

Список сокращений и терминов.

БД — база данных;

ИМ — имитационная модель;

ИС — информационная система;

Введение

Имитационное моделирование (ситуационное моделирование) – метод, позволяющий строить модели, описывающие процессы так, как они проходили бы в действительности. Такую модель можно исполнить как для одного испытания, так и для множества. При этом результаты будут определяться случайным характером процессов. Используя эти данные можно получить в меру устойчивую статистику.

Целью данной курсовой работы является разработка имитационной модели распределенной информационной системы в среде имитационного моделирования AnyLogic.

Для достижения поставленной цели необходимо решить следующие задачи:

- рассмотреть основные понятия имитационного моделирования;
- изучить модели распределенной информационной системы;
- изучить возможности работы среды разработки компьютерных моделей AnyLogic;
- обосновать выбор цели и начальных условий моделирования;
- построить концептуальную модель распределённой информационной системы;
- представить структуру и описание имитационной модели;
- построить имитационную модель распределённой системы;
- провести модельные исследования и анализ их результатов;
- число источников заявок, не менее 15;

- диапазон интенсивностей поступления заявок 30 в сек;
- число устройств обработки заявок, не менее 5.

Особое место среди сред разработки компьютерных моделей сложных систем принадлежит много подходной среде моделирования имитационных моделей – AnyLogic. Разные средства спецификации и анализа результатов, имеющиеся в AnyLogic, позволяют строить модели, имитирующие практически любой реальный процесс. Графическая среда моделирования поддерживает проектирование, разработку, документирование модели, выполнение компьютерных экспериментов, оптимизацию параметров относительно некоторого критерия.

Глава 1. Теоретические основы моделирования

1.1 Понятие имитационного моделирования.

Имитационное моделирование – это разработка и выполнение на компьютере программной системы, отражающей структуру и функционирование (поведение) моделируемого объекта или явления во времени. Такую программную систему называют имитационной моделью этого объекта или явления. Объекты и сущности имитационной модели представляют объекты и сущности реального мира, а связи структурных единиц объекта моделирования отражаются в интерфейсных связях соответствующих объектов модели. Таким образом, имитационная модель – это упрощенное подобие реальной системы, либо существующей, либо той, которую предполагается создать в будущем. Имитационная модель является компьютерной программой, выполнение которой можно считать имитацией поведения исходной системы во времени.

Имитационное моделирование имеет существенные преимущества перед аналитическим моделированием в тех случаях, когда:

- отношения между переменными в модели нелинейны, и поэтому аналитические модели трудно или невозможно построить;
- модель содержит стохастические компоненты;
- для понимания поведения системы требуется визуализация динамики происходящих в ней процессов;
- модель содержит много параллельно функционирующих взаимодействующих компонентов.

Во многих случаях имитационное моделирование – это единственный способ получить представление о поведении сложной системы и провести ее анализ. Конечной целью моделирования является оптимизация тех или иных

параметров системы. Однако, потенциальные возможности имитационного моделирования существенно шире. В зависимости от этапа и назначения проводимых исследований применяется один из трех наиболее распространенных видов имитационных экспериментов:

- исследование относительного влияния различных факторов на значения выходных характеристик системы;
- нахождение аналитической зависимости между интересующими исследователя выходными характеристиками и факторами;
- отыскание оптимальных значений параметров системы (так называемый "экстремальный эксперимент").

Вид эксперимента влияет не только на выбор схемы ее формализации, но также на построение плана эксперимента и выбор метода обработки его результатов.

С точки зрения организации взаимодействия исследователя с моделью в ходе эксперимента ИМ делятся на автоматические и диалоговые.

Автоматическими называются ИМ, взаимодействие пользователя с которыми сводится только к вводу исходной информации и управлению началом и окончанием работы моделей. Диалоговыми называются ИМ, позволяющие исследователю активно управлять ходом моделирования.

1.2. Этапы имитационного моделирования

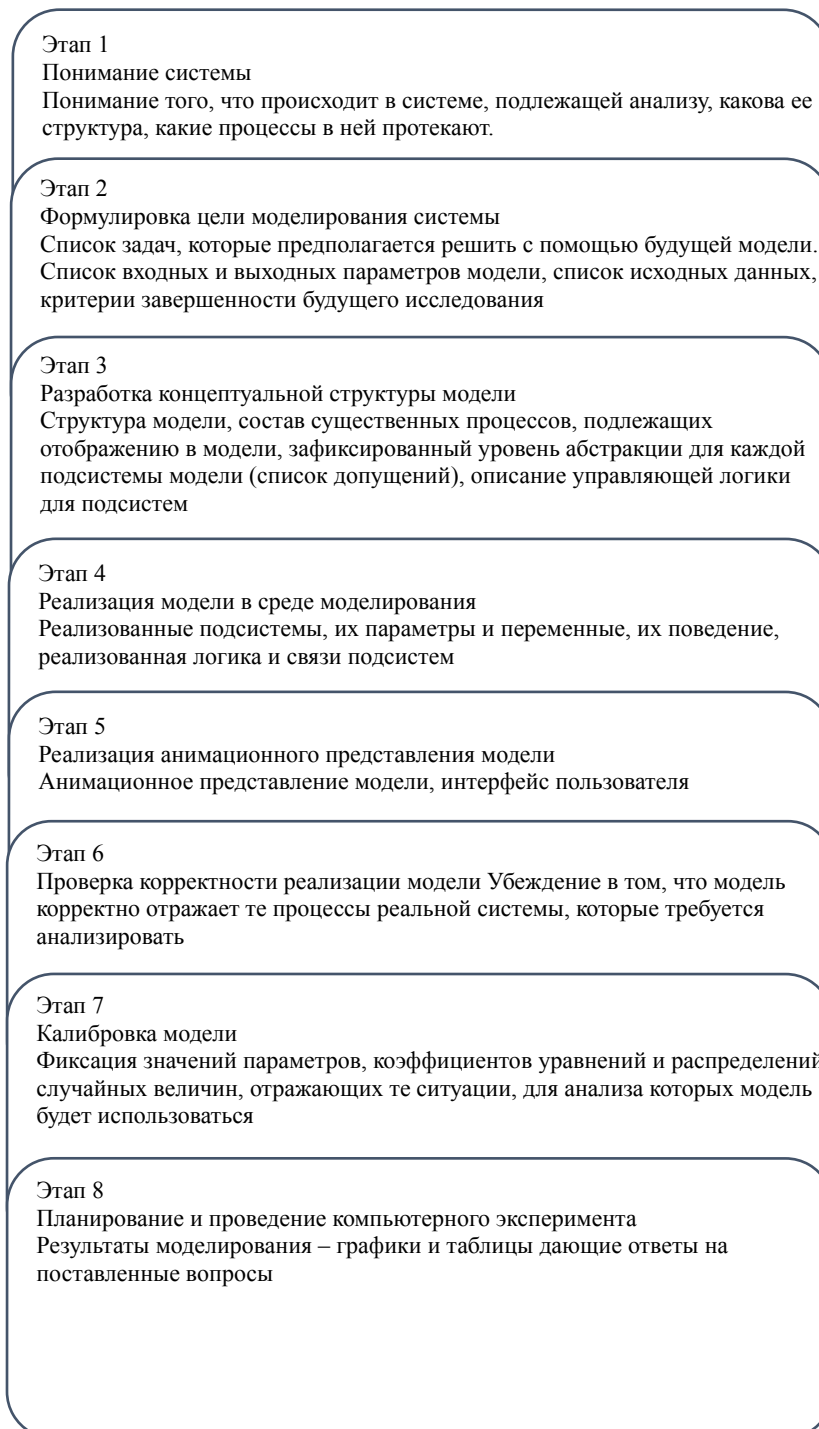
Имитационное моделирование состоит из двух этапов: создания модели и анализа полученной модели с целью принятия решения. Сначала разработчик должен определить, какие задачи будут решаться с ее помощью, из этого следует, что моделированию в любой его форме должна предшествовать формулировка цели моделирования. От цели зависит то, какие процессы в реальной системе следует отразить и выделить в модели, а от каких процессов следует отказаться, какие характеристики этих процессов следует

учитывать, а какие не следует, какие соотношения между переменными и параметрами модели должны быть отражены в модели. Далее следует этап, который можно охарактеризовать как создание концептуальной модели. В нем выполняется структуризация модели, выделение отдельных подсистем, определение элементарных компонентов модели и их связей на каждом уровне иерархии. В имитационном моделировании структура модели отражает структуру реального объекта моделирования на некотором уровне абстракции, а связи между компонентами модели являются отражением реальных связей. Элементы системы, их связи, параметры и переменные, а также их соотношения и законы их изменения должны быть выражены средствами среды моделирования, т. е. в этой среде должны быть определены переменные и параметры модели, построены процедуры вычисления изменения переменных и характеристик модели во времени. При необходимости для большего понимания процессов, протекающих в модели, должно быть разработано анимационное представление этих процессов. Затем построенная модель должна быть проверена с точки зрения корректности ее реализации. Следующий этап – это калибровка или идентификация модели, т. е. сбор данных и проведение измерений тех характеристик в реальной системе, которые должны быть введены в модель в виде значений параметров и распределений случайных величин. Далее, необходимо выполнить проверку правильности модели которая состоит в том, что выход модели проверяется на нескольких тестовых режимах, в которых характеристики поведения реальной системы известны либо очевидны. Последним этапом работы с моделью является компьютерный эксперимент, т.е. собственно то, ради чего и создавалась модель. В простейшем случае компьютерный эксперимент – это запуск на исполнение модели при различных значениях ее существенных параметров и наблюдение ее поведения с регистрацией характеристик поведения. Этот вид использования модели называется прогнозом, или экспериментом типа. Компьютерное моделирование позволяет не только получить прогноз, но и

определить, какие управляющие воздействия на систему приведут к благоприятному развитию событий.

На рисунке 1.1 перечислены этапы компьютерного имитационного моделирования.

Рисунок 1. Этапы компьютерного моделирования.



1.3 Преимущества и сферы применения систем имитационного моделирования.

По сравнению с выполнением экспериментов над реальной системой, использование имитационных моделей предлагает ряд преимуществ.

- Использование имитационной модели помогает принять рациональное решение при минимальных затратах.
- Имитационная модель позволяет экономить время при изменениях в сравнении с проведением экспериментов над реальной системой.
- При помощи имитационной модели можно провести неограниченное количество экспериментов, с целью выявления наилучшего результата.
- Имитационное моделирование способно разрешить множество задач из разных областей.

Имитационное моделирование возможно применять в разного рода сферах деятельности. Далее представлен список задач, при решении которых использование моделирования наиболее эффективно:

- проектирование и анализ производственных систем;
- анализ финансовых и экономических систем;
- модернизация различных процессов в деловой сфере;
- определение требований к оборудованию и программному обеспечению различных компьютерных систем;
- оценка проектов создания различных организаций массового обслуживания, например заведений быстрого питания, больниц, отделений связи, сетевых магазинов;
- анализ и проектирование работы транспортных систем, например: аэропортов, вокзалов, портов и метрополитена;

1.4 Среда имитационного моделирования AnyLogic.

AnyLogic – программное обеспечение для имитационного моделирования, разработанное российской компанией The AnyLogic Company (бывшая "Экс Джей Текнолоджис", англ. XJ Technologies). Инструмент обладает современным графическим интерфейсом и позволяет использовать язык Java для разработки моделей.

Среда моделирования:

- Stock & Flow Diagrams (диаграмма потоков и накопителей) применяется при разработке моделей, используя метод системной динамики.

- Statecharts (карты состояний) в основном используется в агентных моделях для определения поведения агентов. Но также часто используется в дискретно-событийном моделировании, например для симуляции машинных сбоев.

- Action charts (блок-схемы) используется для построения алгоритмов. Применяется в дискретно-событийном моделировании (маршрутизация звонков) и агентном моделировании (для логики решений агента).

- Process flowcharts (процессные диаграммы) основная конструкция, используемая для определения процессов в дискретно-событийном моделировании.

Среда моделирования также включает в себя: низкоуровневые конструкции моделирования (переменные, уравнения, параметры, события и т.п), формы представления (линии, квадраты, овалы и т.п), элементы анализа (базы данных, гистограммы, графики), стандартные картинки и формы экспериментов.

Среда моделирования AnyLogic поддерживает проектирование, разработку, документирование модели, выполнение компьютерных экспериментов с моделью, включая различные виды анализа – от анализа чувствительности до оптимизации параметров модели относительно некоторого критерия.

Библиотеки AnyLogic:

- Process Modeling Library разработана для поддержки дискретно-событийного моделирования в таких областях как Производство, Цепи поставок, Логистика и Здравоохранение. Используя Process Modeling Library, вы можете смоделировать системы реального мира с точки зрения заявок (англ. entity) (сделок, клиентов, продуктов, транспортных средств, и т. д.), процессов (последовательности операций, очередей, задержек), и ресурсов. Процессы определены в форме блочной диаграммы.

- Pedestrian Library создана для моделирования пешеходных потоков в "физической" окружающей среде. Это позволяет Вам создавать модели с большим количеством пешеходного трафика (как станции метро, проверки безопасности, улицы и т. д.). Модели поддерживают учёт статистики плотности движения в различных областях. Это гарантирует приемлемую работу пунктов обслуживания с ограничениями по загруженности, оценивает длину простаивания в определённых областях, и обнаруживает потенциальные проблемы с внутренней геометрией – такие как эффект добавления слишком большого числа препятствий – и другими явлениями. В моделях, созданных с помощью Pedestrian Library, пешеходы двигаются непрерывно, реагируя на различные виды препятствий (стены, различные виды областей) так же, как и обычные пешеходы. Пешеходы моделируются как взаимодействующие агенты со сложным поведением. Для быстрого описания потоков пешеходов Pedestrian Library обеспечивает высокоуровневый интерфейс в виде блочной диаграммы.

- Rail Yard Library поддерживает моделирование, имитацию и визуализацию операций сортировочной станции любой сложности и масштаба. Модели сортировочной станции могут использовать комбинированные методы моделирования (дискретно-событийное и агентное моделирование), связанные с действиями при транспортировке: погрузками и разгрузками, распределением ресурсов, обслуживанием, различными бизнес-процессами.

1.5 Общая характеристика распределенных информационных систем.

Распределенной системой называют такую систему, в которой функционирует более одного сервера БД. За частую это применяется для уменьшения нагрузки на сервер и обеспечения работы территориально удаленных подразделений. Различная сложность разработки, усовершенствования, сопровождения, интеграции с другими системами допускают разделение ИС на классы малых, средних и крупных распределенных систем. Малые ИС имеют небольшой жизненный цикл (ЖЦ), направленность на массовое использование, невысокую цену, невозможность модификации без участия разработчиков, использующие в основном настольные системы управления базами данных (СУБД), однородное аппаратно-программное обеспечение, не имеющие средств обеспечения безопасности. Крупные корпоративные ИС, системы федерального уровня и другие имеют длительный жизненный цикл, миграцию унаследованных систем, разнообразие аппаратно-программного обеспечения, масштабность и сложность решаемых задач, пересечение множества предметных областей, аналитическую обработку данных, территориальную распределенность компонент.

К функциям таких ИС следует отнести, прежде всего, работу с распределенными данными, расположенными на разных физических серверах, различных аппаратно-программных платформах и хранящихся в различных внутренних форматах. В этом случае система должна предоставлять полную информацию о себе и всех своих ресурсах, легко расширяться, быть основана на открытых стандартах и протоколах, обеспечивать возможность интегрировать свои ресурсы с ресурсами других ИС. Для пользователей система должна обеспечивать различные уровни привилегий для пользователей и предоставлять простые интерфейсы доступа к информации.

Данные из разнородных систем обычно объединяются в логические группы, к которой и адресуются запросы. Абстрактная система запросов

предполагает, что система оперирует не конкретным синтаксисом запросов, а его логической сутью на основе абстрактных атрибутов.

При построении распределенных ИС, как правило, используются две базовые архитектуры: Клиент/сервер и Internet Intranet.

Корпоративные ИС, построенные по архитектуре Клиент/сервер, предоставляют клиентам широкий спектр приложений и инструментов разработки, которые ориентированы на максимальное использование вычислительных возможностей клиентских рабочих мест. Ресурсы сервера используются в основном для хранения и обмена документами, а также для выхода во внешнюю среду. Данная архитектура позволяет лучше защитить серверную часть приложений, при этом, предоставляя возможность приложениям либо непосредственно адресоваться к другим серверным приложениям, либо маршрутизировать запросы к ним. Однако, частые обращения клиента к серверу снижают производительность работы сети. Приходится решать вопросы безопасной работы в сети, так как приложения и данные распределены между различными клиентами. Распределенный характер построения системы обуславливает сложность ее настройки и сопровождения

В основе ИС на базе Internet Intranet лежит принцип "открытой архитектуры". ПО ИС реализуется в виде апплетов или сервлетов (программ на языке JAVA) или в виде cgi модулей (программ на Perl или C). ИС данной архитектуры включает Web-yinh\, реализованные при помощи технологий CORBA Enterprise JavaBeans, ActiveX 1X'OM, многоуровневые приложения на основе Java и XML, .Net-концепция с XML, в которой обмен между различными серверами (хранилищами данных, бизнес-приложениями, серверами для мобильных клиентов и другое) производится при помощи нейтрального к любой архитектуре XML.

Под распределенной информационной базой понимается неограниченное количество баз данных, дистанционно отдаленных друг от друга и имеющих ряд общих характеристик:

- функционирующих по единым правилам, определенным централизованно для всех баз данных, входящих в распределенную информационную базу;
- обмен данными осуществляется по правилам, также определенным централизованно.

Организация распределенной базы необходима для компаний, осуществляющих различные виды деятельности, если в их повседневной работе возникает потребность решения следующих задач:

- необходимость оперативного получения информации из баз данных дистанционно отдаленных подразделений (или филиалов);
- необходимость консолидации в единой базе данных информации из баз данных юридических лиц, входящих в структуру компании, для последующего анализа данных и получения отчетности из одной базы, как по компании в целом, так и по каждому юридическому лицу в отдельности;
- необходимость введения централизованного изменения структуры и правил работы баз данных для работы всех дистанционно отдаленных подразделений (филиалов) и юридических лиц (с невозможностью изменения определенных правил непосредственно в отдаленном подразделении);
- необходимость ограничения и осуществления контроля изменения данных в дистанционно отдаленных подразделениях компании (филиалах).

