

робного описания работы участка эти моменты могут быть смоделированы, но в настоящем исследовании перед разработчиками такой задачи не стояло.

Подводя итог вышесказанному необходимо отметить, что анализ обозначенных проблем позволил найти эффективные пути их решения. Часть из них, удалось решить успешным выбором среды моделирования AnyLogic. Для компенсации недостатка быстродействия, помимо обычного в таких случаях укрупнения детализации модели используется выделение части функций в отдельную программу. Повышение точности модели достигается сбором статистических сведений, и последующим вводом корректирующих параметров.

Учитывая вышеприведенные обстоятельства, можно утверждать, что имитационная модель не потеряла своей адекватности и реально отражает происходящие на участке производственные процессы, что было подтверждено совпадением соответствующих практических данных и данных, полученных в ходе проведения моделирования деятельности производственного участка.

Библиографический список

1. Карпов Ю.Г. Введение в моделирование с AnyLogic 5 / Ю.Г.Карпов. – М. : «ВНУ», 2005. – 400 с.
2. Техническая документация ОАО «АСМ-Запчасть» технологического процесса производства культиваторной лапы АСМ 00120 («сор-май»).

Модель для тестирования ПО

С.М. Старолетов

АлтГТУ, г. Барнаул

Работа посвящена математизации тестирования (проверки правильности) современных программ.

Целью исследования является создание своей адекватной математической модели распределённой недетерминированной программной системы и применение её при проведении тестирования сложных систем при помощи разрабатываемых средств. Актуальность работы вытекает из необходимости применения различных методик при тестировании программ, перспективности применения моделей в тестировании программ, необходимости повышения культуры тестирования, отсутствия на рынке популярных средств тестирования при помощи построения моделей.

Ядром исследования является математическая модель взаимодействующих систем. Система представляет собой набор компонентов и связей между ними, а также глобальные множества отсылаемых сообщений и общих блокируемых ресурсов:

$$M = (A^*, CP(A^*), Msg, Res).$$

Здесь A^* – множество расширенных вероятностных многопоточных конечных автоматов, моделирующих поведение компонентов системы в виде мультиагентных программ:

$$A = (q_0, Q, F, \delta, E, msg \subseteq Msg, res \subseteq Res),$$

где Q – множество состояний, $q_0 \in Q$ – начальное, $F \subseteq Q$ – множество заключительных состояний; E – множество событий и исключительных ситуаций; msg – подмножество глобального множества отсылаемых сообщений; res – подмножество глобального множества общих ресурсов; логика переходов описывается недетерминированной функцией переходов δ по вершинам и γ – функцией переходов по ребрам.

Приведем лишь краткую логику работы: находясь в состоянии q кратности n по действию d с вероятностью p в потоке t и по полученному сообщению msg , после снятия блокировки ресурса res модель системы недетерминировано переходит либо в несколько состояний, создав несколько новых потоков из t , или просто в следующее состояние в текущем потоке; при этом возможно возникновение и обработка конечного числа событий или исключительных ситуаций, отсылка сообщения и блокировка некоего ресурса. Здесь под понятием «кратность состояния» понимается число одновременно находящихся в этом состоянии потоков.

$CP(A^*)$ определяет множество точек сопряжения компонентов системы, а также мощности их связей $(I:1, I:N, N:N)$, по принципу «рукопожатия».

Следующим шагом исследования является язык описания модели, который не зависит от используемых при создании системы языков программирования (компоненты распределенной системы могут быть написаны на различных языках) и определяет формальное текстовое описание разработанной модели. Применяется принцип «код и модель – одно целое», при котором описание модели на языке встраивается в исходный код системы в виде комментариев специального вида. Сущности «состояние», «переход», «поток», «исключение» и др. описываются на разработанном языке XML тегами с атрибутами.

Методы исследования программы и ее модели включают анализ взаимодействующих систем, формализация понятий, применение тео-

рии конечных автоматов, теории графов, теории вероятности и Марковских случайных процессов, теории формальных языков, теории алгоритмов.

Для популярных сред разработки Eclipse и Visual Studio разрабатываются расширения (Plugin, VS Package) для визуального описания моделей прямо при создании. Собственно тестирование заключается, во-первых, в анализе модели без запуска системы (off-line), когда сложная структура модели специальным методом преобразуется в ориентированный граф и далее исследуется, и, во-вторых, с запуском системы и сравнение реальной работы системы и модели с использованием сервера тестирования (код обращения к серверу встраивается препроцессором в места описания модели).

Разработка поддержана грантом «УМНИК».

Библиографический список

1. Старолетов, С.М. Тестирование недетерминированного программного обеспечения на основе моделей / С.М. Старолетов // X Белорусская математическая конференция : тезисы докладов Международной научной конференции, Минск, 3-7 ноября 2008 года. – Минск, 2008. – С. 90.

2. Старолетов, С.М. Тестирование распределенных приложений на основе построения моделей / С.М. Старолетов, Е.Н. Крючкова // Прикладная информатика. – М. : Market DS publishing. – 2008. – №6. – С. 124–134.

О роли визуализаторов при обучении алгоритмам

Т.М. Тушкина, Н.В. Павлова

БТИ АлтГТУ, г. Бийск

Самостоятельное изучение студентами алгоритмов обработки данных, как правило, сопровождается сложностью, связанной с тем, что в учебной литературе описание этих алгоритмов изложено зачастую недостаточно детально. В этом случае на помощь студенту могут прийти визуализаторы алгоритмов – программы, демонстрирующие на экране компьютера применение алгоритма к определенным данным, которые можно варьировать. Основной особенностью визуализатора является то, что он позволяет изучать работу алгоритма пошагово, возможно укрупненными блоками.

В настоящее время авторами в рамках процессов НИРС и курсового проектирования ведется работа по созданию визуализаторов ряда алгоритмов, например, сортировки элементов массива, пересчета и