

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Федеральное государственное образовательное бюджетное
учреждение высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
им. проф. М. А. БОНЧ-БРУЕВИЧА»

А.Н.Губин

Современные методы проектирования информационных систем

Краткий конспект лекций

Раздел 3. Основные принципы бездефектного проектирования ИС.

САНКТ-ПЕТЕРБУРГ

2015

Содержание

Раздел 1. Общая характеристика процессов проектирования ИС.

- 1.1. Информационные системы как объекты проектирования.*
- 1.2. Методологические основы проектирования ИС.*
- 1.3. Нормативная база проектирования ИС.*
- 1.4. Стадии проектирования ИС.*
- 1.5. Состав и содержание проектной документации.*

Раздел 2. Основные технологии проектирования ИС.

- 2.1. Структурный подход к проектированию ИС.*
- 2.2. Особенности современных методов и средств проектирования ИС, основанных на CASE-технологии.*

Раздел 3. Основные принципы бездефектного проектирования ИС.

- 3.1. Задачи моделирования ИС.*
- 3.2. Высокоуровневое моделирование ИС..*
- 3.3. Низкоуровневое моделирование средств реализации ИС.*
- 3.4. Имитационное моделирование ИС.*
- 3.5. Анализ и оценка производительности ИС.*

Раздел 4. Типизация проектных решений.

4.1. Особенности технологии типового проектирования ИС.

4.2. Основные методы типового проектирования ИС.

4.3. RAD – технология проектирования.

Раздел 5. Управление проектами ИС.

5.1. Жизненный цикл ИС.

5.2. Расширение и обновление ИС.

5.3. Сопровождение, контроль эффективности и качества ИС.

5.4. Мониторинг безопасности ИС.

5.5. Перспективы и основные направления развития ИС и средств их проектирования.

Раздел 3. Основные принципы бездефектного проектирования ИС..

"Бездефектное" проектирование информационных систем реализуется комплексным применением высокоуровневого моделирования (моделирования функций или бизнес-процессов) предприятия и низкоуровневого моделирования вычислительной системы.

Общая условная схема бездефектного проектирования информационной системы приведена на рис. 3.1.

Использование высокоуровневого моделирования позволяет гарантировать полноту и правильность выполнения информационной системой функций, определенных заказчиком. То есть построенная модель удовлетворяет всем требованиям по функциональности (система должна выполнять то, что задумано). Однако гарантировать, что конкретная реализация вычислительной системы на предприятии будет выполнять эти функции, высокоуровневое моделирование не может.

К системам высокоуровневого моделирования относятся такие системы, как ARIS, Rational Rose. С их помощью реализуются принципы структурного анализа, когда предприятие представляется в виде сложной системы, состоящей из разных компонентов, имеющих различного рода взаимосвязи друг с другом. Эти средства позволяют определить и отразить в моделях основные компоненты предприятия, протекающих процессов, используемой информации, а также представить взаимосвязи между этими компонентами.

Создаваемые модели представляют собой документированную совокупность знаний об ИС предприятия - о его организационной структуре взаимодействиях между предприятием и прочими субъектами рынка, составе и структуре документов, последовательностях шагов процессов, должностных инструкциях отделов и их сотрудников.

Моделирование функций вычислительной системы напрямую сегодня не представляется возможным. Данная задача в полном объеме не разрешима. Однако возможно моделирование работы системы в динамике (динамическое моделирование), при этом его результаты позволяют по косвенным показателям судить о функционировании всей системы.

Так, мы не можем проверить правильность функционирования сервера базы данных и программного обеспечения, однако по выявляемым задержкам на сервере, необслуженным запросам и т. д. мы можем сделать вывод о его работе.

Таким образом, рассматриваемые системы предназначены не для функционального моделирования вычислительных систем (это, к сожалению, невозможно), а для динамического их моделирования.

Моделирование вычислительной системы позволяет произвести более точный, по сравнению с экспертными оценками, расчет необходимой производительности отдельных компонентов и всей системы в целом, в том

числе системного и прикладного программного обеспечения.. При этом появляется возможность использовать не максимальные значения характеристик используемого вычислительного оборудования, а характеристики, учитывающие, специфику использования этого оборудования в конкретном учреждении.

Основу моделирования составляют модели оборудования и процессов (технологий, программного обеспечения), используемых при работе интересующего объекта. При моделировании на компьютере воспроизводятся реальные процессы в обследуемом объекте, исследуются особые случаи, воспроизводятся реальные и гипотетические критические ситуации. Основным достоинством моделирования является возможность проведения разнообразных экспериментов с исследуемым объектом, не прибегая к физической реализации, что позволяет предсказать и предотвратить большое число неожиданных ситуаций в процессе эксплуатации, которые могли бы привести к неоправданным затратам, а может, и к порче оборудования.

В случае моделирования вычислительных систем таким объектом является информационная система, определяющая способы получения, хранения, обработки и использования различной корпоративной и внешней информации.

В процессе моделирования обеспечивается решение следующих задач:

- определение минимально необходимого, но обеспечивающего потребности передачи, обработки и хранения информации оборудования (даже не имеющего реальных аналогов) в настоящее время;
- оценка необходимого запаса производительности оборудования, обеспечивающего возможное увеличение производственных потребностей в ближайшее время (один-два года);
- выбор нескольких вариантов оборудования с учетом текущих потребностей, перспективы развития на основании критерия стоимости оборудования;
- проведение проверки работы вычислительной системы, составленной из рекомендованного оборудования.

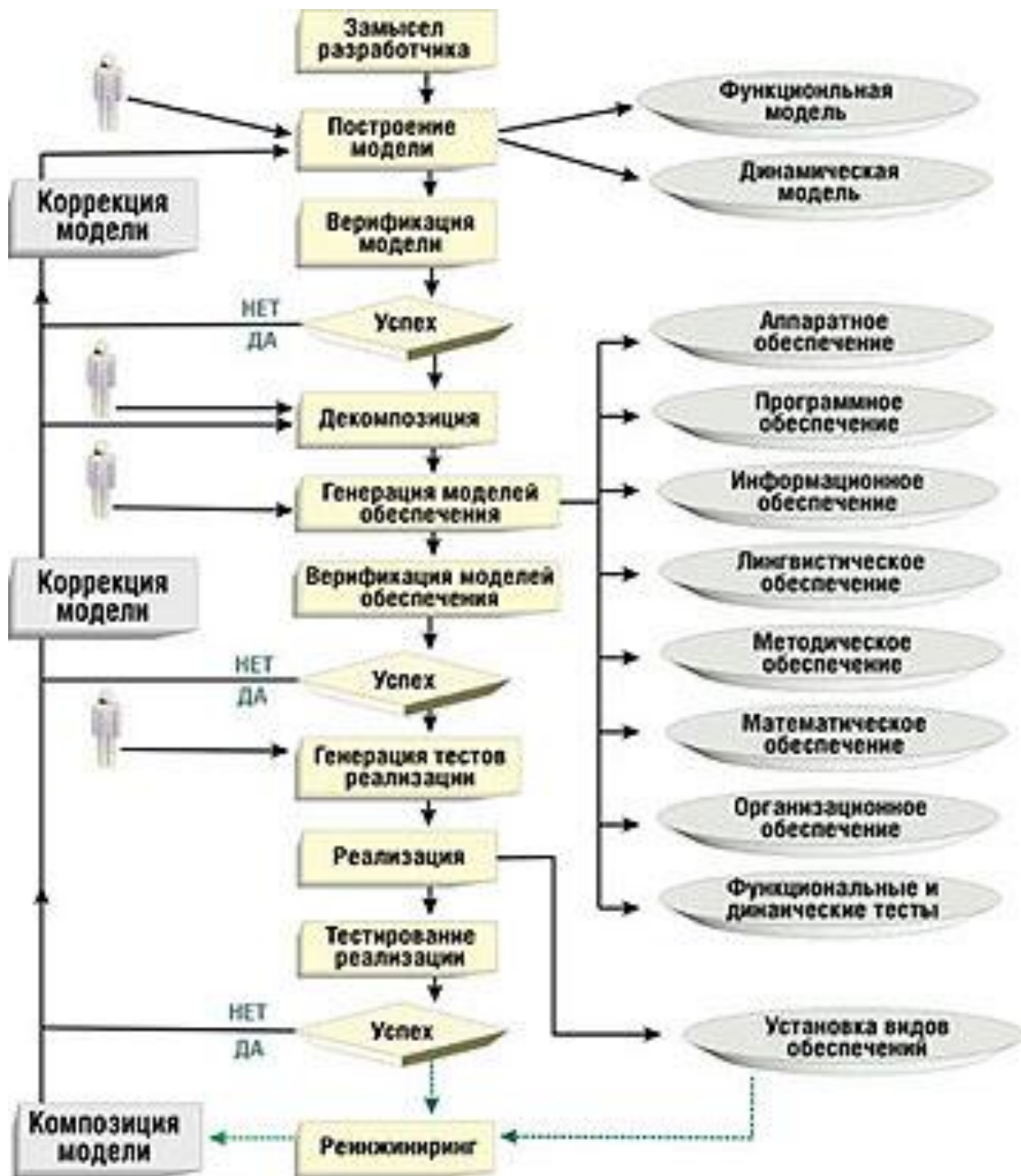


Рис.3. 1. Процесс бездефектного проектирования вычислительной системы

3.1.Задачи моделирования ИС.

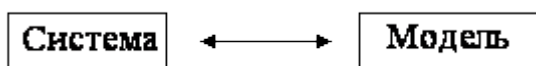
Модель – это объект-«заменитель» объекта-«оригинала», который находится в определенном соответствии с оригиналом и обеспечивает представление о некоторых его свойствах. Модель системы представляет собой абстрактное описание на некотором формальном языке характеристик

системы, важных с точки зрения цели моделирования, ее поведения. При создании системы нельзя ограничиваться созданием только одной модели. Если система сложная, то учет всех ее характеристик в одной модели приведет к чрезвычайной ее сложности. Наилучший подход при разработке любой нетривиальной системы – использовать совокупность нескольких моделей, которые могут быть практически независимыми друг от друга и позволят сделать акценты на разных сторонах системы при решении различных задач поддержания ее жизненного цикла.

В общем случае модели можно разделить на следующие виды: статические, описывающие структурные свойства систем; динамические, представляющие поведенческие свойства систем; функциональные, описывающие функциональные свойства систем. Статическая модель описывает составные части системы, их структуру, атрибуты, взаимосвязи между ними и операции, которые они могут выполнять. Операции статической модели являются событиями динамической и функциями функциональной моделей. Динамическая модель описывает последовательность выполнения операций в процессе функционирования системы. Функциональная модель описывает преобразования, осуществляемые системой. Она раскрывает содержание операций статической модели и событий динамической.

По степени абстракции модели можно разделить на концептуальные модели, представляющие высокоуровневый взгляд на задачу в терминах предметной области; модели спецификации, определяющие «внешний вид» и внешнее поведение системы; модели реализации, которые отражают внутреннее устройство системы, конкретный способ реализации наблюдаемого поведения системы.

Моделированием называется замещение одного объекта, называемого системой, другим объектом, называемым моделью, и проведение экспериментов с моделью (или на модели), исследование свойств модели, опираясь на результаты экспериментов с целью получения информации о системе



Моделирование позволяет исследовать такие системы, прямой эксперимент с которыми:

- а) трудно выполним;

б) экономически невыгоден;

в) вообще невозможен.

Суть компьютерного моделирования заключена в получении количественных и качественных результатов на имеющейся модели. Качественные результаты анализа обнаруживают неизвестные ранее свойства информационной системы: ее структуру, динамику развития, устойчивость, целостность и др. Количественные выводы в основном носят характер анализа существующей ИС или прогноза будущих значений некоторых переменных. Возможность получения не только качественных, но и количественных результатов составляет существенное отличие имитационного моделирования от структурно-функционального. Имитационное моделирование имеет целый ряд специфических черт.

Модель должна строиться целенаправленно. Целенаправленная модель представляет собой замену действительности с той степенью абстракции, которая необходима для поставленной цели. То есть, модель, прежде всего, должна отражать те существенные свойства и те стороны моделируемого объекта, которые определены задачей. При этом важно правильно обозначить и сформулировать проблему, четко определить цель исследования, проводимого с помощью моделирования.

Требования к моделям. Моделирование связано с решением реальных задач и необходимо быть уверенным, что результаты моделирования с достаточной степенью точности отражают истинное положение вещей, т.е. модель адекватна реальной действительности.

Хорошая модель должна удовлетворять некоторым общепринятым требованиям. Такая модель должна быть:

- адекватной;
- надежной;
- простой и понятной пользователю;
- целенаправленной;
- удобной в управлении и обращении;
- функционально полной с точки зрения возможностей решения главных задач;
- адаптивной, позволяющей легко переходить к другим модификациям или обновлять данные;
- допускающей изменения (в процессе эксплуатации она может усложняться).

В зависимости от целевой направленности модели, для нее задаются специальные требования. Наиболее характерными являются: целостность, отражение информационных свойств, многоуровневость, множественность (многомодельность), расширяемость, универсальность, осуществимость (реальная возможность построения самой модели и ее исследования), реализуемость (например, на ЭВМ, возможность материализации модели в

виде реальной системы в задачах проектирования), эффективность (затраты временных, трудовых, материальных и других видов ресурсов на построение моделей и проведение экспериментов находятся в допустимых пределах или оправданы). Значимость или приоритетность требований к модели непосредственно вытекают из назначения модели. Например, в исследовательских задачах, задачах управления, планирования и описания важным требованием является адекватность модели объективной реальности. В задачах проектирования и синтеза уникальных систем важным требованием является реализуемость модели, например в САПР или систему поддержки принятия решений (СППР).

Цель моделирования и задание требований к модели определяют форму представления модели.

Любая модель (прежде чем стать объективно существующим предметом) должна существовать в мысленной форме, быть конструктивно разработанной, переведена в знаковую форму и материализована.

Моделирование - важнейшая сфера применения средств вычислительной техники, когда положения теории моделирования используются в различных областях науки, производства и техники. В то же время сами средства вычислительной техники являются объектами моделирования на этапе проектирования новых и модернизации старых вычислительных систем, при анализе возможности использования вычислительных систем в различных приложениях.

При системном подходе к моделированию систем необходимо, прежде всего, четко определить цель моделирования. Поскольку невозможно полностью смоделировать реально функционирующую систему (систему-оригинал или первую систему), создается модель (система-модель или вторая система) под поставленную проблему. Таким образом, применительно к вопросам моделирования цель возникает из требуемых задач моделирования, что позволяет подойти к выбору критерия и оценить, какие элементы войдут в создаваемую модель. Поэтому необходимо иметь критерий отбора отдельных элементов в создаваемую модель.

3.2. Высокоуровневое моделирование ИС.

Высокоуровневые модели это, как правило, метамоделли, то есть, модели включающие в свой состав модели более низкого порядка.

Обычно высокоуровневые модели (моделирования функций или бизнес-процессов) реализуются с использованием языков программирования высокого уровня. Моделирование высокого уровня позволяет решать следующие задачи:

- исследование проектного пространства;
- имитация устройств ввода – вывода;

- интеграция в симулирующую систему стороннего программного обеспечения;
- интеграция в симулирующую систему стороннего аппаратного обеспечения;
- имитация отсутствующих компонентов проекта;
- ускорение симуляции;
- контроль адекватности проекта;
- совместная отладка программного и аппаратного обеспечения.

К основным достоинствам высокоуровневых моделей следует отнести высокую скорость симуляции и относительную простоту создания моделей. К основным проблемам использования высокоуровневых моделей относится потребность в высококвалифицированных специалистах в областях как программного, так и аппаратного обеспечения.

Высокоуровневое моделирование резко ускоряет выработку решений по требованиям, архитектуре и прочим высокоуровневым аспектам системы ровно в той мере, в какой модель коллективно разрабатывается и служит тем артефактом, который обслуживает работу команды.

Для этого средства моделирования необходимо поддерживать коллективную разработку, то есть:

- подразумевать работу многих людей за одним экраном (для этого экран должен быть большой, шрифты и иконки регулируемого размера -- ведь чем больше людей обсуждают модель, тем дальше они от экрана, и это при том, что хочется уместить на экран как можно больший кусок модели);
- обладать достаточно высокой степенью модульности ;
- поддерживать традиционные для программирования функции интеграции и верификации (тестирование);
- управление конфигурацией и изменениями;
- реализовать функции подсказки в части методологии разработки ;
- модель должна быть также привязана к организации (полномочиям, людям, их компетенциям чтобы к моделированию подключались правильные люди).

Одна из трудностей в высокоуровневом моделировании сегодня - это возможность накопления знаний. По факту, практически все проекты моделирования (и онтологизирования) сегодня приходится начинать практически с нуля так как нет библиотек (библиотеки Modelica тут не в счёт, это исключение, причём они больше не для высокоуровневого моделирования, а для низкоуровневой мультифизики), нет доступных архитектурных решений (приходится понимать их общие черты, а затем рисовать в том или ином моделере в каждом новом проекте), паттернов, онтологий.

Накопление знаний (т.е. повторно используемой информации) многогранно, требует самых разных решений: от поддержки модульности,

развития языка моделирования до возможности прийти на новое место работы, рассчитывая на возможность использования библиотек моделирования, освоенных на старом месте работы.

Моделирование (и онтологизирование) на сегодня в этом плане похоже на программирование, каким оно было в середине семидесятых: про удобство библиотек все знали, но библиотек в мире было крайне мало (разве что парочка фортранных библиотек с численными методами).

Поэтому начальные высокие затраты остаются высокими (каждый раз приходится один и тот же длинный путь, с нуля), а последующие затраты не снижаются.

3.3. Низкоуровневое моделирование средств реализации ИС.

В процессе проектирования систем, состоящих из множества устройств, важная роль отводится этапу моделирования системы.

При разработке моделей обычно пользуются одной из следующих концепций:

- Использование исходных описаний устройств на языке моделирования устройств (аппаратуры). Как следствие, моделирование проводится с использованием средств соответствующего языка;
- Создание абстрактных описаний устройств на произвольном языке высокого уровня.

Моделирование с использованием первой концепции называют в противоположность второму - абстрактному моделированию или высокоуровневым моделированием .

К достоинствам низкоуровневого моделирования относятся:

- Адекватность поведения моделей при моделировании реальной работы устройств (при моделировании используются те же описания, что и при описании работы устройств).
- Простота диагностики ошибок, обнаруженных в работе устройств (средства моделирования языков описания аппаратуры предоставляют полный доступ к внутренней структуре устройств).

С другой стороны очевидны следующие недостатки метода:

- Низкая скорость моделирования (в силу особенностей языков моделирования аппаратуры, которые сильно проигрывают в скорости универсальным языкам).
- Узкая специализация языков моделирования аппаратуры (языки моделирования аппаратуры не предоставляют достаточно средств для разработки программ моделирования).

Преимущества низкоуровневой модели позволяют эффективно решать большинство задач моделирования, однако главный недостаток – скорость работы заставляет использовать абстрактные методы при решении задач, требующих большего объема тестов (сбор статистики, стресс-тесты). В свою

очередь, отказаться от низкоуровневого моделирования невозможно в силу того, что абстрактная модель неприменима при отладке исходного кода устройств и дает низкую точность оценкам производительности систем. В большинстве случаев разработчики создают две модели, каждую из которых используют для своего круга задач. При помощи низкоуровневой модели проводят отладку устройств. При помощи абстрактной – верификацию протоколов и оценку эффективности алгоритмов. Оставшиеся задачи решаются в соответствии со спецификой конкретной разработки и предпочтениями разработчиков при помощи того или иного метода.

Повысив скорость работы низкоуровневой модели можно в некоторых случаях отказаться от написания абстрактного варианта, решая стоящие перед моделированием задачи лишь при помощи низкоуровневой модели.

Основным достоинством низкоуровневого моделирования является то, что, используя исходные описания устройств, мы решаем проблему их адекватного моделирования. При этом имеется полный доступ к внутренним интерфейсам устройств, что крайне важно при отладке.

Однако использование исходных описаний устройств заставляет использовать системы моделирования тех языков, на которых были описаны сами устройства. Данные системы, в силу своих особенностей, обладают крайне низкой производительностью и не являются универсальными (не предназначены для решения широкого круга задач).

Для решения проблемы универсальности разработчики языков моделирования аппаратуры предоставляют пользователям возможность дополнять наборы системных функций языка своими, написанными на языке высокого уровня (обычно C или C++), предлагая для этого соответствующие библиотеки.

В модели кроме тестируемой части имеется множество обслуживающих или сервисных устройств. К ним относятся генераторы нагрузки, мониторы, имитаторы периферии и т.д. В зависимости от уровня сложности системы, доля сервисной части модели может достигать 50%. В отличие от тестируемых устройств, возможность доступа к внутренним интерфейсам сервисных устройств не является необходимой, и вследствие этого использование языков описания аппаратуры при разработке этой части модели не обязательно.

Использование языков высокого уровня для описания серверной части модели может дать серьезный прирост производительности модели. Оно не только увеличит, в силу очевидных причин, скорость работы самой серверной части, но и может ускорить работу тестируемой части модели, так как производительность языков моделирования аппаратуры падает не линейно с ростом объема моделируемых устройств. Кроме того, действуя таким образом, можно избавимся от одного из недостатков низкоуровневого моделирования – отсутствия гибкости, так как при описании серверной части модели на языке высокого уровня мы сможем использовать все их преимущества, а именно, гибкость, высокую скорость и обширные библиотеки.

Сервисное устройство, описанное на языке высокого уровня, тем не менее, должно взаимодействовать с тестируемыми устройствами, написанными на языке описания аппаратуры. Подобное взаимодействие проще всего организовать, добавив к тестируемой части модели интерфейсные модули для каждого из сервисных устройств.

Обмен данными между устройством и его интерфейсным модулем можно организовать с использованием языка моделирования аппаратуры

3.4. Имитационное моделирование ИС.

По Р. Шеннону (Robert E. Shannon — профессор университета в Хантсвилле, штат Алабама, США) "имитационное моделирование — есть процесс конструирования на ЭВМ модели сложной реальной системы, функционирующей во времени, и постановки экспериментов на этой модели с целью либо понять поведение системы, либо оценить различные стратегии, обеспечивающие функционирование данной системы".

Выделим в этом определении ряд важнейших обстоятельств, учитывая особенности применения метода для исследования информационных систем (ИС).

Во-первых, имитационное моделирование предполагает два этапа: конструирование модели на ЭВМ и проведение экспериментов с этой моделью. Каждый из этих этапов предусматривает использование собственных методов. Так, на первом этапе весьма важно грамотно провести информационное обследование, разработку всех видов документации и их реализацию. Второй этап должен предполагать использование методов планирования эксперимента с учетом особенностей машинной имитации.

Во-вторых, в полном соответствии с системными принципами четко выделены две возможные цели имитационных экспериментов:

- либо понять поведение исследуемой системы (о которой по каким-либо причинам было "мало" информации) — потребность в этом часто возникает, например, при создании принципиально новых образцов продукции;
- либо оценить возможные стратегии управления системой, что также очень характерно для решения широкого круга экономико-прикладных задач.

В-третьих, с помощью имитационного моделирования исследуют сложные системы. Понятие "сложность" является субъективным и по сути выражает отношение исследователя к объекту моделирования. Укажем пять признаков "сложности" системы, по которым можно судить о ее принадлежности к такому классу систем:

- наличие большого количества взаимосвязанных и взаимодействующих элементов;
- сложность функции (функций), выполняемой системой;

- возможность разбиения системы на подсистемы (декомпозиции);
- наличие управления (часто имеющего иерархическую структуру), разветвленной информационной сети и интенсивных потоков информации;
- наличие взаимодействия с внешней средой и функционирование в условиях воздействия случайных (неопределенных) факторов.

Очевидно, что некоторые приведенные признаки сами предполагают субъективные суждения. Вместе с тем становится понятным, почему значительное число ИС относят к сложным системам и, следовательно, применяют метод имитационного моделирования.

В-четвертых, методом имитационного моделирования исследуют системы, функционирующие во времени, что определяет необходимость создания и использования специальных методов (механизмов) управления системным временем.

Наконец, в-пятых, в определении прямо указывается на необходимость использования ЭВМ для реализации имитационных моделей, т.е. проведения машинного эксперимента (машинной имитации), причем в подавляющем большинстве случаев применяются цифровые машины.

Даже столь краткий анализ позволяет сформулировать вывод о целесообразности (а, следовательно, и необходимости) использования метода имитационного моделирования для исследования сложных человекомашинных (эргатических) информационных систем. Особо выделим наиболее характерные обстоятельства применения имитационных моделей:

- если идет процесс познания объекта моделирования;
- если аналитические методы исследования имеются, но составляющие их математические процедуры очень сложны и трудоемки;
- если необходимо осуществить наблюдение за поведением компонент системы в течение определенного времени;
- если необходимо контролировать протекание процессов в системе путем замедления или ускорения явлений в ходе имитации;
- если особое значение имеет последовательность событий в проектируемых системах и модель используется для предсказания так называемых "узких" мест;
- при подготовке специалистов для приобретения необходимых навыков в эксплуатации новой техники;
- и, конечно, если имитационное моделирование оказывается единственным способом исследований из-за невозможности проведения реальных экспериментов.

До настоящего момента особое внимание в толковании термина "имитационное моделирование системы" было уделено первому слову. Однако не следует упускать из вида, что создание любой (в том числе и имитационной) модели предполагает, что она будет отражать лишь наиболее

существенные с точки зрения конкретной решаемой задачи свойства объекта-оригинала.

Английский аналог этого термина — *systems simulation* — при дословном переводе непосредственно указывает на необходимость воспроизводства (симуляции) лишь основных черт реального явления (сравним с термином "симуляция симптомов болезни" из медицинской практики). Важно отметить еще один аспект: создание любой (в том числе и имитационной модели) есть процесс творческий (не случайно Р. Шеннон назвал свою книгу "Имитационное моделирование систем — искусство и наука"), и, вообще, каждый автор имеет право на собственную версию модели реальной системы. Однако за достаточно длительное время применения метода накоплены определенный опыт и признанные разумными рекомендации, которыми целесообразно руководствоваться при организации имитационных экспериментов.

Укажем ряд основных достоинств и недостатков метода имитационного моделирования. Основные достоинства:

- имитационная модель позволяет, в принципе, описать моделируемый процесс с большей адекватностью, чем другие;
- имитационная модель обладает гибкостью варьирования структуры, алгоритмов и параметров системы;
- применение ЭВМ существенно сокращает продолжительность испытаний по сравнению с натурным экспериментом (если он возможен), а также их стоимость.

Основные недостатки:

- решение, полученное на имитационной модели, всегда носит частный характер, так как оно соответствует фиксированным элементам структуры, алгоритмам поведения и значениям параметров системы;
- большие трудозатраты на создание модели и проведение экспериментов, а также обработку их результатов;
- если использование системы предполагает участие людей при проведении машинного эксперимента, на результаты может оказать влияние так называемый хауторнский эффект (закрывающийся в том, что люди, зная (чувствуя), что за ними наблюдают, могут изменить свое обычное поведение).

Итак, само использование термина "имитационное моделирование" предполагает работу с такими математическими моделями, с помощью которых результат исследуемой операции нельзя заранее вычислить или предсказать, поэтому необходим эксперимент (имитация) на модели при заданных исходных данных. В свою очередь, сущность машинной имитации заключается в реализации численного метода проведения на ЭВМ экспериментов с математическими моделями, описывающими поведение сложной системы в течение заданного или формируемого периода времени.

Каждая имитационная модель представляет собой комбинацию шести основных составляющих:

- компонентов;
- переменных;
- параметров;
- функциональных зависимостей;
- ограничений;
- целевых функций.

Под компонентами понимают составные части, которые при соответствующем объединении образуют систему. Компоненты называют также элементами системы или ее подсистемами. Например, в модели рынка ценных бумаг компонентами могут выступать отделы коммерческого банка (кредитный, операционный и т.д.), ценные бумаги и их виды, доходы, котировка и т.п.

Параметры — это величины, которые исследователь (пользователь модели) может выбирать произвольно, т.е. управлять ими.

В отличие от них переменные могут принимать только значения, определяемые видом данной функции.

Функциональные зависимости описывают поведение параметров и переменных в пределах компонента или же выражают соотношения между компонентами системы. Эти соотношения могут быть либо детерминированными, либо стохастическими.

Ограничения — устанавливаемые пределы изменения значений переменных или ограничивающие условия их изменения. Они могут вводиться разработчиком (и тогда их называют искусственными) или определяться самой системой вследствие присущих ей свойств (так называемые естественные ограничения).

Целевая функция предназначена для измерения степени достижения системой желаемой (требуемой) цели и вынесения оценочного суждения по результатам моделирования. Эту функцию также называют функцией критерия. По сути, весь машинный эксперимент с имитационной моделью заключается в поиске таких стратегий управления системой, которые удовлетворяли бы одной из трех концепций ее рационального поведения: оптимизации, пригодности или адаптивизации. Если показатель эффективности системы является скалярным, проблем с формированием критерия не возникает и, как правило, решается оптимизационная задача — поиска стратегии, соответствующей максимуму или минимуму показателя. Сложнее дело обстоит, если приходится использовать векторный показатель. В этом случае для вынесения оценочного суждения используются методы принятия решений по векторному показателю в условиях определенности (когда в модели учитываются только детерминированные факторы) или неопределенности (в противном случае).

При реализации имитационной модели, как правило, рассматриваются не все реально осуществляемые функциональные действия (ФД) системы, а только те из них, которые являются наиболее существенными для исследуемой операции. Кроме того, реальные ФД аппроксимируются упрощенными действиями ФД, причем степень этих упрощений определяется уровнем детализации учитываемых в модели факторов. Названные обстоятельства порождают ошибки имитации процесса функционирования реальной системы, что, в свою очередь, обуславливает адекватность модели объекту-оригиналу и достоверность получаемых в ходе моделирования результатов.

3.5. Анализ и оценка производительности ИС.

Целями моделирования информационных систем чаще всего являются оценка их производительности и надежности.

Производительность и надежность информационных систем связаны с временными аспектами функционирования и зависят в первую очередь от параметров аппаратно–программных средств, которые обычно называются вычислительной системой.

При оценке производительности первостепенное значение имеет интенсивность реализации информационных процессов.

При оценке надежности исследуется продолжительность пребывания системы в различных состояниях, которые меняются из-за отказов в устройствах и программах и последующего восстановления работоспособности.

Для вычислительных систем типично наличие случайных факторов, влияющих на характер протекания процессов. Продолжительность процессорной обработки, число, порядок и параметры обращений к периферийным устройствам зависят от исходных данных, которые порождаются вне системы и носят для нее случайный характер. Случайными являются потоки отказов и времена восстановления отказавших элементов. В связи с этим при оценке функционирования вычислительных систем используется вероятностный подход. Этот подход предполагает, что на процессы воздействуют случайные факторы и свойства процессов и системы в целом проявляется статистически на множестве их реализаций.

Процессы, происходящие в вычислительных системах, представляются в моделях как непрерывные или дискретные случайные процессы. При исследовании вычислительных систем чаще всего приходится иметь дело с дискретными случайными процессами, определенными на конечном множестве состояний, причем процессы рассматриваются или в непрерывном, или в дискретном времени.

Вероятностный подход к описанию функционирования вычислительных систем приводит к использованию аппарата теории вероятностей и математической статистики в качестве математической базы методов исследования.

Случайные величины, соответствующие параметрам элементов моделей, могут представляться на разных уровнях, среди которых наиболее широко используются четыре:

- 1) статистическая выборка $a_1 \dots a_n$, определяющая случайную величину набором значений;
- 2) закон распределения случайной величины с его параметрами;
- 3) математическое ожидание и дисперсия;
- 4) математическое ожидание.

Построим аналитическую модель для информационной системы на базе локальной сети типа "файл-сервер". В основу положим отождествление функционирования системы с простым непрерывным марковским процессом.

Процесс взаимодействия пользователей и системы складывается из отдельных повторяющихся этапов, каждый из которых делится на системную и пультовую фазы.

Запрос пользователя находится в системной фазе, если в системе имеется вся необходимая информация для обработки запрошенной программы. В этой фазе производится обработка запрошенной программы, при этом время сервера делится квантами между всеми пользовательскими программами, находящимися в системной фазе.

Если для дальнейшего выполнения программы необходим ввод с пульта новой входной информации, взаимодействие переходит в пультовую фазу. В пультовой фазе пользователь наблюдает за выводом информации на его терминал, обдумывает свою реакцию на выведенную информацию и вводит с пульта новую входную информацию, что следует рассматривать как ввод в систему нового запроса на обслуживание.

Таким образом, процесс взаимодействия пользователя и системы может находиться в одном из двух состояний: либо система имеет программу, которую она должна выполнять для пользователя, а пользователь ждет ответа системы на свой запрос (системная фаза), либо система фактически не имеет программы, которую могла бы дальше выполнять для данного пользователя, и ждет сообщения от пользователя (пультовая фаза). Примем, что случайные величины: время работы пользователя за пультом (т.е. продолжительность пультовой фазы) t_p и продолжительность чистого времени обработки запроса (без учета времени пребывания в очереди к процессору) $t_{об}$ распределены по экспоненциальному закону со средними значениями соответственно $t_{п}$, c_p и $t_{об}$, $c_{р}$. Функции плотностей для случайных величин t_p и $t_{об}$ имеют вид :

$$f_n(t) = \frac{1}{\tau_{n.cp}} e^{-t/\tau_{n.cp}} \quad (t \geq 0)$$

$$f_n(t) = \frac{1}{\tau_{об.ср}} e^{-t/\tau_{об.ср}} \quad (t \geq 0)$$

В системах без совмещения вычислений с обменом информацией с внешним ЗУ время обмена должно включаться в $t_{об}$. Предполагаем, что переключение программ происходит мгновенно и издержки, т.е. потери времени отсутствуют. Модель строится с целью определения основных характеристик системы : среднего числа пользователей, ожидающих ответа, и средней продолжительности ожидания ответа на запрос пользователя. Примем, что состояние системы определяется числом запросов j , находящихся в данный момент в системной фазе. Тогда система с n пользователями имеет $(n + 1)$ состояний $(0 \leq f \leq n)$. Считаем, что вероятность перехода из состояния j в i зависит только от самих состояний j и i и не зависит от предыстории, приведшей систему в состояние j . Сами переходы совершаются только через случайные промежутки времени. При принятых допущениях, в том числе относительно законов распределения для $\tau_{п}$ и $t_{об}$, функционирование системы может быть представлено простым непрерывным (по времени) марковским процессом. Пусть в данный момент времени j запросов находятся в системной фазе, т.е. ожидают обслуживания, а соответственно $(n - j)$ запросов находятся в пультовой фазе. Тогда число запросов, переходящих в системную фазу в единицу времени, или, иначе говоря, интенсивность (скорость) перехода из состояния с j запросами в состояние с $j + 1$ запросами в системной фазе (ожидающими обслуживания) будет :

$$a_{j(j+1)} = \frac{n-j}{\tau_{n.cp}} \quad (0 \leq j \leq n-1)$$

Вместе с тем происходят переходы запросов из системной в пультовую фазу. Если бы каждому из j запросов, находящихся в системной фазе, было выделено все время работы сервера, то интенсивность перехода запросов из системной в пультовую фазу составила бы $j / t_{об.ср}$, ср. Однако на самом деле время сервера делится поровну (квантами) между всеми j пользователями и поэтому интенсивность перехода системы из состояния с j пользователями в состояние с $j-1$ пользователями :

$$a_{j(j+1)} = \frac{1}{t_{об.ср}} \quad (1 \leq j \leq n)$$

Функционирование рассматриваемой системы разделения времени может быть описано цепью Маркова, соответствующей графу состояний системы, представленному на рис. 3.1. На нем указаны интенсивности переходов системы из одних состояний в другие.

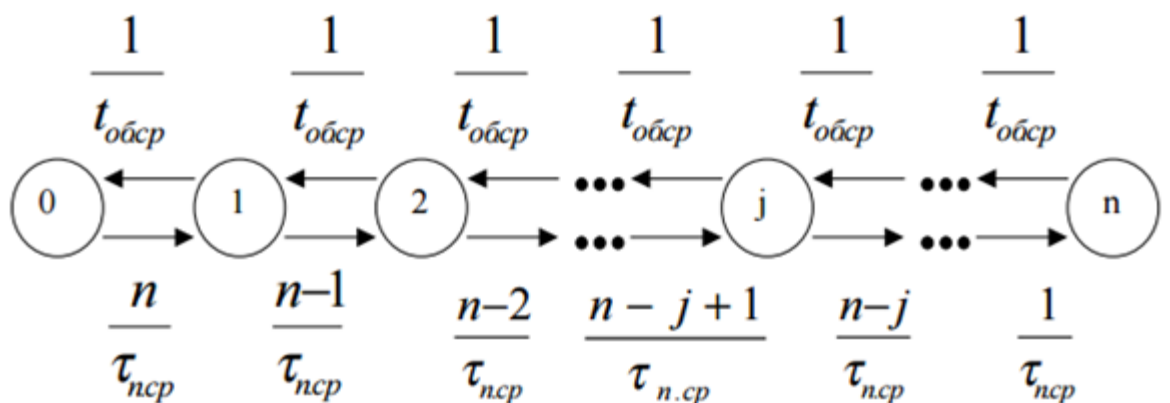


Рис. 3.1. Граф состояний системы

Обозначим через $\Pi = \{\pi_0, \pi_1, \dots, \pi_n\}$ вектор предельных вероятностей состояний системы. Составляющие этого вектора π_j ($0 \leq j \leq n$) есть вероятности пребывания системы в состоянии j в стационарном режиме, причем

$$\sum_{j=0}^n \pi_j = 1$$

Составляющие вектора можно определить из системы уравнений, формируемой по переходным вероятностям на графе состояний системы (см. рис.3.1) :

$$\left. \begin{aligned} & -\frac{n}{\tau_{n.cр}}\pi_0 + \frac{1}{t_{об.ср}}\pi_1 = 0; \\ & \frac{n}{\tau_{n.cр}}\pi_0 - \left(\frac{1}{t_{об.ср}} + \frac{n-1}{\tau_{n.cр}}\right)\pi_1 + \frac{1}{t_{об.ср}}\pi_2 = 0; \\ & \dots \\ & \frac{n-j+1}{\tau_{n.cр}}\pi_{j-1} - \left(\frac{1}{t_{об.ср}} + \frac{n-j}{\tau_{n.cр}}\right)\pi_j + \frac{1}{t_{об.ср}}\pi_{j+1} = 0; \\ & \dots \\ & \frac{1}{\tau_{n.cр}}\pi_{n-1} - \frac{1}{t_{об.ср}}\pi_n = 0. \end{aligned} \right\}$$

Полученные уравнения и получаемые на их основе результаты не зависят от кванта обслуживания, так как было сделано предположение, что нет издержек времени на переключение программ. Обозначив $t_{об.ср}/\tau_{п.ср} = \chi$, получим из вышеприведенной системы :

$$\left. \begin{array}{l} \pi_1 = n\chi\pi_0; \\ \pi_2 = n(n-1)\chi^2\pi_0; \\ \dots\dots\dots \\ \pi_j = \frac{n!}{(n-j)!}\chi^j\pi_0; \\ \dots\dots\dots \\ \pi_n = n!\chi^n\pi_0. \end{array} \right\}$$

Полученные значения представляют собой значения вектора предельных состояний системы, поэтому

$$\left[1 + n\chi + n(n-1)\chi^2 + \dots + \frac{n!}{(n-j)!}\chi^j + \dots + n!\chi^n \right] \pi_0 = 1,$$

Откуда вероятность незанятости процессора в стационарном режиме

$$\pi_0 = \frac{1}{\sum_{j=0}^n \frac{n!}{(n-j)!}\chi^j}$$

Соответственно

$$\pi_i = \frac{\frac{n!}{(n-i)!}\chi^i}{\sum_{j=0}^n \frac{n!}{(n-j)!}\chi^j}$$

Среднее количество запросов, ожидающих ответа :

$$L_{cp} = \sum_{i=0}^n i\pi_i = \frac{\sum_{i=0}^n \frac{in!}{(n-i)!} \chi^i}{\sum_{i=0}^n \frac{n!}{(n-i)!} \chi^i}$$

Подставляя в эту формулу выражения для L_{cp} с учетом выражения π_i после преобразований получаем :

$$L_{cp} = n \frac{t_{омв.ср}}{t_{омв.ср} + \tau_{n.ср}} .$$

Используя выражение для π_0 , полученное выражение можно преобразовать к виду, обеспечивающему получение значения времени ответа на запрос.

$$t_{омв.ср}/t_{обр.ср} = n/(1-\pi_0) - 1/\chi .$$

Список литературы

1. Советов Б.Я., Яковлев С.А. Моделирование систем: Учебник для вузов – 3-е изд., перераб. и доп. –М.: Высшая школа, 2001. –343с.: ил.
2. А. Якобсон, Г.Буч, Дж. Рамбо . Унифицированный процесс разработки программного обеспечения. Изд-во «Питер», 2002
3. Шеннон, Р. Имитационное моделирование систем – искусство и наука / Р. Шеннон. – М., 1978.