Модуль визуализации поиска решения на графе состояний в РДО

Содержание

Перечень сокращений

Терминология

1. Введение
2. Предпроектное исследование
   1. Основные положения языка РДО
   2. Поиск решения на графе пространства состояний
   3. Программный комплекс RAO-XT
   4. Постановка задачи
3. Формирование ТЗ
   1. Введение
   2. Общие сведения
   3. Назначение разработки
   4. Требование к программе или программному изделию
      1. Требования к функциональным характеристикам
      2. Требования к надежности
      3. Условия эксплуатации
      4. Требование к составу и параметрам технических средств
      5. Требование к информационной и программной совместимости
      6. Требование к маркировке и упаковке
      7. Требование к транспортированию и хранению
   5. Требования к программной документации
   6. Стадии и этапы разработки
   7. Порядок контроля и приемки
4. Концептуальный этап проектирования
   1. Место модуля визуализации в системе RAO-XT
   2. Взаимодействие (?)
   3. –
   4. –
5. Технический этап проектирования
   1. Выбор графической библиотеки
   2. Синтаксический анализ бинарных сериализованных данных
   3. Построение древовидной структуры
   4. –
6. Рабочий этап проектирования
   1. Вывод графа на экран
      1. Привязка к трассировке
      2. Вывод графов для нескольких точек принятия решений типа search
   2. Расположение графа в окне
   3. Вывод статистики по поиску решения
   4. –
7. Апробирование разработанной системы в модельных условиях
8. Заключение

Список используемых источников

Список использованного ПО

Приложение А. Исходный код модели, использованной для тестирования модуля

Перечень сокращений

ИМ – Имитационное Моделирование

СДС – Сложная Дискретная Система

IDE – Integrated Development Environment (Интегрированная Среда Разработки)

Терминология

Плагин — независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей.

Трассировка — получение информационных сообщений о работе приложения во время выполнения.

Сериализация — процесс перевода какой-либо структуры данных в последовательность битов.

1. Введение

Имитационное моделирование (ИМ)[1] на ЭВМ находит широкое применение при исследовании и управлении сложными дискретными системами (СДС) и процессами, в них протекающими. К таким системам можно отнести экономические и производственные объекты, морские порты, аэропорты, комплексы перекачки нефти и газа, ирригационные системы, программное обеспечение сложных систем управления, вычислительные сети и многие другие. Широкое использование ИМ объясняется тем, что размерность решаемых задач и неформализуемость сложных систем не позволяют использовать строгие методы оптимизации. Эти классы задач определяются тем, что при их решении необходимо одновременно учитывать факторы неопределенности, динамическую взаимную обусловленность текущих решений и последующих событий, комплексную взаимозависимость между управляемыми переменными исследуемой системы, а часто и строго дискретную и четко определенную последовательность интервалов времени. Указанные особенности свойственны всем сложным системам.

Проведение имитационного эксперимента позволяет:

1. Сделать выводы о поведении СДС и ее особенностях:

* без ее построения, если это проектируемая система;
* без вмешательства в ее функционирование, если это действующая система, проведение экспериментов над которой или слишком дорого, или небезопасно;
* без ее разрушения, если цель эксперимента состоит в определении пределов воздействия на систему.

1. Синтезировать и исследовать стратегии управления.
2. Прогнозировать и планировать функционирование системы в будущем.
3. Обучать и тренировать управленческий персонал и т.д.

ИМ является эффективным, но и не лишенным недостатков, методом. Трудности использования ИМ, связаны с обеспечением адекватности описания системы, интерпретацией результатов, обеспечением стохастической сходимости процесса моделирования, решением проблемы размерности и т.п. К проблемам применения ИМ следует отнести также и большую трудоемкость данного метода.

Интеллектуальное ИМ, характеризующиеся возможностью использования методов искусственного интеллекта и прежде всего знаний, при принятии решений в процессе имитации, при управлении имитационным экспериментом, при реализации интерфейса пользователя, создании информационных банков ИМ, использовании нечетких данных, снимает часть проблем использования ИМ.

Разработка интеллектуальной среды имитационного моделирования РДО выполнена в Московском государственном техническом университете (МГТУ им. Н. Э. Баумана) на кафедре "Компьютерные системы автоматизации производства". Причинами ее проведения и создания РДО явились требования универсальности ИМ относительно классов моделируемых систем и процессов, легкости модификации моделей, моделирования сложных систем управления совместно с управляемым объектом (включая использование ИМ в управлении в реальном масштабе времени) и ряд других, сформировавшихся у разработчиков при выполнении работ, связанных с системным анализом и организационным управлением сложными системами различной природы.

2. Предпроектное исследование

2.1 Основные положения языка РДО

Основные положения системы РДО могут быть сформулированы следующим образом:

* Все элементы СДС представлены как ресурсы, описываемые некоторыми параметрами. Ресурсы могут быть разбиты на несколько типов; каждый ресурс определенного типа описывается одними и теми же параметрами.
* Состояние ресурса определяется вектором значений всех его параметров; состояние СДС - значением всех параметров всех ресурсов.
* Процесс, протекающий в СДС, описывается как последовательность целенаправленных действий и нерегулярных событий, изменяющих определенным образом состояние ресурсов; действия ограничены во времени двумя событиями: событиями начала и событиями конца.
* Нерегулярные события описывают изменения состояния СДС, непредсказуемые в рамках продукционной модели системы (влияние внешних по отоношению к СДС факторов либо факторов, внутренних по отношению к ресурсам СДС). Моменты наступления нерегулярных событий случайны.
* Действия описываются операциями, которые представляют собой модифицированные продукционные правила, учитывающие временные связи. Операция описывает предусловия, которым должно удовлетворять состояние участвующих в операции ресурсов, и правила изменения состояния ресурсов в начале и в конце соответствующего действия.
* Множество ресурсов R и множество операций O образуют модель СДС.

2.2. Поиск решения на графе пространства состояний

В языке имитационного моделирования РДО помимо возможности использования для описания законов управления формализмов продукционных правил введены так называемые точки принятия решений, позволяющие осуществлять оптимальное управление.

Механизм точек принятия решений в языке имитационного моделирования РДО позволяет гибко сочетать имитацию с оптимизацией. Для этого используется поиск на графе состояний.

Примерами задач, которые решаются с использованием точек принятия решений, могут служить:

* Различные транспортные задачи (например, выбор последовательности объезда пунктов транспортным средством при минимуме пройденного пути, времени или стоимости).
* Задачи укладки грузов при минимизации занимаемого ими объема (в более общем случае – задачи размещения).
* Нахождение решения логических задач за минимальное число ходов (например, расстановка фишек в игре «Пятнашки»).
* Задачи теории расписаний (например, задачи определения последовательности обработки различных деталей на станках при минимизации времени обработки, либо обслуживания клиентов с минимумом отклонений от запланированного времени изготовления заказов и т. д.).

Граф — основной объект изучения математической теории графов, совокупность непустого множества вершин и наборов пар вершин (связей между вершинами).

Граф задается двумя множествами:

* – множество вершин графа. Каждой вершине ставится в соответствие состояние системы (база данных);
* – множество дуг. Каждой дуге , принадлежащей и соединяющей пару вершин, ставится в соответствие правило преобразования (продукционное правило). Если дуга направлена от вершины к вершине , то в данном случае будет являться вершиной-родителем, а – вершиной-потомком (преемником).

Маршрутом в графе называют конечную последовательность вершин, в которой каждая вершина (кроме последней) соединена со следующей в последовательности вершиной ребром.

Цепью называется маршрут без повторяющихся рёбер.

Циклом называют цепь, в которой первая и последняя вершины совпадают.

Граф называется связным, если для любых вершин существует путь из в .

Граф называется деревом, если он связный и не содержит нетривиальных циклов.

В графах, представляющих интерес для поиска, у каждой вершины должно быть конечное число вершин-преемников. С дугой может быть связана некоторая величина – стоимость дуги, она отражает затраты (в смысле заданного критерия оптимизации) применения соответствующего правила.

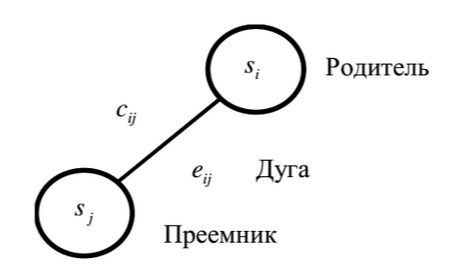
**

Рис 1. Простейший граф состояний

В графе имеются две особые вершины:

* – начальная вершина, или другими словами, вершина, представляющая собой исходную базу данных;
* – целевая вершина, иначе – вершина, представляющая собой базу данных, удовлетворяющую терминальному условию поиска. Таких вершин может быть не одна, а множество, и такое множество будет называться целевым множеством.

Путь в графе – последовательность вершин, в которой каждая последующая является преемником:

,

где – длина пути.

Каждому пути ставится в соответствие его стоимость, которая равна сумме стоимостей применения правил по всему пути на графе .

2.3. Программный комплекс RAO-XT

Программный комплекс RAO-XT предназначен для разработки и отладки имитационных моделей на языке РДО. Основные цели данного комплекса - обеспечение пользователя легким в обращении, но достаточно мощным средством разработки текстов моделей на языке РДО, обладающим большинством функций по работе с текстами программ, характерных для сред программирования, а также средствами проведения и обработки результатов имитационных экспериментов.

Система имитационного моделирования RAO-XT представляет собой плагин для интегрированной среды разработки Eclipse – свободной интегрированной среды разработки модульных кроссплатформенных приложений. Система написана на языке Java и состоит из трех основных компонентов:

* rdo – компонент, производящий преобразование кода на языке РДО в код на языке Java.
* rdo.lib – библиотека системы. Этот компонент реализует ядро системы имитационного моделирования.
* rdo.ui – компонент, реализующий графический интерфейс системы с помощью библиотеки SWT.

На момент начала выполнения курсового проекта, система не имела возможности выводить на экран пользователя граф пространства состояний для моделей, содержащих точки принятия решений. Вывод информации о поиске осуществлялся в текстовом формате в графическом окне модуля трассировки.

2.4 Постановка задачи

Проектирование любой системы начинается с выявления проблемы, для которой она создается. Под проблемой понимается несовпадение характеристик состояния систем, существующей и желаемой.

В результате предпроектного исследования было выявлено отсутствие в программном комплексе RAO-XT подсистемы графической визуализации поиска решения на графе пространства состояний.

Для повышения эффективности процесса моделирования была установлена необходимость разработать и внедрить в комплекс RAO‑XT такую подсистему.

3. Формирование ТЗ

3.1. Введение

Целью выполнения работ на стадии формирования ТЗ на проектируемую систему является уточнение и детализация требований заказчика, а также разработка требований к составу и содержанию работ по созданию системы, порядку приемки, документации, к составу и содержанию работ по подготовке проектируемой системы к эксплуатации.

3.2. Общие сведения

Основание для разработки: задание на курсовой проект.

Заказчик: Кафедра «Компьютерные системы автоматизации производства» МГТУ им. Н. Э. Баумана

Разработчик: студент кафедры «Компьютерные системы автоматизации производства» Стрижов К. А.

Наименование темы разработки: «Проектирование модуля визуализации поиска решения на графе состояний в РДО»

3.3. Назначение разработки

Разработать подсистему визуализации поиска решений на графе состояний и добиться ее интеграции в текущую рабочую версию среды RAO-XT.

Функциональным назначением разрабатываемого модуля является предоставлению пользователю системы RAO-XT возможности получить информацию о результатах моделирования в графическом формате, облегчить восприятие собранной по модели статистической информации.

3.4. Требование к программе или программному изделию

3.4.1. Требования к функциональным характеристикам

Модуль визуализации должен удовлетворять следующим требованиям:

* Корректно отображать граф пространства состояний для соответствующей точки принятия решений типа search;
* Вызов графического окна модуля должен осуществляться по нажатию на соответствующую точке принятия решений строку модуля трассировки системы RAO-XT;
* Графическое окно должно содержать всю необходимую информацию о точке принятия решений, как-то:
  + - Название точки принятия решений;
    - Статистика по поиску на графе состояний;
    - Подробная информация по вершине при выделении ее кликом мыши;
    - Выделение цветом вершин графа, относящихся к решению, в случае его существования оного;
* Иметь возможность отображать несколько графов в случае наличия в модели нескольких точек принятия решений.

3.4.2. Требования к надежности

Основное требование к надежности направлено на поддержание в исправном и работоспособном состоянии ЭВМ, на которой происходит использование программного комплекса RAO-XT.

Сохранение работоспособности системы при отказе по любым причинам подсистемы или ее части

Количество отказов из-за не выявленных ошибок не более 1 на 1000 сеансов работы с программой

3.4.3. Условия эксплуатации

Эксплуатация должна производиться на оборудовании, отвечающем требованиями к составу и параметрам технических средств, и с применением программных средств, отвечающим требованиям к программной совместимости.

Аппаратные средства должны эксплуатироваться в помещениях с выделенной розеточной электросетью 220В ±10%, 50 Гц с защитным заземлением.

3.4.4. Требование к составу и параметрам технических средств

Программный продукт должен работать на компьютерах со следующими характеристиками:

* объем ОЗУ не менее 1 Гб;
* объем жесткого диска не менее 50 Гб;
* микропроцессор с тактовой частотой не менее 1ГГц;
* монитор с разрешением от 800\*600 и выше.

3.4.5. Требование к информационной и программной совместимости

Система должна работать под управлением следующих ОС:

* Windows 7;
* Ubuntu 14.10.

3.4.6. Требование к маркировке и упаковке

Требования к маркировке и упаковке не предъявляются.

3.4.7. Требование к транспортированию и хранению

Требования к транспортированию и хранению не предъявляются.

3.5. Требования к программной документации

Требования к программной документации не предъявляются.

3.6. Стадии и этапы разработки

Разработка должна быть проведена в три стадии:

* техническое задание
* технический и рабочий проекты
* внедрение

На стадии «Техническое задание» должен быть выполнен этап разработки и согласования настоящего технического задания.

На стадии «Технический и рабочий проект» должны быть выполнены перечисленные ниже этапы работ:

* разработка программы
* разработка методики тестирования
* испытания программы

На стадии «Внедрение» должен быть выполнен этап разработки «Подготовка и передача программы».

3.7. Порядок контроля и приемки

Контроль и приемка работоспособности системы осуществляются с помощью следующих методов:

* Опрос экспертов. Используется для оценки дизайна, качества и удобства использования новой системы графического вывода;
* Многократное тестирование с помощью имитационных моделей, написанных на языке РДО.