

РАЗРАБОТКА ДИНАМИЧЕСКОЙ МОДЕЛИ КОМПЬЮТЕРНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Лаздынь С.В., Землянская С.Ю.

Донецкий национальный технический университет,
кафедра автоматизированных систем управления

E-mail: slazd@ukr.net

Abstract

Lazdyn S.V., Zemlianskaya S.Y. The computer information system dynamic model's development. The computer information system dynamic model's development is discussed. The model based on the object-oriented method is used for the operating benefits evaluation and recommendations about system's productivity increasing are issued. The experimental model testing has been made.

Характеристика проблемы

Современные компьютерные информационные системы (КИС) – это сложный комплекс взаимосвязанных и согласованно функционирующих аппаратных и программных компонентов: компьютеров, коммуникационного оборудования, операционных систем, сетевых приложений [5, 9].

Главным требованием, предъявляемым к компьютерным информационным системам, является выполнение КИС её основной функции – обеспечение пользователям возможности быстрого доступа к необходимым им данным, расположенным зачастую на различных компьютерах [7]. При проектировании и эксплуатации компьютерных информационных систем необходимо обеспечить согласованную и эффективную работу всех их составляющих. Для того, чтобы система работала эффективно, необходимо решить ряд задач, связанных с определением конфигурации (топологии) компьютерной сети, параметров компьютерного и коммуникационного оборудования, с распределением приложений и данных по узлам системы, с анализом производительности КИС.

Для решения указанных выше задач необходимо создать модель КИС, которая бы обеспечила достаточно точное отображение всех динамических процессов, протекающих в КИС, получение оценок работы отдельных устройств и системы в целом. Несмотря на проведенные исследования, проблема построения динамической модели КИС не получила окончательного решения и имеет актуальное значение.

Краткий анализ проведенных исследований по моделированию КИС

Основными видами моделирования, которые могут быть использованы для КИС, являются: - физическое и математическое моделирование [1]. Примером физического вида моделирования служит пилотная сеть, с помощью которой изучается принципиальная возможность построения системы на основе тех или иных компьютеров, коммуникационных устройств, операционных систем и приложений. Возможности физического моделирования довольно ограничены. Оно позволяет решать отдельные задачи при задании небольшого количества сочетаний исследуемых параметров системы. Проверка на практике большого количества разных компонентов и их сочетаний связана не только с большими усилиями и временными затратами, но и с немалыми материальными затратами, поэтому, при создании КИС во многих случаях предпочтительным оказывается использование математического моделирования.

Для математического описания информационно-вычислительных процессов, протекающих в компьютерной информационной системе применяются методы теории массового обслуживания [4, 8]. Однако, при построении моделей на базе теории СМО используется ряд упрощений и ограничений, которые снижают точность получаемых результатов моделирования. Поэтому указанный класс моделей применим в основном для получения усредненных оценок несложных КИС на этапе эскизного проектирования систем.

Другим классом математических моделей, используемых в КИС, являются имитационные модели [2, 4]. В таких моделях процессы, протекающие в компьютерной информационной системе отображаются соответствующими алгоритмическими конструкциями, реализованными с помощью выбранного языка программирования. К недостаткам имитационных моделей следует отнести то, что с усложнением структуры, изменением конфигурации и масштабов моделируемой системы, возрастает сложность и объем программного кода самой модели. Это делает затруднительным использование таких моделей для анализа работы реальных КИС, состоящих из десятков, сотен, а иногда и тысяч единиц оборудования различных типов.

Для моделирования вычислительных сетей разработаны специальные программные системы, например: Prophecy, COMMNET III, NetMaker XA, MIND, Perfomance и др. [6]. Такие программные системы генерируют модель сети на основе исходных данных о ее топологии и используемых протоколах, об интенсивностях потоков запросов между компьютерами сети, протяженности линий связи, о типах используемого оборудования и приложений. К недостаткам указанных выше программных систем следует отнести их высокую стоимость (от нескольких до десятков тысяч долларов США), узкую специализацию на моделирование вычислительных сетей, а также то, что для построения моделей сетей в них используются рассмотренные ранее методы теории СМО или имитационное моделирование.

В целом, проведенный анализ используемых методов и средств моделирования показал, что они обладают рядом недостатков, не обеспечивают достаточную точность, не позволяют моделировать сложные компьютерные информационные системы, включающие большое количество компьютерного, периферийного и сетевого оборудования, с учетом изменяющихся условий их функционирования.

Поэтому необходимо создание новой, более совершенной модели КИС, обладающей высокой степенью адекватности, позволяющей моделировать системы с различной структурой и конфигурацией.

Структурный анализ и выбор типовых компонентов КИС.

Компьютерная информационная система как объект моделирования представляет собой совокупность соединенных друг с другом и взаимодействующих компонентов. Весь комплекс программно-аппаратных средств можно разделить на несколько уровней. На первом уровне системы находится аппаратный слой стандартизованных компьютерных платформ. В настоящее время широко и успешно применяются компьютеры различных классов – от персональных до мейнфреймов и суперЭВМ. Второй уровень – это коммуникационное оборудование. Кабельные системы, повторители, мосты, коммутаторы, маршрутизаторы и модульные концентраторы из вспомогательных компонентов сети превратились в основные наряду с компьютерами и системным программным обеспечением как по влиянию на характеристики КИС, так и по стоимости. К третьему уровню, образующему программную платформу системы, относятся операционные системы (ОС), сетевые базы данных, почтовые системы, средства архивирования данных, приложения пользователей. и другие программные средства.

Проведенный структурный анализ показал, что в составе КИС можно выделить сле-

дующие типовые компоненты, параметры которых влияют на производительность системы: компьютерное оборудование (серверы, рабочие станции), сетевое оборудование (концентраторы, коммутаторы), программное обеспечение (приложения пользователей), данные (запросы, ответы, пакеты).

Для моделирования типовых компонентов КИС выбран объектно-ориентированный метод. Его преимущества по сравнению с традиционным имитационным моделированием – простота и наглядность отображения элементов и функций исследуемого объекта и возможность построения моделей для КИС с произвольной структурой. Для моделирования функционирования указанных компонентов разработаны соответствующие классы объектов, описаны их свойства и методы.

Разработка объектных моделей типовых компонентов КИС

В состав компьютерного оборудования КИС входят серверы и рабочие станции. Эти типовые компоненты отличаются выполняемыми функциями, но имеют ряд общих свойств, связанных с наличием вычислительного устройства (процессора), оперативной памяти, возможностью приема и передачи пакетов информации с помощью сети. Поэтому для моделирования объектов этих типов разработан базовый класс «Процессорный узел» (TProcUnit). Состав базового класса «Процессорный узел» представлен на рисунке 1.

Основной задачей процессорного узла является прием и передача сообщений, разбитых на пакеты, т.е. разбиение сообщения на пакеты при передаче и компоновка сообщения при приеме из поступающих на этот узел пакетов. В соответствии с этим необходимо знать следующие характеристики процессорного узла: объем оперативной памяти (Мбайт), скорость передачи сетевой карты (Мбит/с); размер пакета, передаваемого в сеть (Кбит); размер буфера сетевой карты (Кбайт); входная очередь пакетов; выходная очередь пакетов. Когда активизируется процессорный узел, то вызывается метод обработки пакета процессорным узлом: *Transfer()* – передача пакета, *ReceivePacket()* – приём пакета.

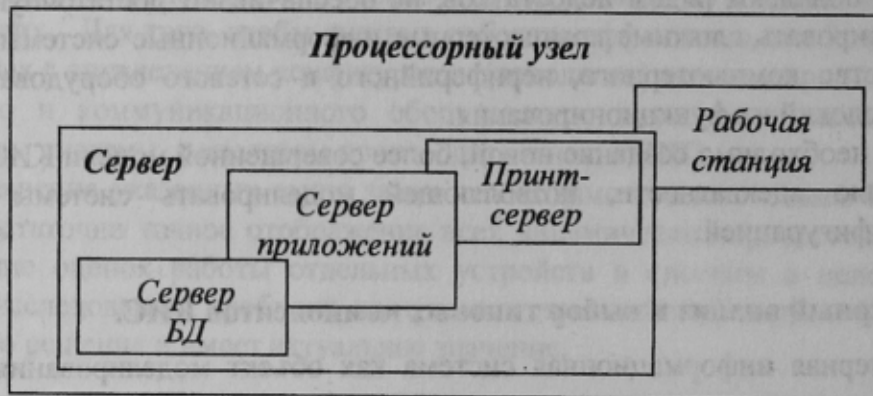


Рисунок 1. Состав класса «Процессорный узел»

Серверы КИС как объекты модели являются производными от базового класса «Процессорный узел» и выполняют такие функции как прием и обработка запросов. Прием запроса происходит одинаково для всех видов серверов, поэтому можно выделить базовый класс TServer, осуществляющий прием запросов из сети (метод *ReceiveMessage()*) и их обработку (метод *FullfillQuery()*) и обладающий следующими свойствами: очередь запросов; длина очереди запросов; время освобождения; количество отказов. В зависимости от выполняемых функций серверы можно разделить на сервер баз данных, сервер приложений, принт-сервер. Для каждого из видов серверов в модели разработан соответствующий производный от TServer класс: TDBServer, TAppServer и TPrintServer.

Сервер баз данных (класс TDBServer) является основным средством хранения и

обработки данных в компьютерных информационных системах. Данный класс характеризуется следующими свойствами: объем оперативной памяти, общий и свободный объемы дискового пространства, входная очередь запросов, выходная очередь ответов, производительность – максимальное количество транзакций ТРС-С, которые могут быть обработаны сервером за минуту.

Основные методы, используемые этим классом: метод *FullfillQuery()* - обработка запроса, метод *FormMessage()* - формирование ответа на обработанный запрос.

Сервер приложений (класс *TAppServer*) характеризуется набором приложений, запускаемых на данном сервере, объемом доступного ОЗУ (Мбайт) для запуска приложений. Каждое приложение, запускаемое на этом сервере имеет определенное время выполнения. Методы данного класса: *FullfillQuery()* – обработка запроса – проверяет наличие приложения, запрашиваемого запросом: если такого приложения нет или запустить его невозможно (недостаточен объем ОЗУ), то запрос получает отказ и удаляется из входного списка сообщений; метод формирования ответа для запросившей запуск этого приложения рабочей станции – *FormRequest()*.

Принт-сервер (класс *TPrintServer*) характеризуется скоростью печати знаков(байт)/сек, и объемом буферной памяти (Мбайт). Метод обработка запроса (*FullfillQuery()*) включает проверку возможности выполнения запроса (наличие свободного места в ОЗУ), занятие сервера на время, пропорциональное объему запроса и удаление запроса из входного списка сообщений после обработки.

Рабочая станция (класс *TWorkStation*) является источником запросов в КИС и характеризуется такими свойствами: набором запускаемых приложений, объемом оперативной памяти, дисковым пространством для хранения данных. На рабочей станции осуществляются следующие основные операции: запуск приложений с заданной интенсивностью (методы *InitWS()*, *StartApplication()*); приём ответов на запросы (метод *ReceiveAnswer()*); окончание работы приложения (метод *DelApplication()*).

Концентратор (класс *THub*) повторяет входные сигналы на всех своих портах, кроме того, поддерживает в сети физическую топологию звезда, но логическая топология остается без изменений – общая шина, т.е. концентратор является одноканальным объектом и необходимо моделировать передачу пакетов по общей шине с разрешением коллизий. Следовательно, концентратор должен иметь следующие параметры: сетевой адрес; количество портов; скорость передачи портов, Мбит/с; интервал отсрочки; максимальное число попыток передачи; максимальное число возрастания диапазона паузы; длительность паузы после коллизии, секунд; время обработки, секунд; очередь пакетов; момент времени освобождения. Класс *THub* использует следующие методы: *Work()* – обработка пакета; *PutInQueue()* – постановка пакета в очередь; *Transfer()* – метод передачи пакета следующему узлу сети; формируется время обслуживания; *FromQueue()* – метод выбора из очереди пакета с минимальным временем поступления. Если пакетов с таким временем несколько, то для каждого из них вызывается метод определения коллизии *CollisionDetect()*

Коммутатор (класс *TSwitch*) осуществляет передачу пакета только между теми портами, к которым подключены соответствующие узлы (источник и получатель) в соответствии с маршрутной таблицей, т.е. в данном случае коллизии при передаче по общей шине не возникают и коммутатор можно рассматривать как многоканальный объект, который должен обладать следующими свойствами: сетевой адрес; количество портов; скорости передачи портов (Мбит/с); очереди к портам; моменты времени освобождения портов; время обработки (поиск адреса узла, на который необходимо передать пакет, в маршрутной таблице); адресная таблица, ставящая в соответствие номеру порта сетевой адрес узла. Методы, используемые классом *TSwitch*: *Work()* – анализ состояния пакета; *FindPort(Node)* – поиск порта коммутатора, соответствующего узлу с сетевым адресом *Node*

по адресной таблице; *Transfer()* – передача пакета; *FromQueue()* и *PutInQueue()* – методы извлечения и помещения пакета в очередь.

Приложение (класс *TApplication*). Приложения запускаются на рабочих станциях с заданной интенсивностью. Интенсивность запуска приложений – случайная величина, зависящая от множества факторов. Для приложений характерны следующие параметры: математическое ожидание интервала времени запуска; дисперсия интервала времени запуска; приоритет; набор запросов, генерируемых приложением. Приложения используют следующие методы: *TApplication()* – конструктор приложения; *FormQuery()* – формирование запроса (для чтения, модификации наборов данных, запуска приложений на серверах приложений или печати данных на принт-серверах); *ReceiveMessage()* – активизация приложения после получения ответа на отправленный запрос.

Запрос, ответ (класс *TQuery*). Для получения данных, их модификации или обработки используются запросы к серверам. Для запроса характерны следующие свойства: вид запроса (чтение, запись или обработка данных); узел-источник запроса; узел-обработчик; маршрут прохождения запроса от источника к получателю (адреса компонентов системы, через которые проходит запрос от источника к получателю); количество транзакций, необходимое для выполнения запроса; объем запроса; объем ответа на запрос; приоритет запроса; код приложения (для запуска приложения на сервере приложений); текущее количество пакетов (при сборке/разборке при передаче по сети); момент времени отправления запроса и прибытия его на конечный узел – для оценки задержек, возникающих при передаче запроса по сети. Конструктор запроса (*TQuery()*) формирует запрос, параметры которого выбираются из справочника по коду запроса. Метод *FormPath()* формирует маршрут прохождения пакета от начального узла к конечному по матрице сетевых взаимосвязей в виде последовательности сетевых адресов. Метод *CreatePacket()* производит формирование пакетов из исходного запроса.

Пакет (класс *TPacket*). Для передачи сообщений по сети осуществляется их разбивка на пакеты в соответствии с протоколом передачи, используемым в сети. В соответствии с этим пакет обладает следующими свойствами: информация о сообщении-родителе, к которому относится пакет; адрес узла-источника, с которого был отправлен пакет; адрес узла-получателя пакета; маршрут прохождения запроса от источника к получателю (адреса узлов сети, через которые проходит запрос от источника к получателю); текущее число попыток передачи пакета; порядковый номер в сообщении; приоритет; адрес узла сети, на котором пакет находится в текущий момент; время прибытия пакета на текущий узел. Методы класса: конструктор *TPacket()* – производит формирование пакета; метод *GoToNext()* – переводит пакет в следующую фазу обработки; метод *InQueue()* – помещает пакет в очередь; метод *CollisionDetect()* – производит обработку конфликтных ситуаций при передаче; метод *DelPacket()* – выполняет удаление пакета из общего списка пакетов при достижении узла назначения.

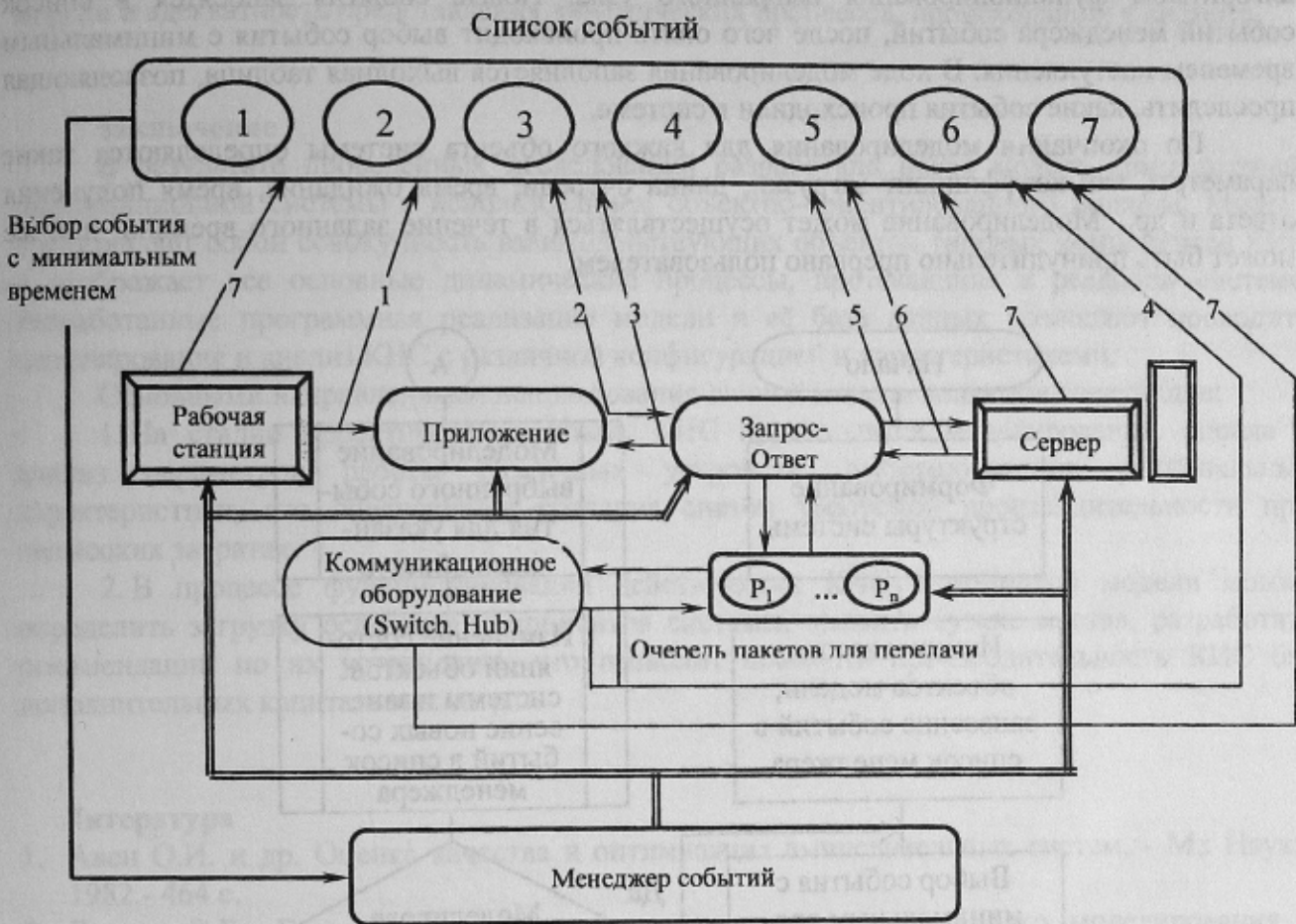
Построение общей динамической модели КИС

Модель КИС представляет собой совокупность взаимодействующих объектов разработанных классов ее типовых компонентов. Углубленная структура взаимодействия объектов модели представлена на рисунке 2. Для организации взаимодействия между объектами в модель добавлен новый класс – менеджер событий (*TManager*), содержащий информацию о структуре системы и процессах, которые в ней происходят.

Менеджер событий содержит сведения обо всех объектах, входящих в моделируемую КИС. Кроме того, для управления работой модели необходимо обрабатывать события, происходящие в системе, поэтому менеджер событий содержит информацию о состоянии объектов в виде списка происходящих событий: запуск приложения; появление запроса; переход пакета между узлами, освобождение узла системы. После обработки события оно

удаляется из списка событий. Таким образом, можно выделить следующие основные свойства менеджера событий: списки компонентов КИС; таблица связей между узлами КИС; список пакетов, передающихся по сети; список событий.

Менеджер событий осуществляет координацию функционирования отдельных компонентов модели путем отслеживания событий, происходящих в системе: просматривает очередь событий, выбирает событие с минимальным временем наступления и активизирует соответствующий метод обработки события того объекта, состояние которого связано с этим событием. В результате взаимодействия объектов через посредство менеджера событий происходит моделирование реальных процессов, протекающих в КИС: запуск приложений, формирование запросов, передача пакетов, обработка запросов, передача ответов и т.д.



Типы событий:

- | | |
|-------------------------------------|--------------------------------------|
| 1 – Запуск приложения | 5 – Поступление запроса на обработку |
| 2 – Формирование запроса | 6 – Формирование ответа |
| 3 – Получение ответа | 7 – Освобождение узла сети |
| 4 – Поступление пакета на узел сети | |

Рисунок 2. Схема взаимодействия объектов модели КИС

Укрупненный алгоритм объектного моделирования представлен на рисунке 3. В начале формируется структура модели системы, для которой задаются количества и типы входящего в ее состав оборудования, его технические характеристики, способы его соединения. Затем производится создание и инициализация объектов модели в соответствии с ее структурой. Далее для объектов – рабочих станций формируется список запущенных приложений, которые в свою очередь генерируют запросы к серверам. Каждый запрос

разбивается на пакеты для передачи по сети. Когда все пакеты какого-либо запроса достигают пункта назначения, начинается обработка этого запроса сервером. По окончании обработки запроса сервером формируется ответ, который также разбивается на пакеты и передается по сети до получения его приложением, сгенерировавшим запрос.

С созданием и функционированием указанных выше динамических объектов (приложений, запросов, ответов, пакетов) связаны соответствующие события в модели, которые заносятся в список менеджера событий. Менеджер выбирает событие с минимальным временем наступления, затем определяет тип события и объект, к которому относится это событие; для выбранного объекта вызывается соответствующий метод обработки, в результате чего могут быть сформированы новые события в соответствии с алгоритмом функционирования выбранного узла. Новые события заносятся в список событий менеджера событий, после чего опять происходит выбор события с минимальным временем наступления. В ходе моделирования заполняется выходная таблица, позволяющая проследить, какие события происходили в системе.

По окончании моделирования для каждого объекта системы определяются такие параметры, как коэффициент загрузки, длина очереди, время ожидания, время получения ответа и др. Моделирование может осуществляться в течение заданного времени, или же может быть принудительно прервано пользователем.

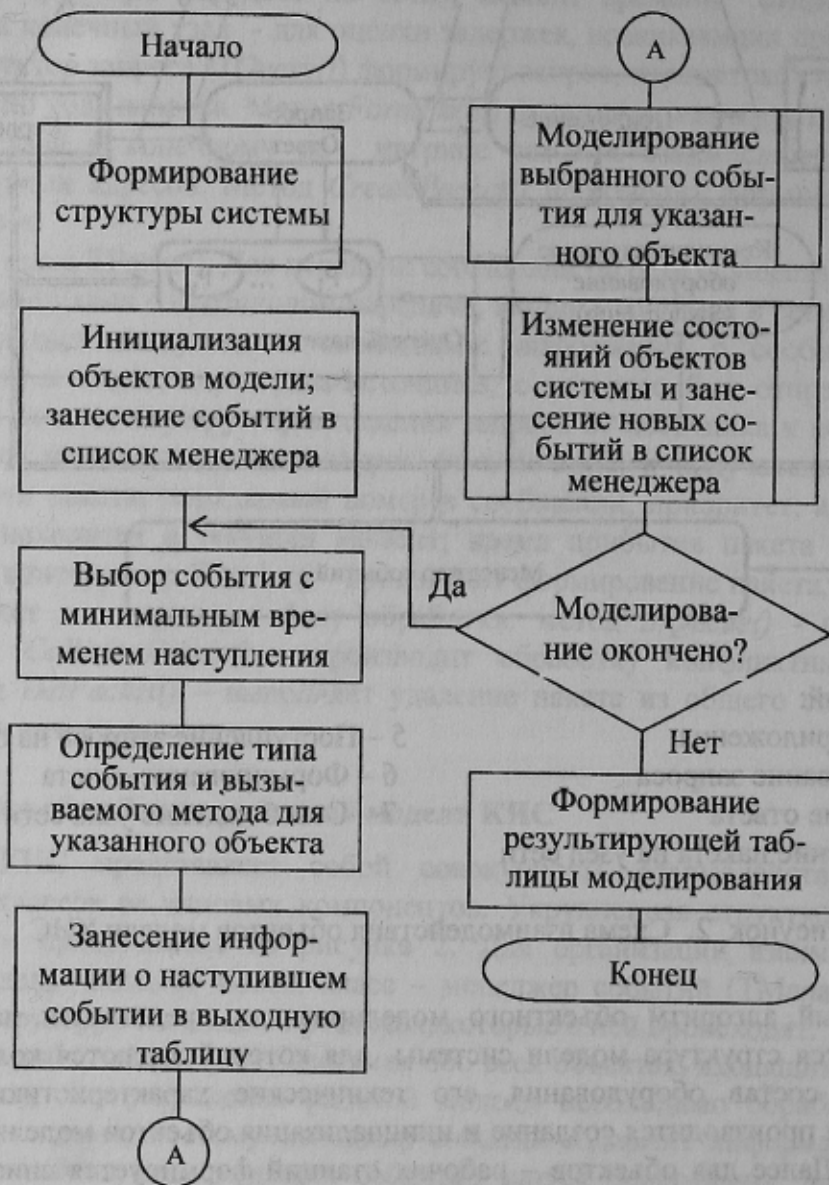


Рисунок 3. Алгоритм моделирования КИС

Программная реализация модели.

На основе построенной объектной модели КИС разработана программа моделирования с использованием системы визуального программирования C++ Builder. Для хранения параметров моделируемой КИС и результатов моделирования разработана база данных в формате Paradox. В состав базы данных входят таблицы, содержащие информацию о структуре КИС, характеристики входящего в ее состав оборудования различных типов, сведения о приложениях и генерируемых ими запросов.

Проведена экспериментальная проверка разработанной модели на примере компьютерной системы, включающей несколько рабочих станций, два сервера, коммутатор и концентратор. Проверка модели на реальных данных подтвердила работоспособность модели и адекватность представления динамических процессов, происходящих в системе.

Заключение

В результате проведенных исследований разработана новая модель компьютерной информационной системы с использованием объектно-ориентированного подхода. Модель представляет собой совокупность взаимодействующих объектов типовых компонентов КИС и отображает все основные динамические процессы, протекающие в реальной системе. Разработанные программная реализация модели и её база данных позволяют проводить моделирование и анализ КИС с различной конфигурацией и характеристиками.

Основными направлениями использования данной модели являются следующие:

1. На стадии проектирования новых КИС производится моделирование, оценка и анализ параметров работы различных устройств, выбираются их рациональные характеристики, что обеспечивает создание систем требуемой производительности при невысоких затратах.
2. В процессе функционирования действующих КИС с помощью модели можно определить загрузку основных компонентов системы, выявить «узкие места», разработать рекомендации по их устранению, что позволит повысить производительность КИС без дополнительных капитальных затрат.

Литература

1. Авен О.И. и др. Оценка качества и оптимизация вычислительных систем. - М.: Наука, 1982.- 464 с.
2. Бигдан В.Б., Гусев В.В. Становление и развитие имитационного моделирования в Украине. <http://simulation.org.ua>
3. Компьютерные системы и сети. - М.: Финансы и статистика, 1999.- 464 с.
4. Марков А.А. Моделирование информационно-вычислительных процессов: Учебное пособие для вузов. - М: Изд-во МГТУ им. Н.Э. Баумана, 1999.- 360 с.
5. Науманн Ш. Компьютерная сеть: проектирование, создание, обслуживание. - М.: ДМК, 2000.
6. Олифер Н.А., Олифер В.Г. Средства анализа и оптимизации локальных сетей. <http://www.citforum.ru/nets/optimize>
7. Пятибратов А.П. Вычислительные системы, сети и телекоммуникации: Учебное пособие для вузов.- М: Наука, 1998.- 387 с.
8. Шварц М. Сети связи: протоколы, моделирование и анализ. Т. 1,2. - М: Наука, 1992
9. Яковлев Ю.С. О проблеме интеграции аппаратной и программной платформ при создании компьютерных систем. // Управляющие системы и машины. 2000. № 5. - С. 138-142.