слайд 2

Целью работы являлась разработка библиотеки имитационного моделирования для языка Haskell. За образец была взята система имитационного моделирования общего назначения GPSS (как ...). Так как в рамках дипломного проекта не представляется возможным реализовать все функции и возможности GPSS, прежде всего необходимо было выбрать некоторое подмножество функций этой системы, которые и следует реализовать. В частности необходимо обозначить множество блоков, разрешенных к испольованию при формировании модели.

Чтобы осуществитьэтот выбор, было принято решение рассмотреть учебную задачу из курса МОК АПК, и реализовать множество блоков необходимое для решения этой задачи.

слайд 3

В качетсве типовой задачи была выбрана следующая:

В вычислительной системе, содержащей N процессоров и M каналов обмена данными, постоянно находятся K задач. Разработать модель, оцени- вающую производительность системы с учетом отказов и восстановлений процессоров и каналов. Имеется не более L ремонтных бригад, которые ремонтируют отказывающие устройства с бесприоритетной дисциплиной. Интенсивность отказов, восстановлений, средние времена обработки сооб- щения и среднее время обдумывания также известны.

слайд 4

Для моделирования описанной в задаче системы понадобятся следующие блоки: –Generate и terminate для создания и уничтожения заявок –advance для задержки заявок, чтобы промоделировать время, которое тратится на обработку задач на различных фазах –seize и release. так как мпромоделировать многоканальное обслуживающее устройство, с возможностью независимого выхода из строя отдельных каналов при помощи хранилищ не представляется возможным, устрйоства обслуживания придется моделировать при помощи множества блокво seize/release и блока transfer в режиме all/ –enter leave предназначены для занятия и освобождения ресурсво хранилища, которое моделирует ремонтные бригады –preempt/return испольуются для моделирования выхода из строя процессоров и каналов. –transfer помимо вышесказанного используется для моделирования замкнутости системы.

слайд 5

Процесс формирования модели может быть представлен как вычисление с состоянием. Начальное состояние представляет собой вырожденную модель, не содержащую ни одного блока. Каждая последующая функция принимает на входе текущее состояние модели и параметры добавляемого боока и на выходе выдает новое состояние модели с добавленым блоком.

Блогадоря механизму монад haskell удалось скрыть явную передачу состояния между вызовами функций формирующих блоки. А благодаря нотации "do удалось записать вызовы этих функций в виде схожем с синтаксисом gpss.

слайд 6

Результат такого представления показан на слайде. Видно, что несмотря на очевидные различия (названия боков написаны строчными буквами, параметры взяты в скобки а имена очередей и устройств – в кавычки) в целом разработанный синтаксис весьма схож с синтаксисом gpss и может быть легко освоен человеком знакомым с этой системой.

Слайд 7

Состояние модели в процессе моделирования хранит следующую информацию: текущее модельное время список блоков, формирующих модель списки устройств, хранилищ и очередей модели списки будущих и текущих событий, в которых соответственно хранятся транзакты, продвижение которых отложено до наступления некоторого момента модельного времени и транзакты, готовые к продвижению прямо сейчас

состояния обслуживающих устройств и хранилищ характеризуется числом доступных ресурсов, статистикой их использования на текущий момент а также списками транзактов, которые были заблокированы при попытке занять устройство и ожидают его освобождения

слайд 8

сам процесс имитационного моделирования модет быть описан следущим алгоритмомю

на первом шаге происходит активация всех блоков generate, определяетсЯ время появляения первых транзактов и они щаносятся в списко будущих событий.

затем транзакты с наименьшим временем переносятся из списка будущих событий в список текущих событий. при этом модельное время изменяется.

на третьем шаге, до тех пор пока список ткущих событий не опустеет из него извлекается транзакт с наибольшим приоритетом и перемещается от блока в блок, до тех пор пока не будет уничтожен, заблокирован или перенесен в список будущих событий.

шаги два и три повторяются до тех пор, пока не будет завершено достаточное чилос транзактов, указанное при запуске процесса моделирования.

слайд 9

в процессе продвижения транзактов по блоокам они оказывают воздействие на остальную часть системы. конкретный вид воздецтсвия зависит от того, в какой именно блок заходит транзакт. алгоритмы обработчиков захода транзакта в блоки enter и leave показаны а слайде.

при заходе транзакта в бок enter определяется есть ли в соответствующем хранилище достаточное количество свободны ресурсов. если ресурсов достаточно, ресурсы выделяются транзакту и он продолжает двидение. в противном случае транзакт попадает в список заблокированных транзактов данного хранилища.

при заходе в блок leave транзакт освобождает заданное количество ресурсов. поосле этого проверяется есть ли транзакты в списке заблокированных транзактов соответствущего хранилища.если траназкты там есть, то выбирается транзакт с наибольшим приоритетом и если ресурсов для него стало ддостаточно, они ему выделяются и траназкт продолжает движение по блокам, в проивном случае он остается в списке заблокированных.

слайд 10

общая структура разработанной библиотеки покаана на слайде. можно выделить группу модулей отвечающих за формирование модели, модуль отвечающий непосредственно за процесс имитационного моделирования, группу модулей с обработчиками захода транзактов в различный блоки и интерфейсный модуль, предназначенный для непосредственного подключения к программе использующей библиотеку и экспортируюющий все нужные сущности.

слайд 11

в целях проверки реализованных алгоритмов было построено аналитическое решение рассмотренной ранее типовой задачи.

сперва, при помощи метода укрупнения состояний модели были вычилены вероятности отказа того или иного количества процессоров и каналов. так как для починки используется общая група ремнтных бригад, эти вероятности взаимно зависят друг от друга. поэтому задачу пришлось решать итерационно. в качестве начального приближения были равные вероятности отказа любого возможного числа каналов.

затем дважды применив метод укрупнения модели сперва была получена интенсивность обработки на укрупненном узле, полученным путем объединения процессороной и канальной фазы,а затем и искомая производительность всей системы.

слайд 12

в целях демонстрации возможностей разработанной библиотеки, а также проверки корректности реализации приведенных алгоритмов была разработанна демонстрационная программа.

в пользовательском интерфейсе программы задаются парметры системы из рассмотреной цифровой задачи и выбирается параметрт подлежащий тварьированию. затем, при помощи рассмотреной аналлитической модели и имитационной модели реализованой библиотеки вычисляется производительность моделируемой системы при различных значениях выбранного параметраа и строятся графики, полученных зависимостей.

слайд 13

Было проведено модульное тестирование разработанного по. на слайде показана степень покрытия тестами кода разных частей системы. тестирование производилось как на системах семейства Linux (Ubuntu 12.04 и 13.04) так и на windows 7. На всех системах тесты были успещо пройдены.

слайд 14

в качестве дополнительного тестирования было проведено сравнение аналитического решения рассмотреной типовой задачи с решением полученным при помощи разработанной библиотеки.

на данном слайде показана зависимость производтельности моделируемой системы от интенсивности обработки задач на канальной фазе. видно, что результаты полученные путем имитационного моделирования соответствуют полученным аналитически

слайд 15

здесь показана ависимость производительность от интенсивности отказов процессоров. видно что результаты имитационного моделирования также соответствут аналитическим с хорошей точностью.

помимо продемонстрированных был проведен еще ряд опытов с варьированием различных параметров. во всех из них результаты имитационной модели соответствовали аналитическим.

слайд 16

В ходе работы была разработана и реализована бибилиотека имитационного моделирования, позволяющая описывать и исследовать заданный класс систем массового обслуживания. – Построены аналитическая и имитационная модель тестовой системы. – Реализована демонстрационная программа, показывающая возможности разработанной библиотеки. – Проведен ряд опытов, подтверждающих корректность разработанных алгоритмов и их реализации.