

УДК 681.3

И.А.Буячевский, А.Н.Коварцев, М.А.Шамашов

НЕКОТОРЫЕ ВОПРОСЫ МОДЕЛИРОВАНИЯ СИСТЕМ СБОРА И ОБРАБОТКИ ДАННЫХ НАУЧНОГО ЭКСПЕРИМЕНТА

В настоящее время тенденция использования средств автоматизированного управления научным экспериментом (НЭ) поставила ряд задач, связанных с проектированием систем сбора и обработки данных (ССО) в том числе и управляющие вычислительные машины (УВМ). Применение УВМ обусловлено большой емкостью обрабатываемой информации, а также сложностью структуры ССО.

Уникальность (в смысле поставленных задач и методов их решения) сложного научного эксперимента предполагает разработку специализированной УВМ, ориентированной на решение задач данного НЭ. Такой процесс дорог и сопряжен с большими трудозатратами, поэтому становится весьма актуальным вопрос автоматизации проектирования как УВМ, так и всей ССО в целом.

Эту задачу частично решает построение имитационной системы, моделирующей ССО. Кроме того, имитационная модель (ИМ) позволяет исследовать сопряжение УВМ с работой ССО, а также провести анализ алгоритмов управления и обработки данных НЭ.

Предлагаемая модель может применяться как на этапе структурного проектирования ССО, когда основное внимание уделяется не столько содержательной стороне информации, сколько временным соотношениям между работой компонентов системы в процессе ее функционирования и принципам организации прохождения потока информации (алгоритмы диспетчеризации информационных потоков), так и на этапе доработки и отладки программ УВМ, осуществляющих управление и обработку данных НЭ.

Эту модель можно разделить на три подсистемы :

1. Модель измерительной системы, включающей ряд физических устройств, а также внешнюю среду.

2. Модель УВМ;

3. Группа управляющих подпрограмм - программа "Календарь" I.

Последняя группа подпрограмм дает возможность реализовать на ЭЦВМ работу ИМ таким образом, как она происходит в действительности,

то есть соблюдена причинно-следственные связи (во времени) между событиями, вызванными совместной работой измерительной системы и УВМ.

Программа "Календарь" строится на базе принципа "управления событиями" (такая концепция реализована, например, в языке "Симскрипт"). Это позволяет разработчикам системы создавать модели измерительной системы и УВМ независимо.

Модель измерительной системы. Измерительная система (ИМ) включает модели научной аппаратуры, устройств сбора, передачи, предварительного сжатия и хранения данных НЭ, а также модель внешней среды.

Обычно с целью ускорения процесса обмена информации и упрощения управления НЭ в измерительных системах на различных уровнях ставятся коммутаторы, осуществляющие снятие, хранение и управление информационными потоками, поступающими с научной аппаратуры.

В общем случае (на высоких уровнях коммутации) модель коммутатора строится достаточно просто. Значительно более сложно моделировать так называемые локальные коммутаторы, которые непосредственно работают с измерительной аппаратурой.

Такие локальные коммутаторы (ЛК) характеризуются наличием устройств первичного сжатия информации, буферной памяти (БП), многорежимностью работы, а тот факт, что ЛК снимают информацию непосредственно с датчиков и научной аппаратуры, вызывает необходимость моделировать и внешнюю среду.

Создать универсальную модель ЛК, ввиду многообразия существующих физических систем, невозможно (даже если бы такая модель была создана, то настройка последней на заданную физическую систему само по себе сложная задача). Однако общие принципы работы таких систем позволяют сделать предположение, что методы разработки подобных моделей едины. Наиболее трудную часть здесь составляет модель внешней среды, влияющей на показания датчиков измерительной системы, а следовательно и на функционирование всего комплекса в целом.

Рассмотрим пример модели ЛК, имеющего два режима работы (программный и адаптивный) и осуществляющего процесс снятия информации с рентгеновских телескопов. В данном случае внешней средой является рентгеновское излучение заданной области пространства. Как известно такое излучение с высокой достоверностью описывается простейшим потоком. Если учесть, что замеры накопленной энергии в

камере телескопа производится через заданные промежутки времени T , то количество элементарных частиц, попавших в камеру за это время, можно описать Пуассоновским законом распределения [2]. Изменение поведения внешней среды (например, попадание в область объектива сильных источников рентгеновского излучения) можно описать, изменяя интенсивность потока частиц λ во времени. К упрощающим предположениям при создании имитационной модели ЛК можно отнести предположения о том, что за время опроса датчиков T изменение $\lambda(\xi)$ потока частиц меняется незначительно, следовательно, для моделирования можно использовать следующие формулы:

$$P_m(t_k, t_{k+1}) = \frac{[\lambda(\xi)]^m}{m!} e^{-\lambda(\xi)}, \quad (1)$$

где

$$\lambda(\xi) = \int_{t_k}^{t_{k+1}} \lambda(t) dt = \lambda(\xi) \cdot T \quad \text{и} \quad t_k < \xi < t_{k+1}$$

$P_m(t_k, t_{k+1})$ - вероятность того, что за интервал времени (t_k, t_{k+1}) в камеру попадет ровно m частиц. Реализация (1) на ЭЦВМ не представляет особого труда, для этого используется датчик псевдослучайных чисел экспоненциального закона распределения, на отрезке времени $[t_k, t_{k+1}]$ разыгрывается случайная величина интервала поступления частиц в камеру ξ_i и отмечается на временной шкале $\xi_{i+1} = \xi_i + \xi_{i+1}$ до тех пор, пока не выполнится условие $\xi_{i+1} > t_{k+1}$, тогда число скачков за время $T = t_{k+1} - t_k$ будет равно i и распределено по закону (1).

По формуле Пальма [1] закон распределения интервала времени до первого прибытия частицы в камеру будет также экспоненциальным.

Кроме модели внешней среды и НА, ЛК включает также устройства сжатия информации, БП и т. д. В задачи данной статьи не входит детальное рассмотрение моделей этих устройств, поэтому на них останавливаться не будем. С большим или меньшим успехом строятся модели и других коммутаторов, что в совокупности составляет модель измерительной системы, работа которой управляется УВМ.

Модель УВМ. В настоящее время в идеале стремятся к созданию полностью параметризуемой программы, способной моделировать любую из существующих или предполагаемых вычислительных машин. Однако в ближайшем будущем построение такой программы вряд ли реально из-за частых изменений в технологии и организации ЭВМ и широком разнообразии целей моделирования.

В предлагаемой работе разрабатывалась программа, моделирующая конкретную специализированную УВМ. Эта модель включает в себя не только интерпретатор арифметических и логических операций одной ЭВМ на другой. Написать такой интерпретатор довольно просто. Имитационная модель ССО требует эмулятора полной конфигурации УВМ, включая периферию. При этом наиболее существенно моделирование системы прерываний (СПР), предназначенной для обеспечения мультипрограммного режима работы УВМ, и эмуляция команд обмена, позволяющих осуществлять связь между измерительной системой и управляющей машиной.

На первоначальном этапе моделирование СПР, УВМ и построение алгоритмов модулей, осуществляющих интерпретацию команд обмена целесообразно использовать *Computer Design Language* [2] — язык описания вычислительных машин, позволяющий опеределить функциональную организацию, алгоритмы и последовательности выполнения операций на УВМ. *CDL* обладает мощными средствами для описания элементов машины — регистров, индикаторов, блоков памяти, терминалов. С помощью языка описания вычислительных машин можно наглядно представить информацию о статусе вычислительной машины, которая включает состояние регистров, ячеек памяти и т.д. Описание на *CDL* дает возможность в дальнейшем достаточно просто эмулировать изменение массива "Регистры и память УВМ", интерпретируя команды УВМ совокупностью команд языка высокого уровня (ПЛ/Т), а также с помощью Ассемблера.

Так как скорость эмуляции является чрезвычайно важной для обеспечения сравнимого масштаба времени вычислительного процесса и допустимым считается замедление в 10–20 раз, очевидно, следует писать эмулятор УВМ на Ассемблере, обеспечив удобное обращение к эмулятору и управление им.

Опыт разработки подобных имитационных систем за рубежом показывает, что наиболее используемые результаты дает метод моделирования, когда внутреннее функционирование системы воспроизводится в привязке к календарю текущих событий [3]. При написании

эмулятора использовался ряд возможностей управляющей программы "Календарь", входящей в состав модели ССО. Наибольшее употребление нашли процедуры : создание события с задержкой, создание события по условию и отмена события.

Под событием в эмуляторе понимается изменение некоторой области массива "память и регистры" ЭВМ, происходящие в один и тот же момент системного времени в реальной ЭВМ.

Проиллюстрировать использование управляющей подпрограммы в эмуляторе можно на примере интерпретации двух команд ввода-вывода, стоящих в одной программе ЭВМ друг за другом. Вторая команда обмена на реальной ЭВМ начнет выполняться только после освобождения канала и устройства ввода-вывода, занятых выполнением первой команды. Освобождение канала и устройства ввода-вывода характеризуется установлением в нуль определенных триггеров. Этот факт отражается в эмуляторе созданием события приравнивания нулю некоторых переменных, соответствующих по описанию на *CDL* этим триггерам, с необходимой задержкой от момента начала первой операции. Вторая операция "начнет выполняться" только при условии равенства нулю этих переменных, то есть используется создание события по условию.

Группа управляющих подпрограмм позволяет достаточно просто сопрягать модель измерительной системы и модель ЭВМ.

Кроме моделирования СПР и интерпретации команд обмена, определенные сложности возникают и при эмуляции циклических участков программ ЭВМ. Для обеспечения достаточно высокой скорости эмуляции видимо невыгодно осуществлять многократное распознавание и раздельную интерпретацию команд управляющей машины, входящих в циклические участки. Чтобы избежать этого, следует переходить от цикла ЭВМ к некоторой промежуточной форме - последовательности ряда интерпретирующих модулей в рамках Ассемблера универсальной ЭЦВМ, соответствующих командам ЭВМ входящих в данный цикл, и многократной реализации этих модулей по числу циклов.

Л И Т Е Р А Т У Р А

1. Гнеденко Н.П. Введение в теорию массового обслуживания. М., "Наука", 1966.
2. Чу Я. Организация ЭВМ и микропрограммирование, М., "Мир", 1975.

3. Facey P, Gaines B. *Real-time system design under an emulator embedded in a high-level language.*
Datatair 78 Conference Papers Vol. 2, London

УДК 681.3.06 : 51

И.А.Будячевский, М.А.Кораблин, С.В.Смирнов

МОДЕЛИРОВАНИЕ ДИСКРЕТНЫХ СИСТЕМ В РАМКАХ ПЛ/І (ДОС ЕС)

I. Практическая реализация метода имитационного моделирования систем с помощью ЭВМ включает в себя ряд необходимых этапов [1,2] Здесь рассматриваются вопросы, связанные с составлением программы для ЭВМ применительно к моделям дискретных систем.

Моделируемую систему будем называть дискретной, если происходящие в ней процессы можно представить в модели некоторой последовательностью дискретных изменений ("событий"). Подобным образом можно описать весьма широкий класс реально функционирующих сложных систем (АСУ производством, экспериментальными исследованиями и т.п.).

В отличие от широко применяемых универсальных языков программирования, трансляторы с которых включены в заводское математическое обеспечение современных ЭВМ, трансляторы со специализированных языков моделирования имеются лишь в отдельных организациях, и получение подобного математического обеспечения с соответствующей документацией весьма проблематично.

Описываемый ниже набор подпрограмм, формализующих и воспроизводящих динамические свойства моделируемой системы, наряду со стандартными возможностями дисковой операционной системы ЕС ЭВМ и имеющегося в ней подмножества языка ПЛ/І [3] позволит пользователю достаточно быстро и при оперировании единой системой понятий составлять сложные машинные имитационные программы. Следует отметить, что средства универсального языка программирования ПЛ/І оказываются весьма удобными для описания таких сложных объектов как системы управления экспериментом, включающие в себя ЭВМ.

II. Имитация динамики поведения моделируемой системы осуществляется в соответствии с имитационной концепцией языков событий