Математические структуры и моделирование
2002, вып. 10, с. 172–191

УДК 519.876.5:681.518

ИМИТАЦИОННОЕ ИССЛЕДОВАНИЕ КОНЦЕПЦИЙ СБОРА ИНФОРМАЦИИ ДЛЯ ИНДЕКСОВ ПОИСКОВЫХ СИСТЕМ

И.А. Земсков

In the e article the description of web resources' information state monitoring system is presented. Then two simulation models of this system for two building concepts were built. Computer models were realized on Python with SimPy package. The result of our own experiment is considered.

Введение

Общий круг проблем, стоящих перед разработчиками подсистем сбора информации для поисковых систем, был обозначен в статье [1]. В этой же статье кратко были рассмотрены возможные подходы к решению поставленных проблем. Эти подходы формируют три основные концепции (роботов, сенсоров, мобильных роботов) реализации подсистем создания представления об информационном содержимом сети Интернет. В настоящее время наибольшее развитие получила концепция роботов. Фактически она является единственной концепцией, применяемой поисковыми системами сети Интернет. Так как поиск публикаций, проливающих свет на причины такого предпочтения, не дал никакого результата (утверждение о том, что концепцию роботов легко внедрить, за аргумент не принималось), то остаётся загадкой фактически нулевой интерес со стороны разработчиков и исследователей к альтернативным концепциям. Однако отсутствие интереса к альтернативным концепциям не означает отсутствие интереса к развитию различных стратегий в рамках концепции роботов. А вот здесь становится заметной практика предварительного имитационного моделирования для исследования вновь предлагаемых стратегий разработки роботов с целью нахождения наиболее оптимальных вариантов построения алгоритмов их функционирования [2]. Применение имитационного моделирования в исследованиях тесно связанно с термином «система». Пристальный взгляд на обозначенную проблемную область позволяет заметить, что мы действительно имеем дело с системой. Согласно Р.Шеннону [3], система определяется как группа, или совокупность объектов, объединённых некоторой формой регулярного взаимодействия или взаимозависимости для выполнения заданной функции. В нашем случае совокупностью объектов системы S можно назвать совокупность информационных ресурсов и узла поисковой системы, занимающегося мониторингом

E-mail: zemskov@univer.omsk.su
Омский государственный университет

информационного состояния доступных ресурсов. На входе система S имеет потоки запросов и потоки изменений, направленные к каждому информационному источнику. На выходе системы имеем представление об информационном содержании сети Интернет. Целевая функция данной системы – приобретение. Конкретнее, «дешёвое» приобретение полного и адекватного представления о содержимом информационных ресурсов для индексов поисковой системы. Таким образом, можно сказать, что мы имеем дело с системой (S) мониторинга информационного состояния сети Интернет (S является подсистемой более крупной – поисковой системы).

1. Постановка задачи

Очевидно, что стоит актуальная задача для исследования, которая будет заключаться в сравнении существующих концепций создания представления об информационном содержимом сети Интернет. Другими словами, нам нужно найти оптимальный вариант построения системы S. Однако такая постановка задачи требует предварительно найти ответы на следующие вопросы:

- Существуют ли такие числовые критерии (критерии «эффективности»), которые были бы максимально общими и могли бы послужить нам объективным инструментом сравнения любых концепций (а также разных стратегий в рамках одной концепции) реализации исследуемой системы?
- Если предположить, что нам удастся найти и описать критерии сравнения, то существуют ли доступные средства, позволяющие получить цифровые (вероятностно-временные характеристики) значения критериев эффективности для концепций и стратегий, которые ещё не реализованы на практике (но имеют описание алгоритма функционирования)?

Ответ на первый вопрос, очевидно, можно найти, анализируя сформулированную ранее целевую функцию исследуемой системы. Проведя небольшой анализ и сравнив его результаты с результатами других исследователей [4, 5], мы пришли к выводу, что наиболее общими и объективными критериями эффективности для всех концепций и стратегий будут следующие. Во-первых, «свежесть» (freshness) информации текущего представления. Во-вторых, объём (size) «циркулирующей» в системе информации, необходимой для поддержания представления в актуальном («свежем») состоянии. Таким образом, поиск оптимального варианта заключается в поиске варианта, который будет иметь максимальное значение параметра свежести и минимальное значение параметра объёма скачиваемой информации. Причём для объективного сравнения значений этих параметров нужно фиксировать длину отрезка времени, на котором они получены.

А вот положительный ответ на второй вопрос видится в использовании средств имитационного моделирования. Причём, если не гнаться за абсолютной точностью оценок критериев эффективности, то можно значительно упростить разрабатываемые модели, не перегружая их лишними подробностями и в то же время получить адекватный инструмент для изучения «поведения» интересующей нас системы в случае каждой конкретной концепции или стратегии её

построения.

Таким образом, сформулируем нашу задачу как задачу создания средств имитационного моделирования следующим образом:

Объект моделирования. Система мониторинга информационного состояния сети Интернет.

Цели моделирования. Научное исследование производительности модуля, который отвечает за мониторинг информационного содержания сети Интернет, при различных вариантах построения системы. Анализ влияния различных параметров информационных ресурсов сети Интернет на показатели производительности исследуемого модуля и в конечном итоге всей исследуемой системы.

Требования к средствам моделирования

1. Возможность использовать преимущества объектно-ориентированного подхода.
2. Преемственность. Т.е. возможность лёгкой модификации ранее разработанных моделей для создания моделей новых стратегий и концепций.
3. Возможность справиться с большим объёмом вычислений при наличии техники с малым количеством системных ресурсов.
4. Возможность сбора и удобного представления необходимых статистических данных.

2. Описание системы

Ограничим рассмотрение доступных концепций только двумя «базовыми» вариантами, а именно концепциями роботов и сенсоров. Описание системы начнём с варианта её реализации в рамках концепции роботов.

2.1. Система «робота»

В рассматриваемой системе можно выделить объекты двух видов. Объектами первого вида являются информационные ресурсы. В общем случае будем придерживаться определения информационного ресурса, которое дано в [1]. А именно это некий файл, имеющий внутреннюю структуру согласно спецификации одного из существующих типов данных (HTML, JPEG, GIF, PNG, SWF, WAV, MIDI, MP3, RA и т.п.) и находящийся на специальном сервере (информационный источник) в Интернет. Сервер должен обеспечивать доступ к этому файлу по протоколу HTTP. Использование протокола HTTP для доступа к файлу подразумевает, что файл имеет свой однозначный URL-адрес. Т.е. при использовании для доступа к ресурсу этого URL сервером будет «отдан» именно тот файл, который нам нужен. Под словом «отдан» понимается передача данных файла по сети Интернет с одного узла на другой.

Помимо URL информационный ресурс описывается:

- размером (в байтах)
- текущим состоянием (доступен, недоступен с кодом ошибки)
- потоком внешних запросов на его скачивание

- потоком заявок на изменение его содержимого или состояния

В общем случае нам ничего не известно о характере потока запросов и потока изменений ресурса. Однако можно сделать предположение, что они имеют экспоненциальные законы распределения времени наступления соответствующих событий.

Независимо от интенсивности потока запросов на скачивание ресурса будем считать, что сервер, на котором расположен ресурс, имеет достаточно большую пропускную способность канала связи с сетью Интернет, и оборудование сервера может справиться практически с любым «наплывом» запросов различных информационных ресурсов данного сервера. Другими словами, будем считать, что любой запрос на скачивание ресурса будет удовлетворён в тех временных рамках, которые устанавливает технология (т.е. заведомо до истечения времени выдачи сообщения «Time out»).

Таким образом, информационный ресурс «живёт» своей жизнью. Например, за некоторый промежуток времени он может успеть: родиться и вырасти, затем исчезнуть, снова появиться, «усохнуть» и, наконец, «умереть» окончательно. При этом в разные моменты времени его «хотят увидеть».

За всеми перипетиями этой «жизни» и призван следить модуль мониторинга информационного состояния ресурсов. Собственно, он и является объектом второго вида. В рамках рассматриваемой концепции функции данного модуля реализуются с помощью специальной программы-робота. Робот запускается на выполнение на специальном узле поисковой системы. Начальной информацией для работы робота является список URL адресов ресурсов. А вот то, что он делает с этим списком дальше, зависит от той стратегии, которую выбирают разработчики каждого робота. Самой простой стратегией поведения является последовательный «обход» ресурсов, представленных в списке (на начальном этапе нашего исследования мы ограничимся рассмотрением только этой стратегии). Под обходом понимается то, что робот скачивает к себе на узел содержимое ресурса с использованием протокола HTTP. Следующий шаг (который мы должны упомянуть) в судьбе скачанного ресурса это то, что он поступает в модуль анализа. На выходе модуля анализа мы получаем список URL-адресов на новые ресурсы, которые не присутствовали в исходном списке. Полученный список URL-адресов добавляется к исходному списку, и работа робота продолжается уже с изменённым списком. Куда именно добавляется найденный список и как продолжается работа с объединённым списком зависит, сугубо от стратегии, которую выбирают разработчики. В связи с тем, что остальные концепции могут решать проблему обнаружения новых ресурсов принципиально по-другому, мы предлагаем в рамках исследования зафиксировать обследуемый набор ресурсов, т.е. временно отказаться от рассмотрения функции поиска новых ресурсов в рамках системы мониторинга.

В связи с тем, что пропускная способность канала, через который узел с роботом подключены к Интернет, очень сильно влияет на скорость обхода (время скачивания ресурса зависит от «размера» канала), будем считать, что пропускная способность канала робота имеет максимально возможный размер, чтобы минимизировать время скачивания любого, отдельно взятого ресурса и тем са-

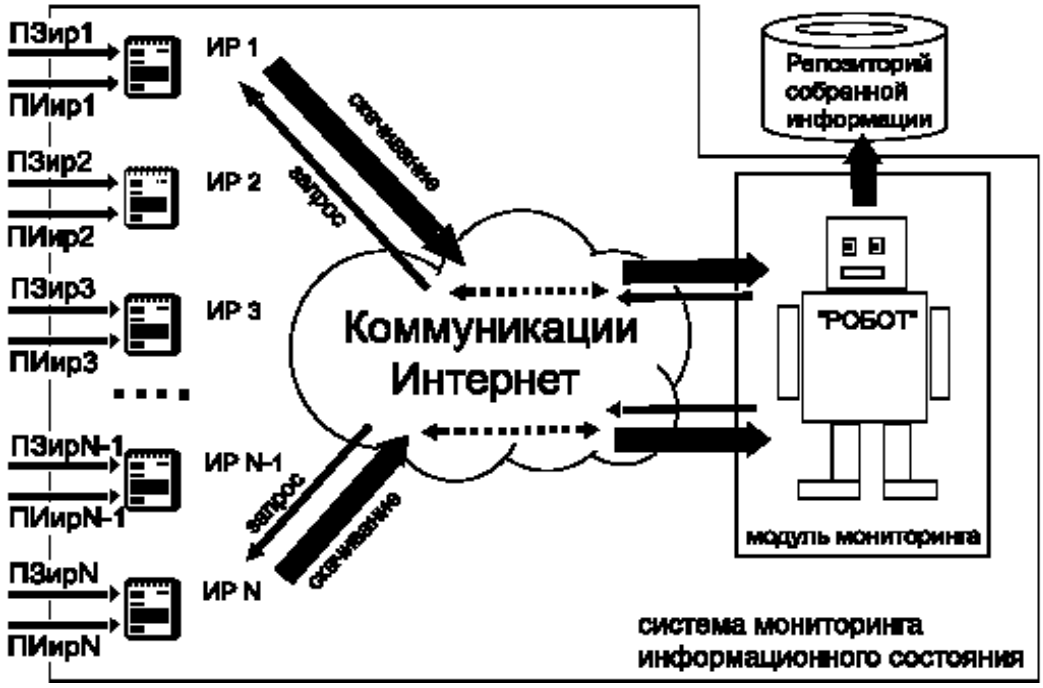

Рис. 1. Система мониторинга информационного состояния (вариант «робота»)

мым исключить возникновение так называемого «узкого места».

Наши договорённости по поводу пропускных способностей каналов серверов с информационными ресурсами и узла робота вызваны стремлением избежать ненужного усложнения модели впоследствии. Действительно, если задавать пропускную способность канала каждого ресурса, то при вычислении времени, которое требуется на скачивание этого ресурса в ответ на поступивший запрос, нужно будет учитывать «занятость» этого канала обслуживанием запросов других ресурсов, которые также расположены на этом сервере. Другими словами, нам придётся учитывать множество второстепенных факторов, которые имеют отдалённое отношение к концепции построения системы мониторинга. Тогда как время, которое потребуется на скачивание ресурса в ответ на поступивший запрос, можно определить как случайную величину, которая, например, равномерно распределена между минимально и максимально возможными значениями времени скачивания. Так как скачивание ресурса роботом и скачивание ресурса в рамках ответа на внешний запрос ничем технологически не отличаются, то правило расчёта времени, затрачиваемого на скачивание, для них будет одинаковым.

Суммируя всё выше сказанное, можно представить систему в таком виде, как это изображено на рис.1.

На рисунке: ИР1-ИРN обозначают информационные ресурсы с 1 по N (технологически любая их комбинация может располагаться на разных серверах информационных источников, но мы не учитываем их реальное расположение, т.к. это усложняет рассмотрение). ПЗир1-ПЗирN означают потоки запро-

сов, направленные соответственно к 1,..., N-тому информационному ресурсу. ПИир1-ПИирN означают потоки изменений, направленные соответственно к 1,..., N-тому информационному ресурсу. Основным рабочим элементом модуля мониторинга является программа-робот. Робот по очереди посылает запросы на скачивание информационных ресурсов (ИР). На время скачивания каждого ИР субъективно влияет структура коммуникаций Интернет. На выходе модуля мониторинга и всей системы в целом мы получаем репозиторий собранной информации. Он является «сырым» (т.е. исходным материалом для модуля построения индексов) представлением информационного содержания Интернет.

Осталось заметить, что в данном варианте построения системы модуль мониторинга никоим образом не использует и не учитывает потоки запросов к информационным ресурсам, т.к. в концепции нет технологической основы для их использования. Однако в нашем рассмотрении про них нужно помнить, т.к. они могут ещё пригодиться в других концепциях построения системы.

2.2. Система «сенсоров»

Так как концепция сенсоров ещё не имеет устоявшихся и апробированных вариантов реализации, а имеет только описание примерного алгоритма реализации [1, 6], то будем опираться при рассмотрении системы на это описание. Собственно, нам должно быть этого достаточно, чтобы проверить возможность рассмотрения новых концепций и стратегий без их практической реализации.

В рассматриваемой системе можно выделить объекты трёх видов. Объектами первого вида являются информационные ресурсы (ИР). В общем случае, для данной концепции, определение ИР совпадает с определением ИР, которое было дано для концепции роботов. То есть ИР имеет: размер (в байтах), текущее состояние (доступен, недоступен с кодом ошибки), поток внешних запросов на его скачивание, поток заявок на изменение его содержимого или состояния.

Однако если в случае концепции роботов мы ничего не говорили о программном обеспечении (ПО) специальных серверов (т.н. информационных источников), т.к. концепция никоим образом его не затрагивает, то в случае концепции сенсоров мы должны будем сказать несколько слов на эту тему. Основу концепции сенсоров как раз и составляет предложение внести дополнение в состав ПО серверов. Точнее предлагается расширить ПО сервера программным модулем («сенсором», отсюда и пошло название концепции), который будет заниматься обнаружением произошедших изменений в состоянии информационных ресурсов и сообщать о найденных изменениях на головной узел модуля мониторинга. Обобщённый алгоритм функционирования ПО сервера и модуля-сенсора выглядит следующим образом:

1. На сервер информационного источника поступает запрос на скачивание определённого (посредством URL-адреса) информационного ресурса.
2. ПО сервера информационного источника «готовит» ответ (ИР) на запрос, и прежде, чем этот ответ отправляется запрашивающему, он поступает на обработку модулю-сенсору.
3. Если запрашиваемый ИР ранее сенсору был неизвестен, то формирует-

ся признак наступления изменения (чтобы послать «сигнал тревоги»), и алгоритм переходит на шаг 5. Иначе шаг 4.

4. Используя несколько критериев-признаков наступления изменения, сенсор пытается определить факт изменения ИР.
5. ПО сервера информационного источника «отдаёт» ответ на запрос. При обнаружении факта изменения ИР сенсор «посылает» на головной узел модуля мониторинга «сигнал тревоги» и «запоминает» характеристики «нового» состояния ИР.
6. Запрос обслужен и, если нужно, послан «сигнал тревоги». Сервер информационного источника переходит в режим ожидания следующего запроса.

Остаётся добавить, что модуль-сенсор и есть объект второго вида в рассматриваемой системе.

Последним, объектом третьего вида, будет являться головной узел модуля мониторинга. Как уже можно понять, у программы-робота (она является основой модуля мониторинга) остаётся только одна большая функция: скачивание ИР на головной узел модуля мониторинга. Причём следующего кандидата на скачивание определяют для него «сигналы тревоги». Другими словами, основными функциями робота будут: ожидание «сигнала тревоги» от сенсоров и скачивание новой версии изменившегося ИР. Таким образом, в данной концепции программа-робот теряет активную составляющую.

Однако даже при таком кратком описании системы в рамках концепции сенсоров можно выдвинуть гипотезу о большом влиянии интенсивности потока запросов ИР и потока изменений ИР на критерии эффективности функционирования всей системы в целом.

Сделаем несколько замечаний для полноты картины изложения:

1. Оставим для этой концепции в силе договорённость по поводу пропускных способностей каналов серверов с информационными ресурсами и головного узла модуля мониторинга, т.к. это поможет избежать неоправданного усложнения модели системы и сравняет по условиям рассмотрения две концепции.
2. Решение о включении в рассмотрение системы мониторинга потоков запросов к информационным ресурсам оказалось верным (т.е. эти потоки нам пригодились).
3. Из обобщённого алгоритма функционирования ПО сервера и модуля- сенсора уже сейчас проглядывается способ реализации (в рамках данной концепции) функции обнаружения новых информационных ресурсов. А именно основную часть работы подразумевается возложить на сенсоры. Для увеличения результативности работы этой функции видится возможным расширить головной узел модуля мониторинга блоком анализа. Блок анализа должен будет дополнительно анализировать поступающие на узел копии ИР на предмет наличия адресов URL, не внесённых ранее в список известных ИР. Дальше он должен передавать найденные адреса в блок скачивания программы-робота для их последующей обработки. В силу предыдущей договорённости при создании модели описываемой системы функцию поиска новых ИР учитывать не будем.

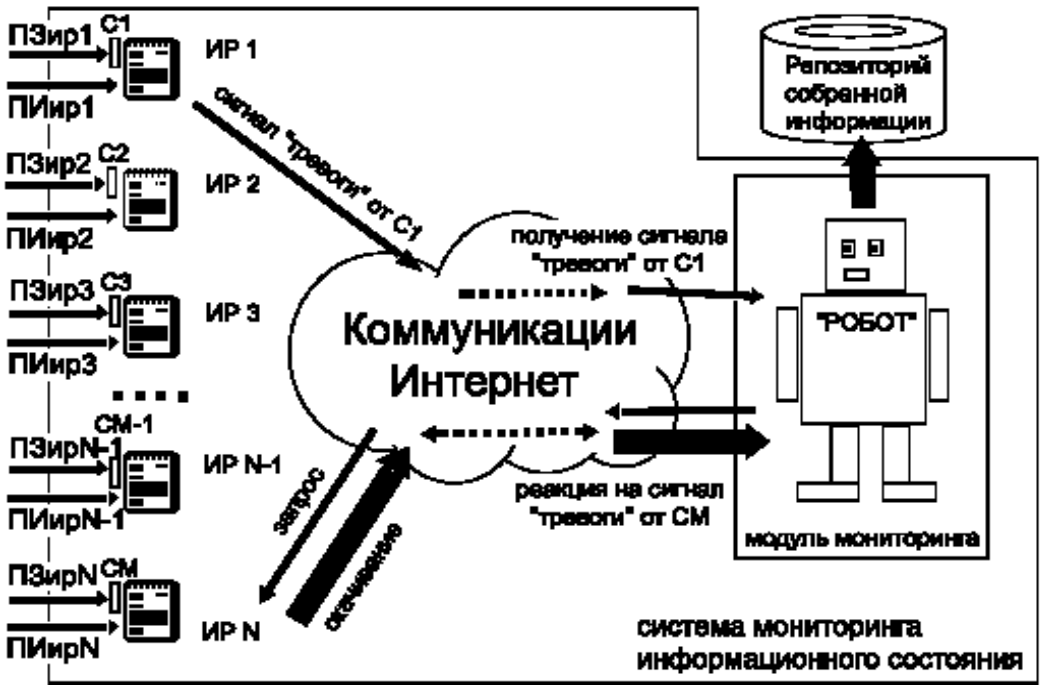

Рис. 2. Система мониторинга информационного состояния (вариант «сенсоров»)

Суммируя всё выше сказанное, можно представить систему в таком виде, как это изображено на рисунке (1).

Рис.2 во многом напоминает рис.1, однако есть и принципиальные отличия. Во-первых, это наличие «сенсоров» С1-СМ на пути соответствующих потоков запросов ПЗир1-ПЗирN. Как было сказано ранее, каждому серверу информационных источников соответствует только один сенсор, а количество серверов может быть (практически всегда) меньше количества информационных ресурсов. Собственно поэтому сенсоры имеют свою нумерацию с 1 по M (M, в общем случае, не равно N). Во-вторых, с появлением сенсоров заметно преобразились алгоритмы циркуляции информации внутри системы. Теперь робот посылает запрос на скачивание определённого ИР и получает его содержимое в ответ на свой запрос только в случае предварительного получения им сигнала «тревоги» от сенсора, который «наблюдает» за изменениями в состоянии этого самого ИР. На выходе модуля мониторинга и всей системы в целом мы, так же как и в случае системы «робота», получаем репозиторий собранной информации, который является «сырым» (т.е. это исходный материал для модуля построения индексов) представлением информационного содержания Интернет.

3. Критерии эффективности

Основные критерии эффективности уже были указаны при постановке задачи. Однако нужно дать некоторые пояснения, связанные с ними. Начнём с критерия «свежесть».

В связи с тем, что у исследователей нет единства в определении термина «свежесть», необходимо зафиксировать определение, которым мы будем пользоваться в нашем исследовании. А именно под свежестью созданного модулем мониторинга представления будем понимать процентное выражение количества страниц, оставшихся неизменными на текущий момент времени по отношению к общему количеству страниц. Таким образом, для определения свежести созданного представления нужны следующие величины:

1. Текущий момент времени (вычисление значения свежести тесно связано с использованием «принципа Δt (дельта t)», т.е. это когда состояние моделируемой системы S проверяется каждые Δt временных единиц [7]).
2. Общее количество страниц (вспомним о договорённости зафиксировать количество рассматриваемых ИР) – обозначим P.
3. Количество страниц, для которых на текущий момент времени не требуется обновление информации об их содержимом, накопленной модулем мониторинга, обозначим Fp.
4. Количество изменившихся страниц на текущий момент времени – обозначим Cp.

Очевидно, что верно равенство $Fp = P - Cp$. Составим такое отношение: $P - 100\%$, $Fp - x\%$. Таким образом, получаем формулу для нахождения свежести:

$$Freshness = \frac{Fp * 100}{P}$$

или более подробно:

$$Freshness = \frac{(P - Cp) * 100}{P}.$$

Тем самым мы получим значение свежести, измеряемой в процентах.

Переходим к рассмотрению критерия «объём циркулирующей в системе информации». Под этим критерием мы будем понимать суммарный объём (в байтах) всех ИР, которые модуль мониторинга перекачал в репозиторий собранной информации за фиксированный промежуток времени наблюдения. «Физический» смысл этого критерия заключается в том, что перекачка информации создаёт нагрузку на коммуникации Интернет (можно предположить, что разные варианты построения системы будут иметь сильное расхождение в уровнях нагрузки). А если учесть ещё и то, что объём скачанной информации, имеет однозначный денежный эквивалент, т.е. свою денежную стоимость (причём за т.н. «трафик» платят обе стороны, как владелец ИР, так и владелец узла мониторинга), то становится понятным желание снизить числовое значение этого критерия.

Прежде чем заняться оптимизацией системы по основным критериям эффективности, следует определить вспомогательные критерии оценки результатов функционирования системы, то есть описать критерии, которые помогут нам оценить адекватность моделей и которые выступят вспомогательными инструментами анализа и сравнения вариантов построения системы. А в конечном итоге они помогут выбрать (при возникновении спорных ситуаций) лучший вариант построения системы.

Во-первых, продолжая разговор про объём скачиваемой модулем мониторинга информации, можно предположить, что будет полезным рассмотрение критерия «процент полезных скачиваний». Поясним на примере концепции роботов. Алгоритм функционирования модуля мониторинга (в общем случае) при данной концепции таков, что модуль скачивает ИР вне зависимости от того, требуется это или нет для поддержания в «актуальном» (свежем) состоянии репозитория собранной информации. То есть мы явно будем иметь процент «бесполезных» скачиваний ИР (эта очевидная гипотеза, однако, требует экспериментального исследования).

Во-вторых, вернувшись к рассмотрению критерия свежести, можно сходу предложить ещё несколько вспомогательных критериев оценки результатов функционирования системы. Например, следующие:

1. Количество изменений ИР, произошедших между последовательными скачиваниями модулем мониторинга содержимого этого самого ИР. То есть данный критерий предлагает учитывать количество изменений в состоянии ИР, которые «остались без внимания» со стороны модуля мониторинга. Как оценку результатов функционирования системы нам, очевидно, будет интересно использовать минимальное, среднее и максимальное значения данного критерия в рамках всей системы.
2. Если за начальную точку отсчёта времени выбирать момент наступления некоторого изменения в состоянии любого ИР, а за конечную точку отсчёта брать момент, когда результаты этого изменения попадут в репозиторий собранной информации, то для полученного промежутка времени нам будут интересны его минимальное, среднее и максимальное значения в рамках всей системы. То есть данные критерии пытаются оценить количество времени, которое потребуется модулю мониторинга, чтобы «узнать» о произошедшем изменении и актуализировать состояние репозитория собранной информации. Однако при подсчётах значений этих критериев нужно учитывать тот факт, что за расчётный период времени может произойти ещё несколько изменений.

4. Концептуальные модели

При описании концептуальных моделей использовались приведённые ранее описания двух вариантов построения системы мониторинга. Модели представляют собой отражение основных элементов системы с учётом специфики их имитации. В связи с тем, что должны получиться две модели одной и той же системы, то они будут очень похожи. Однако, несмотря на это, мы будем давать их полное описание.

4.1. Модель «робота»

В концептуальной модели «основными» объектами являются объекты типа «информационный ресурс» - ИР. Объекты этого типа имеют следующий набор характеристик:

- содержимое (некая последовательность символов);
- размер содержимого (количество байт);
- текущее состояние (набор допустимых значений);
- признак того, что серия последних изменений стала известна в репозитории собранной информации (т.е. признак того, что в репозитории хранится «свежая» информация).

Все остальные объекты модели делятся по принципу их отношения к информационным ресурсам на группы «влияющих» и «наблюдающих». Начнём рассмотрение с первой и самой многочисленной группы, к которой относятся объекты типа «источники изменений» (ИИ).

Каждый объект этой группы имитирует поток «изменений», который направлен к одному-единственному ИР. Другими словами, одному ИР соответствует один ИИ и наоборот (т.е. многочисленность группы «влияющих» и обусловлена таким соответствием). Под изменением в данном контексте понимается один из двух возможных вариантов поведения:

1. Попытка изменить текущее состояние ИР (следует пояснить, что сюда относятся состояния типа «доступен-недоступен»).
2. Попытка изменить содержимое ИР (причём, например, если предыдущее состояние было равносильно «недоступен», то с наступлением изменения содержимого текущее состояние может измениться на «доступен»).

Поток изменений характеризуется законом распределения времени появления следующего изменения (смены состояния ИР или смены содержимого ИР) и законом распределения вероятности появления определённого (одного из возможных вариантов) «изменения». Таким образом, взаимодействие конкретной пары объектов типа ИР и типа ИИ можно описать следующим алгоритмом:

1. Согласно закону распределения времени появления следующего изменения генерируется момент наступления нового изменения текущего ИР - Тизм.
2. Согласно закону распределения вероятности появления одного из возможных вариантов «изменения» и с учётом текущего состояния ИР «выбирается» новое состояние и/или новое содержимое (Инов), в которое «изменится» ИР при наступлении момента изменения (Тизм).
3. В момент времени Тизм вступает в силу Инов и тем самым текущие характеристики ИР изменяются на новые. Если произошло изменение содержимого ИР или его текущего состояния, то признак «свежести» репозитория принимает значение «ложь». Переход к шагу 1.

Теперь рассмотрим объекты группы «наблюдающих». К этой группе относятся объекты типа «робот» (Р) и «репозиторий» (Хр). Основная цель Р — поддержание репозитория (Хр) собранной информации в «свежем» (актуальном) состоянии. Алгоритм его действий по достижению этой цели следующий.

1. Создаётся список всех «наблюдаемых» ИР.
2. В получившемся списке выбирается первый ИР.
3. Используя закон распределения времени скачивания ИР, определяется момент завершения процесса скачивания текущего ИР - Тскач.

4. В момент времени Тскач роботу становятся известны значения характеристик ИР.
5. Характеристики ИР анализируются роботом с целью обнаружения признаков произошедших «неучтённых» изменений. Если таковых обнаружено не было, то осуществляется переход к шагу 8.
6. Если становится известно, что с момента последнего посещения роботом данного ИР произошло его изменение, то робот обновляет хранимую в репозитории информацию о содержимом ИР на новые сведения.
7. У текущего ИР робот изменяет значение признака свежести, хранимого в репозитории информации, на значение «истина».
8. Завершив работу с текущим ИР, робот переходит к обследованию следующего ИР. Следующий обследуемый ИР робот выбирает из списка всех «наблюдаемых» ИР согласно установленной последовательности. Если весь список пройден, то следующим обследуемым выбирается ИР, который стоит первым в списке (т.о. получается, что робот постоянно обследует состояние вверенных ему ИР).
9. Переход к шагу 3.

4.2. Модель «сенсоров»

Аналогично ранее рассмотренной модели в данной концептуальной модели «основными» объектами являются объекты типа «информационный ресурс» - ИР. Объекты этого типа имеют следующий набор характеристик:

- содержимое (некая последовательность символов);
- размер содержимого (количество байт);
- текущее состояние (набор допустимых значений);
- признак того, что серия последних изменений стала известна в репозитории собранной информации (т.е. признак того, что в репозитории хранится «свежая» информация).

Все остальные объекты модели делятся по их отношению к информационным ресурсам на группы «влияющих» и «наблюдающих». Начнём рассмотрение с первой группы, к которой относятся объекты следующих типов:

1. Источники изменений, ИИ.
2. Источники запросов, ИЗ.

Каждый объект первого типа (ИИ) имитирует поток «изменений», который направлен к одному-единственному ИР. Другими словами, существует однозначное соответствие между одним ИР и одним ИИ. Под изменением в данном контексте понимается один из двух возможных вариантов поведения:

1. Попытка изменить текущее состояние ИР (следует пояснить, что сюда относятся состояния типа «доступен-недоступен»).
2. Попытка изменить содержимое ИР (причём, например, если предыдущее состояние ИР было равносильно «недоступен», то с наступлением изменения содержимого текущее состояние может измениться на «доступен»).

Поток изменений характеризуется законом распределения времени появления следующего изменения (смены состояния ИР или смены содержимого ИР)

и законом распределения вероятности появления определённого (одного из возможных вариантов) «изменения». Таким образом, взаимодействие конкретной пары объектов типа ИР и типа ИИ можно описать следующим алгоритмом:

1. Согласно закону распределения времени появления следующего изменения генерируется момент наступления нового изменения текущего ИР - Тизм.
2. Согласно закону распределения вероятности появления одного из возможных вариантов изменения и с учётом текущего состояния ИР выбирается новое состояние и/или новое содержимое (Инов), в которое изменится ИР при наступлении момента изменения Тизм.
3. В момент времени Тизм вступает в силу Инов и тем самым текущие характеристики ИР изменяются на новые. Если произошло изменение содержимого ИР или его текущего состояния, то признак «свежести» репозитория принимает значение «ложь». Переход к шагу 1.

Каждый объект второго типа (ИЗ) имитирует поток «запросов», который направлен к конкретному ИР (одному-единственному). Другими словами, между одним ИР и одним ИЗ существует однозначное соответствие. Таких пар в модели будет ровно столько, сколько в ней присутствует объектов типа ИР. Под запросом в данном контексте понимается попытка инициализировать процесс передачи содержимого ИР за пределы рассматриваемой системы. Поток запросов характеризуется законом распределения времени появления следующего запроса. Таким образом, взаимодействие конкретной пары объектов типа ИР и типа ИЗ можно описать следующим алгоритмом:

1. Согласно закону распределения времени появления следующего запроса генерируется момент наступления нового запроса ИР - Тзапр.
2. В момент времени Тзапр определяется доступность ИР для скачивания. В случае доступности ИР, используя закон распределения времени скачивания информационных ресурсов, определяется момент завершения процесса скачивания текущего ИР - Тзапрскач.
3. Переход к шагу 1.

Теперь рассмотрим объекты группы «наблюдающих». К этой группе относятся объекты типа «сенсор» (Сен), «робот» (Р) и «репозиторий» (Хр). Согласно описанию концепции сенсоров процесс поддержания репозитория (Хр) собранной информации в «свежем» (актуальном) состоянии не осуществляется единолично роботом (как это можно было видеть в модели «робот»), а распределён между объектами типа «сенсор» и «робот». Таким образом, функции по обнаружению произошедших изменений состояния ИР или изменения содержимого ИР возлагаются на объекты типа «сенсор». Для простоты рассмотрения будем считать, что каждому объекту ИР соответствует один объект С (и наоборот). Алгоритм функционирования объектов типа «сенсор» следующий.

1. В момент времени Тзапр вместе с началом обработки поступившего внешнего запроса на скачивание ИР в работу включается сенсор и пытается обнаружить признаки произошедших «неучтённых» изменений ИР. Если таковых обнаружено не было, то сенсор завершает свою работу до следу-

ющего поступления запроса, иначе переход к шагу 2.

2. Если становится известно, что с момента последнего запроса данного ИР произошло его изменение, то сенсор «посылает» роботу уведомление о найденном новом состоянии ИР. Для имитации процесса посылки уведомления определяется время Тувед когда сигнал уведомления «дойдёт» до робота (время Тувед имеет свой закон распределения).
3. Послав уведомление роботу, сенсор обновляет свои знания об ИР и завершает свою работу до следующего поступления запроса.

Алгоритм функционирования объекта Р в данной ситуации следующий:

1. В момент времени Тувед робот получает сигнал о том, что один из «наблюдаемых» ИР изменился.
2. Робот «посылает» запрос на скачивание ИР. Для имитации процесса посылки запроса используется закон распределения времени скачивания информационных ресурсов. С помощью этого закона определяется момент завершения процесса скачивания изменившегося ИР - Тскач.
3. В момент времени Тскач роботу становятся известны значения характеристик ИР.
4. Характеристики ИР анализируются роботом с целью обнаружения признаков произошедших «неучтённых» существенных изменений. Если таковых обнаружено не было, то осуществляется переход к шагу 1 (это тот случай, например, когда могло произойти изменение в HTML-разметке документа, но «содержательное» состояние документа не изменилось, т.е. изменение оказалось «несущественным»).
5. Если становится известно, что с момента последнего посещения роботом данного ИР всё же имеет место изменение, то робот обновляет хранимую в репозитории Хр информацию о содержимом ИР на новые сведения.
6. У обрабатываемого ИР робот изменяет значение признака свежести, хранимого в репозитории информации, на значение «истина».
7. Переход к шагу 1.

5. Программные средства для моделирования

В процессе поиска программной платформы для реализации компьютерных моделей нами было опробовано несколько сред. На первом этапе наиболее привлекательной была среда разработки программ Borland Delphi 6 [8]. В целях изучения основных принципов имитационного моделирования и в целях проверки «живучести» предлагаемых моделей нами была предпринята попытка написания программ, которые реализуют описанные ранее модели. За основу мы приняли соображение о том, что для фиксированного набора ИР последовательность событий изменений (потоки изменений) и запросов (потоки запросов) этих ИР одинаково пригодна для проведения экспериментов обеих моделей (роботов и сенсоров). То есть было предложено выделить в отдельную программу алгоритмы, максимально обще описывающие «жизнь» ИР, а результат работы программы сохранять в базе данных. Тем самым мы могли бы достигнуть экономии времени и ресурсов, т.к. на полученном наборе событий (логе) можно

проводить эксперименты с любыми компьютерными моделями («роботов» или «сенсоров»), уже не тратя вычислительные мощности на одинаковые операции моделирования событий, происходящих с ИР. В процессе написания программ мы столкнулись с тем, что в целях придания им большей гибкости и преемственности необходимо использование специальных библиотек, организующих модельную среду с едиными (для моделей) правилами. Однако такой поиск специализированных библиотек ничего не дал. Очевидно, что собственная реализация модельной среды заняла бы много времени, поэтому было признано нецелесообразным дальнейшее использование Delphi. В результате этого мы решили обратить свой взор в сторону специализированных сред имитационного моделирования.

Таким образом, на втором этапе мы познакомились со средой GPSS World [14]. Нас не остановило то, что эту среду большинство исследователей рекомендовало как среду моделирования систем массового обслуживания. По нашим предположениям, в этой среде наши модели также можно было легко реализовать. А то, что она имела достаточно проработанную организацию модельной среды, лишь усиливало наш интерес. В основу разработки программ было принято соображение о том, что целесообразнее всего будет одному ИР сопоставить один транзакт GPSS, а характеристики ИР хранить в параметрах этого транзакта. Реализация первой программы (что радовало) не заняла много времени (однако пришлось отказаться от принципа «лога», предложенного на первом этапе). Пробные запуски программы (это была модель «робота») обнадёживали, и поэтому мы попытались реализовать программу второй модели («сенсора»). На переделку программы модели «робота» под особенности модели «сенсора» нам потребовалось около суток суммарного времени. Тем самым реализуемость принципа преемственности программ видна «на лицо». А вот дальше, при начале крупномасштабных экспериментов с программами, нас ждало большое разочарование. Среда GPSS World от фирмы Minuteman [9] не могла справиться с большим объёмом вычислений (пределом оказалось моделирование «жизни» около 40000 ИР).

Поиски другой доступной специализированной среды имитационного моделирования к успеху не привёли, т.к. любая более-менее развитая система моделирования является коммерческим продуктом. Поэтому на третьем этапе мы снова решили вернуться к вопросу об использовании неспециализированной среды программирования, но только теперь мы решили не ограничиваться какой-то одной средой программирования, а провести подробный поиск и сравнение всех возможных вариантов. Основным принципом для поиска и выбора среды разработки теперь было наличие для неё проработанной библиотеки создания среды моделирования. По результатам поиска на первое место вышли две библиотеки для двух различных языков программирования:

1. C++SIM для языка C++.
2. SimPy для языка Python.

В силу многих факторов, библиотеку C++SIM мы решили оставить «на чёрный день» и попробовать разобраться с языком Python и с библиотекой SimPy, в частности.

Близкое рассмотрение языка Python показало, что с его помощью можно быстро писать довольно мощные и функциональные программы. Причём среда программирования Python имеется для большинства существующих, системных платформ и распространяется согласно лицензии совместимой с GPL (GPL-compatible) [10].

Библиотека SimPy представляет собой библиотеку классов для моделирования дискретных процессов [11]. Это классы, связанные с ведением календаря событий и обработкой списков (SIMULA-like).

Выбор языка Python в качестве основы создания программ моделей позволил реанимировать предложенный на первом этапе работ принцип «лога», т.е. когда единожды создаётся карта (журнал) событий, происходящих в системе, а затем с ней работают программы моделей. Для того чтобы в программах на языке Python появилась возможность работы с базой данных, потребовалось установить дополнительную библиотеку MySQLdb [12], а также установить и сам сервер баз данных MySQL [13].

Таким образом, в основу программных средств имитационного моделирования рассматриваемой системы положены:

- приведённые ранее концептуальные модели системы;
- объектно-ориентированный подход;
- принцип «лога» («журнала») событий;
- программные средства Python 2.2 и SimPy 0.5;
- программные средства MySQL 3.23.38.

В результирующий программный комплекс вошли следующие программы:

1. SimPages – создаёт базу данных, которая содержит описание набора моделируемых ИР и содержит журнал событий, происходивших с каждым ИР (в частности, журналы запросов и журналы изменений).
2. SimRobot – на основе базы, созданной SimPages, программа моделирует поведение системы в случае варианта концепции робота.
3. SimSensor – на основе базы, созданной SimPages, программа моделирует поведение системы в случае варианта концепции сенсоров.

6. Результаты компьютерного моделирования

Прежде чем начать исследование системы с помощью компьютерных экспериментов, мы решили поставить серию пробных экспериментов. Цель этих экспериментов была в том, чтобы получить возможность оценить адекватность моделей и пригодность программного комплекса к проведению любых необходимых экспериментов. В качестве примера приведём один из компьютерных экспериментов, в котором исследуется поведение системы при мониторинге информационного состояния 200000 информационных ресурсов.

6.1. Начальные условия

Моделируется функционирование системы мониторинга информационного состояния 200000 информационных ресурсов (под ИР понимается html-страница)

Таблица 1

Номер состояния-изменения	1	2	3	4	5	6
Относительная частота	0,083	0,125	0,125	0,25	0,25	0,166

в течение недели (6048000 единиц) модельного времени. За единицу модельного времени приняты 10 миллисекунд (msec) реального времени. Исходный размер (байт) каждого информационного ресурса определяется использованием равномерного распределения с минимальным значением, равным 65 байт (минимальный размер html-разметки страницы), и максимально возможным значением, равным 122880 байт (рекомендуемый максимальный размер для страниц). Время скачивания любого ИР определяется равномерным распределением с минимальным значением, равным единице модельного времени, и максимальным значением, равным 40 единицам модельного времени.

Время поступления запросов на скачивание распределено экспоненциально со значением интенсивности, равной 70 (10 запросов в день). Время наступления события изменения состояния или содержимого ИР распределено экспоненциально со значением интенсивности, равной 6 (6 изменений в неделю). Определены 6 типов возможных состояний-изменений ИР:

1. Ошибка 403
2. Ошибка 404
3. Ошибка 500
4. Уменьшение размера страницы
5. Увеличение размера страницы
6. Страница доступна (нет изменений)

Относительная частота появления определённого номера состояния- изменения при наступлении события изменения задаётся табл.1.

Считается, что в начальный момент времени страница находится в 6-м состоянии (т.е. доступна) и репозиторий хранит информацию о текущем содержимом ИР (т.е. репозиторий «свежий»).

Перечислим возможные варианты, при которых считалось, что страница всё же не изменила своё состояние (или своё содержимое) в результате наступления события изменения:

1. Случай, когда номер нового изменения равен номеру предыдущего и одновременно принадлежит следующему множеству номеров изменений: 1, 2, 3, 6.
2. Случай, когда номер нового изменения равен 6, а номер предыдущего изменения равен одному из следующих номеров изменений: 4, 5.

Во всех остальных случаях считалось, что страница изменила своё состояние или изменила своё содержимое в результате наступления очередного события изменения.

При моделировании системы, построенной по варианту концепции роботов, мониторингом занимается только один экземпляр робота, который должен «обходить» весь набор информационных ресурсов.

При моделировании системы, построенной по варианту концепции сенсоров, мониторингом занимаются сенсоры, которые посылают роботу сигнал уведомления. Время поступления уведомления от сенсора любого ИР определяется равномерным распределением с минимальным значением, равным единице модельного времени, и максимальным значением, равным 3 единицам модельного времени. При получении сигнала уведомления робот инициирует процесс скачивания изменившейся страницы независимо от количества запущенных в ответ на поступившие ранее сигналы процессов скачивания.

6.2. Результаты эксперимента

На первом этапе проводилось создание (инициализация) «виртуального» набора страниц и создание полного журнала событий (SimPages). Эта работа продолжалась 27 часов 8 минут реального времени. За это время были описаны свойства всех страниц и все события (запросы и изменения), связанные с ними. Смоделированный начальный объём набора (200000) «виртуальных» страниц равен 12278728299 байтам. База данных с описанием страниц и журналом заняла 565999048 байт дискового пространства.

На втором этапе выполнялась имитация работы системы в случае концепции роботов (SimRobot). Эта работа продолжалась 1 час 10 минут реального времени. В течение 6048000 единиц модельного времени робот сделал 2 полных цикла обхода. Первый цикл закончился в 3122177 единицу модельного времени, а второй закончился в 5876590 единиц модельного времени. Затем начался третий цикл, который не был полностью закончен в связи с завершением моделирования. Суммарный объём скачанной роботом к себе информации равен 16186441922 «виртуальных» байт. Минимальное время ожидания страничкой завершения её скачивания роботом с момента её же первого «неучтённого» изменения равно 4 единицам модельного времени. Максимальное время ожидания завершения скачивания равно 3119397 единицам модельного времени. Среднее значение времени ожидания равно 1642301,8 единиц модельного времени.

На третьем этапе выполнялась имитация работы системы в случае концепции сенсоров (SimSensor). Эта работа продолжалась 7 часов 4 минуты реального времени. Суммарный объём скачанной роботом к себе информации равен 33168534243 «виртуальным» байам. Минимальное время ожидания начала скачивания роботом страничкой с момента её первого «неучтённого» изменения равно единице модельного времени. Максимальное время ожидания начала скачивания равно 1104508 единицам модельного времени. Среднее значение времени ожидания равно 85179,4 единиц модельного времени. Максимальное количество одновременно обрабатываемых роботом уведомлений об изменившихся страницах равно 12.

Во время выполнения второго и третьего этапов вёлся сбор дополнительных статистических данных. В частности, с шагом 10000 единиц модельного времени измерялась свежесть репозитория собранной информации. Результаты этого замера можно видеть на рис.3. Начиная примерно с 3500000 единиц модельного времени, система входит в стационарный режим в обоих вариантах построе-

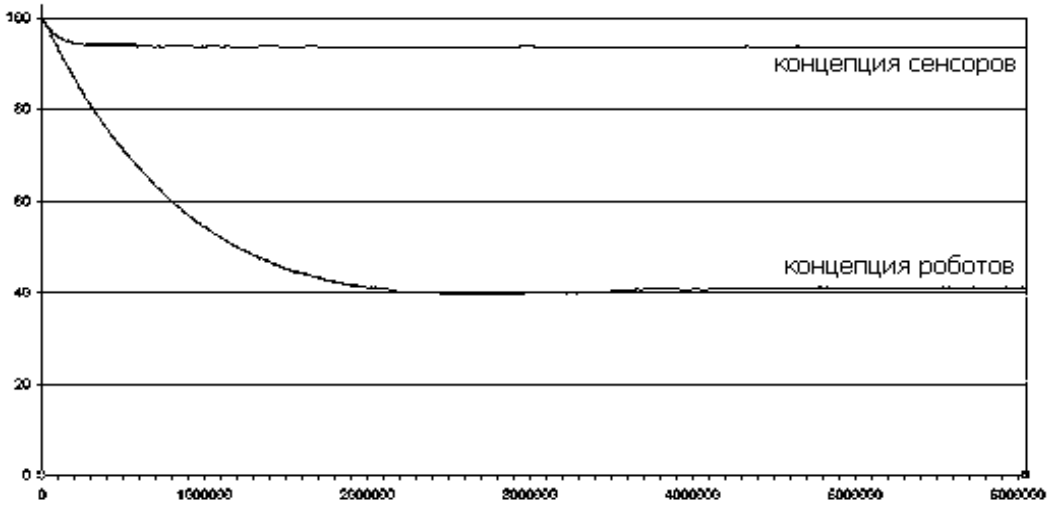

Рис. 3. График изменения критерия свежести репозитория собранной информации

ния системы. Среднее значение критерия свежести для концепции сенсоров в стационарном режиме равно 93,6%. Среднее значение критерия свежести для концепции роботов в стационарном режиме равно 40,7%.

Остаётся описать тестовую установку, на которой получены данные результаты. Эксперимент проводился на Pentium 4-1.6GHz; 512Mb ОЗУ; 60Gb HDD; Windows 2000 Professional.

6.3. Обсуждение результатов

Как и следовало ожидать, концепция сенсоров показывает лучшие результаты по критерию свежести, чем это делает концепция роботов. В то же самое время объём скачанной информации у концепции роботов меньший. Однако эти полученные значения критериев имеют логичное объяснение. Благодаря большой интенсивности запросов ИР сенсоры быстрее обнаруживают произошедшие изменения и в то же время заставляют робота чаще производить скачивание ИР (в этом случае мы говорим о «полезном скачивании»).

7. Направление дальнейших работ и исследований

- Требуется дополнительный анализ существования всевозможных «узлов времени» в моделях и, соответственно, потребуется доработка ПО-моделирования с целью их устранения. Например, поступление запроса на страницу и наступление изменения произошли в одну и ту же единицу модельного времени. Как рассматривать такое совпадение? Чем оно чревато для наших результатов?
- Требуется доработка ПО-моделирования дополнительными средствами сбора статистики. Например, хотелось бы знать средние значения количества «полезных» и «бесполезных» скачиваний ИР роботом.

- Уже сейчас можно прикинуть возможную продолжительность одного эксперимента по моделированию системы с 2млн. ИР. По нашим скромным прикидкам получилось, что такой эксперимент может проводиться в течение примерно 20 дней. Поэтому становится актуальной задача создания плана экспериментов и затем проведение экспериментов согласно этому плану. Анализ полученных результатов также требует предварительной подготовки.

Заключение

Результаты поставленных экспериментов позволили нам признать модели пригодными к дальнейшему использованию (т.е. адекватными) и наметить направления будущих работ, в том числе по улучшению программного комплекса моделирования.

Литература

1. Земсков И.А. *Сбор информации о доступных ресурсах Интернет.* – `ftp://cmm.univer.omsk.su/pub/sbornik9/zemskov.zip`
2. Brandman O., Cho J., Garcia-Molina H., Shivakumar N. *Crawler-Friendly Web Servers.* – `http://rose.cs.ucla.edu/ cho/papers/cho-server.pdf`
3. Р. Шеннон. *Имитационное моделирование систем - искусство и наука.* М.: Мир, 1978.
4. Shkapenyuk V., Suel T. *Design and Implementation of a High-Performance Distributed Web Crawler.* – `http://cis.poly.edu/tr/tr-cis-2001-03.pdf`
5. Najork M., Heydon A. *High-Performance Web Crawling.* – `http://citeseer.nj.nec.com/najork01highperformance.html`
6. Земсков И.А. *О концепции индексации информационных ресурсов сети Интернет.* – `ftp://cmm.univer.omsk.su/pub/sbornik8/zemskov.zip`
7. Советов Б.Я., Яковлев С.А. *Моделирование систем.* М.: Высшая школа, 1985.
8. Borland Delphi home page – `http://www.borland.com/delphi/`
9. Minuteman Software – `http://www.minutemansoftware.com/`
10. Python home page – `http://www.python.org/`
11. SimPy: A Python-based simulation package – `http://sourceforge.net/projects/simpy/`
12. Python interface to MySQL – `http://sourceforge.net/projects/mysql-python`
13. MySQL home page – `http://www.mysql.com/`
14. Шрайбер Т.Дж. *Моделирование на GPSS.* М.: Машиностроение, 1980.