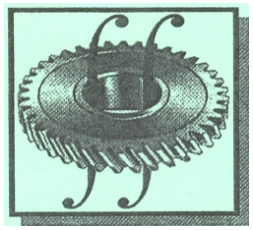

**ФУНДАМЕНТАЛЬНЫЕ ФИЗИКО-МАТЕМАТИЧЕСКИЕ ПРОБЛЕМЫ И МОДЕЛИРОВАНИЕ ТЕХНИКО-ТЕХНОЛОГИЧЕСКИХ СИСТЕМ**

УДК 519.6.

# РАЗРАБОТКА ОТКРЫТОГО ПАКЕТА МОДЕЛИРОВАНИЯ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ

Квитка М.Е., Кулябов Д.С., Семкин Ю.Ю., Томила С.О.

1 РУДН, кафедра телекоммуникаций mekvitka@sci.pdu.edu.ru

2 РУДН, кафедра телекоммуникаций dharma@sci.pdu.edu.ru

3 РУДН, кафедра телекоммуникаций ysyomkin@sci.pdu.edu.ru

4 РУДН, кафедра телекоммуникаций stomila@sci.pdu.edu.ru

*Язык GPSS был разработан для моделирования систем массового обслуживания (СМО) и не имеет аналогов в этой области. К сожалению отсутствует свободно-распространяемый транслятор этого языка. В статье изучается задача создания открытой реализации языка GPSS.*

## 1. Введение

Вычислительные эксперименты, необходимые для расчета работы сложных и очень больших систем, зачастую дороги и невыполнимы для реализации в реальной жизни. Компьютерное моделирование – один из способов изучения сложных систем, позволяющий сымитировать реальную ситуацию, выявить основные факторы и свойства изучаемых объектов.

Один из видов компьютерного моделирования – имитационное. Этот способ подразумевает под собой использование программного комплекса для компьютера, алгоритм работы которого представляет собой функционирование математической модели исследуемого объекта.

Появление ИМ и превращение его в эффективное средство анализа сложных и больших систем было, с одной стороны, обусловлено потребностями практики, а с другой – развитием метода статистических испытаний (метод Монте-Карло), открывшего возможность моделировать случайные факторы, которые существенно влияют на процесс функционирования систем. Кроме того, была создана материальная (аппаратурная) база для реализации ИМл — мощные вычислительные средства второго и третьего поколения.

Введение понятия ИМ в науке (начало 60-х годов) было сопряжено с возникновением определенной терминологической путаницы, разнообразием трактовок этого понятия. Одна из причин, повлиявших на это, связана с тем, что сам термин, обозначаемый в англоязычной литературе как simulation и введенный в отечественной литературе как «имитационное моделирование» , неудачен с чисто лингвистической точки зрения, поскольку в первом случае его можно перевести просто как «моделирование», а во втором – истолковать в тавтологическом смысле, рассматривая термины «имитация» и «моделирование» как синонимы («моделирующее моделирование»). В действительности, когда речь идет об ИМ, то имеется в виду моделирование особого рода, противостоящее в известном смысле аналитическому моделированию. Последнее связано с двумя основными обстоятельствами: во-первых, имитационная модель должна с необходимой полнотой воспроизводить как структуру объекта-оригинала, так и его функционирование (при обязательном сохранении схожести поведения по отношению к объекту-оригиналу), а во-вторых, ИМ ориентируется на получение знаний о прототипе не путем аналитического исследования или однократных численных расчетов, а путем целенаправленных экспериментов на ИМ

Инструменты для имитационного моделирования получили бурное развитие в 70-е года 20 века, когда в разработке находилось множество языков моделирования. На данный момент, можно утверждать, что в мире доминирует лишь несколько таких языков. Одним из таких долгожителей является GPSS (General Purpose Simulating System) – язык для моделирования дискретных систем. Началом истории развития GPSS можно считать 1961 год. В 1970-е он входил в десятку самых популярных языков имитационного моделирования, и в декабре 2006 года в рамках конференции WSC-2006 было торжественно отпраздновано сорокапятилетие GPSS.

Автором и разработчиком первых версий языка был Джеффри Гордон, выдающийся американский специалист в моделирования. С тех пор созданы сотни языков имитационного моделирования (GASP, SIMSCRIPT, SIMULA, и т. д.). От многих осталось лишь название, некоторые успешно развиваются. Одни ориентированы на

моделирование процессов, другие — на моделирование событий. Но среди всех этих языков и систем GPSS стоит особняком. У GPSS много сторонников и много критиков. Сторонники превозносят универсальность, наглядность и простоту использования языка. Критики указывают на некоторую тяжеловесность конструкций, отсутствие современных средств визуализации и консерватизм разработчиков. Но и те и другие признают уникальность GPSS и его вклад в развитие всего рынка имитационного моделирования [1, 2, 3, 4, 5].

## 2. Обзор существующих систем ИМ на основе языка GPSS.

Постепенно, по мере развития языков моделирования, выделились три основных разработчика, «диктующих моду» в мире GPSS.

Джеймс Хендриксен из Wolverine Software и его GPSS/H - это мощная и надежная система имитационного моделирования, основная разработка и активное использование которой велось еще до широкого введения персональных компьютеров. Первая коммерческая версия вышла в свет еще в 1977 году. Несмотря на то, что в настоящее время существует большое множество альтернативных программных продуктов, GPSS/H считается довольно мощной, надежной и гибкой системой имитационного моделирования. Хотя стоит отметить, что посравнению с новейшими разработками, GPSS/H не хватает современных интерактивных технологий. Новейшая версия GPSS/H насчитывает более 70 типов блоков, этот программный продукт как и более ранние версии позиционируется, как система для профессионального моделирования процессов. Распространяется на правах платной лицензии для операционных систем Windows, но существуют shareware-версии, ограниченные в функциональности.

Ингольф Столл из Стокгольмской Школы Экономики разработал компактную версию языка GPSS специально для использования ее в учебном процессе. Эта версия языка называется Micro GPSS. Насчитывает она около 22 типов блоков и позволяет пользователю создавать свои собственные блоки, также хотелось бы отметить небольшую встроенную обучающую систему. Лицензия распространения данного продукта бесплатная. По мере развития этой идеи появилась некая модификация Misro GPSS - WebGPSS, система имитационного моделирования для использования в сети Интернет. Идея состоит в том, что основное ядро системы находится на университетском сервере, а пользователь, отправляя запросы через Интернет, может настраивать и запускать небольшие учебные модели.

Одна из самых удачных и, наверное, самая распространенная из современных разработок - GPSS World - программный продукт от компании Minuteman Software. Основа этой системы - GPSS/PC - более ранняя версия языка моделирования этой же компании. Главный плюс данной системы - наличие удобного графического интерфейса и универсальной многооконной диалоговой оболочки, а также мощная вычислительная способность, рассчитанная на профессионального пользователя. Лицензия распространения данного программного продукта - платная [2, 6, 7, 8].

## 3. Почему GPSS

Математические модели систем массового обслуживания используются во многих предметных областях. В том числе и в среде телекоммуникаций. Кафедра систем телекоммуникаций Российского Университета Дружбы Народов проводит множество разработок, требующих подтверждения опытным путем. Один из способов проверить верность предположений – имитационное моделирование. Язык имитационного моделирования GPSS создан специально для моделирования систем массового обслуживания, поэтому системы на основе GPSS лучшим образом подходят для нужд кафедры, так как основой имитационных алгоритмов в GPSS является дискретно-событийный подход. В GPSS разработчикам удалось очень четко и изящно пройти по грани как соответствия проблемной области (по терминологии, по функциям, методике исследований и т.д.), так и эффективности программирования (удобства разработки моделей, быстродействия, использованию ресурсов ЭВМ и т.д.).

К сожалению, существующие версии систем ИМ на основе языка GPSS либо слишком дороги для покупки кафедрой, либо ограничены в возможностях и не позволяют провести все необходимые исследования. Исходя из этого, было решено создать свою версию модульного компилятора GPSS. Язык, на котором планируется написать компилятор – Haskell.

Haskell является чисто функциональным языком программирования общего назначения, который включает много последних инноваций в разработке языков программирования. Haskell обеспечивает функции высокого порядка, нестрогую семантику, статическую полиморфную типизацию, определяемые пользователем алгебраические типы данных, сопоставление с образцом, описание списков, модульную систему, монадическую систему ввода - вывода и богатый набор примитивных типов данных, включая списки, массивы, целые числа произвольной и фиксированной точности и числа с плавающей точкой. Haskell является и кульминацией, и кристаллизацией многих лет исследования нестрогих функциональных языков [9, 11].

## 4. Пример модели GPSS.

Далее разобран пример моделирования простой СМО в теории и приведены желаемые результаты моделирования.

Постановка задачи.

Рассмотрим модель однолинейной СМО с накопителем бесконечной емкости. На систему поступает рекуррентный поток заявок, промежутки между приходами которых распределены равномерно на интервале 18+6.

Длительности обслуживания заявок являются независимыми в совокупности случайными величинами, равномерно распределенными на интервале 16±6. Моделирование необходимо закончить, когда через систему пройдет 1000 заявок.

Логика моделирования.

Модель системы состоит из семи блоков. Каждый блок снабжен кратким комментарием для лучшего понимания модели.

Рассмотрим подробнее программу моделирования. Блок GENERATE моделирует поступление заявок в систему, а сами заявки моделируются транзактами. В блоке явно заданы первые два операнда, говорящие о том, что промежутки времени между приходами заявок распределены равномерно на интервале 18±6. Поступление заявки в накопитель моделируется поступлением транзакта в блок QUEUE. Единственный операнд этого блока QUE задает имя очереди в символьном виде. Если накопитель не пуст и/или прибор занят, то заявка остается в накопителе, ожидая своей очереди на обслуживание (запрещается вход транзакта в блок SEIZE и он остается в блоке QUEUE). При освобождении прибора заявка занимает его, покидая при этом накопитель. Этот процесс моделируется последовательностью блоков SEIZE - DEPART. Операнд А блока SEIZE задает символическое имя прибора, а блока DEPART - символическое имя той же очереди, что операнд Л блока QUEUE. Длительность обслуживания на приборе моделируется блоком ADVANCE . Явное задание операндов А и В в этом блоке указывает на то, что время обслуживания имеет равномерное распределение на интервале 16±6. Процесс освобождения прибора моделируется прохождением транзакта через блок RELEASE, операнд А которого указывает символическое имя освобождаемого прибора. Напомним, что прохождение транзакта через блок RELEASE позволяет следующему по порядку транзакту (если такой имеется) войти в блок SEIZE, т.е. занять прибор. Таким образом по умолчанию подразумевается дисциплина FIFO выбора транзактов из очереди.

Освободившая прибор заявка покидает систему, что моделируется прохождением транзакта через блок TERMINATE. При этом из счетчика числа завершении вычитается единица (значение операнда А блока TERMINATE). Начальное значение этого счетчика задается в интерактивном режиме командой START и по условию задачи полагается равным 1000. После выхода из системы 1000 заявок счетчик числа завершений обнулится и моделирование закончится.

Результаты моделирования.

По окончании прогона модели интерпретатор распечатывает стандартную статистику. В начале указывается значение TMB (18063.296), при котором завершился процесс моделирования. Вслед за строкой времени дается информация о блоках: номер блока (LOC), название блока (BLOCK TYPE), счетчик числа входов в блок (ENTRY COUNT), указывающего на количество транзактов, вошедших в данный блок за все время моделирования, счетчик текущего содержимого блоков (CURRENT COUNT), указывающий на количество транзактов, находящихся в момент окончания моделирования в данном блоке. Эта информация может быть полезна при проверке логики моделирования.

Далее следует раздел FACILITIES, в котором собраны статистические данные по всем приборам, встречающимся в модели. Первый столбец раздела указывает на имена приборов. В рассматриваемом примере использовался один прибор SERV. В втором столбце дается общее число транзактов, занимавших прибор - 1001, что соответствует числу входов в блок SEIZE (блок №3). В третьем столбце приводится значение загрузки прибора (UTIL.), равное 0.882.

В следующем столбце дается значение среднего времени пребывания транзактов на приборе (AVE.TIME), которое в нашем примере равно 15.920. Столбец с названием AVAIL, содержит информацию о занятости прибора в момент остановки моделирования (0 - прибор свободен, 1 - прибор занят), а в столбце (OWNER) выдается номер транзакта, обслуживанием которого занят прибор в этот момент (если таковой имеется). В нашем примере это транзакт с номером 1001. Информация, содержащаяся в остальных столбцах для нас несущественна.

В разделе QUEUE приводится статистика по всем регистраторам очередей, внесенным в модель разработчиком. В рассматриваемом примере статистика собиралась по одной очереди с именем QUE (первый столбец). Во втором столбце выдается максимальное число транзактов в очереди в течение всего времени моделирования. В примере это значение равно 3. Далее, в столбце CONT. дается текущее содержимое очереди. В нашем примере оно равно 1, т.к. один транзакт в момент остановки моделирования находится в блоке SEIZE и освободит очередь только после прохождения блока DEPART. В столбце ENTRY приводится значение числа транзактов, вошедших в очередь. Оно совпадает с счетчиком числа входов в блок QUEUE, равным 1001. В столбце ENTRY(O) регистрируется число транзактов с пулевым временем пребывания в очереди (482), а следующий столбец AVE.CONT выдает среднее число транзактов в очереди (0.172) в течении всего времени моделирования. Следующие два столбца AVE.TIME и AVE.(-O) дают среднее время пребывания транзакта в очереди соответственно с учетом транзактов с нулевым временем пребывания (3.095) и без их учета (5.969).

Заметим, что интерпретация полученных статистических данных в терминах моделируемой СМО лежит на разработчике, но обычно не вызывает каких-либо трудностей.

Пример моделирования 1. Листинг модели.

Модель однолинейной СМО с накопителем бесконечной емкости и с интервалами между поступлениями и длительностями обслуживания заявок распределенными по равномерному закону

```
GENERATE 18,6 ;поступление заявок
QUEUE que ; присоединение к очереди
SEIZE serv ;занятие прибора
DEPART que ;уход из очереди
ADVANCE 16,6 ;обслуживание на приборе
RELEASE serv ;освобождение прибора
TERMINATE1 ;уход из системы
```

Пример моделирования 1. Листинг результатов.

| START TIME | END TIME | BLOCKS | FACILITIES | STORAGES |
|---|---|---|---|---|
| 0.000 | 18063.296 | 7 | 1 | 0 |

| NAME | VALUE |
|---|---|
| QUE | 100000,000 |
| SERV | 100001,000 |

| LOC | BLOCK TYPE | ENTRY COUNT | CURRENT COUNT | RETRY |
|---|---|---|---|---|
| 1 | GENERATE | 1001 | 0 | 0 |
| 2 | QUEUE | 1001 | 0 | 0 |
| 3 | SEIZE | 1001 | 1 | 0 |
| 4 | DEPART | 1000 | 0 | 0 |
| 5 | ADVANCE | 1000 | 0 | 0 |
| 6 | RELEASE | 1000 | 0 | 0 |
| 7 | TERMINATE | 1000 | 0 | 0 |

| FACILITY | ENTRIES | UTIL.AVE | TIME | AVAIL |
|---|---|---|---|---|
| SERV | 1001 | 0.882 | 15.920 | 1 |

| FACILITY | OWNER | PEND | INTER | RETRY | DELAY |
|---|---|---|---|---|---|
| SERV | 1001 | 0 | 0 | 0 | 0 |

| QUEUE | MAX | CONT | ENTRY | ENTRY(0) | AVE. CONT |
|---|---|---|---|---|---|
| QUE | 3 | 1 | 1001 | 482 | 0.172 |

| QUEUE | AVE.TIME | AVE.(-0) | RETRY |
|---|---|---|---|
| QUE | 3.095 | 5.969 | 0 |

[10].

### 5. Заключение.

Задачу написания модульного компилятора можно разбить на 4 основных этапа.

Первый этап – выбор наречия языка GPSS и описание его в форме Бэкуса-Наура (БНФ), которая используется для описания синтаксиса языков программирования, данных, протоколов и т.д. За основу выбрано наречие GPSS World . По-сравнению с другими версиями, эта дополнена новыми объектами языка GPSS. В первоначальной версии компилятора будут реализованы только основные и наиболеее часто используемые блоки и команды языка. В дальнейшим планируется расширение возможностей и написание новых, своих блоков.

Второй этап – написание лексического анализатора текста. Основная задача лексического анализа - разбить входной текст, состоящий из последовательности одиночных символов, на последовательность слов, или лексем, то есть выделить эти слова из непрерывной последовательности символов. Все символы входной последовательности с этой точки зрения разделяются на символы, принадлежащие каким-либо лексемам, и символы, разделяющие лексемы (разделители).

Третий этап – написание синтаксического анализатора. Процесс определения принадлежности данной строки языку, порождаемому данной грамматикой, и, в случае указанной принадлежности, построение дерева разбора для этой строки, называется синтаксическим анализом. Основная сложность данного этапа – большой объем работ, т.к. каждому наречию языка GPSS соответствует своя БНФ и соответственно пишется отдельный модуль для анализа программы.

Четвертый этап – построение хэш-таблицы и генерация исполняемого кода. Хэш-таблица – эффективной структуры для хранения и получения динамических данных. Типичное применение хэш-таблиц – символьная таблица, которая ассоциирует некоторое значение (данные) с каждым членом динамического набора строк (ключей). Обычно компилятор использует хэш-таблицу для управления информацией о переменных в программе. Возможны отдельные модули для интерпретации исходного кода и для его трансляции.

### Список источников.


1. Девятков В. В., Якимов И. М. Проект http://gpss.ru
2. Девятков В. Предсказание погоды/ Компьютерра от 11.07.2003, http://offline.computerra.ru/offline/2003/496/27500/index.html
3. Бахвалов Л. Компьютерное моделирование: долгий путь к сияющим вершинам?/ Компьютерра от 06.10.1997, http://offline.computerra.ru/1997/217/814/
4. Губарь Ю.В. Введение в компьютерное моделирование/ http://www.intuit.ru/department/calculate/intromathmodel/
5. Компьютерное моделирование. http://ru.wikipedia.org/wiki/Компьютерное_моделирование
6. GPSS/H. http://www.iteam.ru/soft/modelling/990/

7. GPSS/H — Serving the Simulation Community. http://www.wolverinesoftware.com/GPSSHOverview.htm
8. http://www.minutemansoftware.com/
9. Себастиан Сильван. Сильные стороны языка Haskell/ http://ru.wikibooks.org/wiki/Сильные_стороны_языка_Haskell
10. Матюшенко С. И., Спесивов С. С. Основы имитационного моделированнния в среде GPSS World. Москва: издательство Российского Университета Дружбы Народов, 2006
11. Хьюдак П., Петерсон Д., Фасел Д., перевод Москвин Д. Мягкое введение в Haskell/ RSDN Magazine от 01.2007