки в интерактивном режиме, Вы должны использовать блок SPLIT, а не GENERATE блок.

"Вы не можете использовать UNSPECIFIED, чтобы инициализировать объект Logicswitch"

Произошла синтаксическая ошибка. Объекты Logicswitch не могут принять значение UNSPECIFIED. В окне модели Вы можете использовать поиск / следующая ошибка или поиск / предыдущая ошибка переместить курсор непосредственно ошибочного оператора. Если ошибка произошла в файле - включения, Вы можете получить диагностические сообщения, временно помещая операторов в объект моделирования.

"Вы не можете ожидать в этом протоколе SNA"

Активная заявка ожидает условия, которые никогда не могут выполняться. Такая заявка не способна ввестись блок. Вы должны изменить значения, операнды, или поток Заявок в моделировании, чтобы предотвратить эту ошибку. Не пытайтесь произвести блокирование на Интегрированной Пользовательской Переменной. Используйте с этой целью команду INTEGRATE.

"Сначала Вам необходимо остановить эксперимент"

Предпринятая операция требует, чтобы текущее моделирование сначала было остановлено.

"Вы должны определить протокол SNA"

Вы должны определить операнд, который является правильным системным числовым атрибутом.

Протоколы SNA рассматриваются в разделе 3.4 *Справочного описания GPSS*.

"Вы должны определить параметры для этого эксперимента"

Вы должны изменить значения, операнды, или поток заявок в моделировании, чтобы предотвратить эту ошибку.

"Вы должны использовать команду Conduct, чтобы вызвать эксперимент"

PLUS эксперименты могут начинаться только при помощи прямой команды Conduct.