

УДК

К.О. Савенков, к.ф.-м.н., ВМК МГУ имени М.В.Ломоносова, Москва  
Е.В. Чемерицкий, студент 4 курса, ВМК МГУ имени М.В.Ломоносова,  
Москва

**СРЕДА ВЫПОЛНЕНИЯ ДИСКРЕТНО-СОБЫТИЙНЫХ  
ИМИТАЦИОННЫХ МОДЕЛЕЙ: ОТ УНИВЕРСАЛЬНОСТИ К  
РАСШИРЯЕМОСТИ И ПОВТОРНОМУ ИСПОЛЬЗОВАНИЮ  
(DISCRETE-EVENT SIMULATION RUNTIME: FROM GENERICITY TO  
EXTENDABILITY AND REUSE)**

**Short Annotation**

This paper presents an extendable architecture for a discrete-event simulation runtime (DESR). The architecture is based on a set of logic blocks (LB), which encapsulate a part of the DESR functionality and provide an interface to that functionality for other LB. To reflect differences in requirements imposed by different simulation problems are encapsulated in distinct LB. Therefore, there is no need either in performance trade-offs or in a custom development of the whole DESR.

**Введение**

Данная работа посвящена разработке архитектуры среды выполнения дискретно-событийных имитационных моделей, состоящей из набора логических блоков. Набор блоков, их интерфейсы и функциональность позволяют скомпоновать среду выполнения с учётом требований конкретной задачи имитационного моделирования.

Лаборатории вычислительных комплексов МГУ приходится решать множество разнородных исследовательских задач, так или иначе связанных с моделированием распределённых систем. Круг подобных задач включает в себя моделирование бортовых комплексов (авиационных, морских, автомобильных), компьютерных сетей, наборов команд спецпроцессоров. Объединяет эти задачи то, что они нацелены на моделирование функциональности вычислительной системы (обработки данных) и её производительности, а также используемый аппарат – дискретно-событийное имитационное моделирование.

Подобные задачи решаются в ЛВК с 1982 года, и в своё время, с ростом числа проектов и направлений было принято решение создать единую среду выполнения дискретно-событийных имитационных моделей [1]. Она базировалась на едином подходе к моделированию вычислительных систем и использовала для описания моделей специализированный язык.

Однако выяснилось, что подобный подход не вполне применим для разработки наукоёмких исследовательских проектов. Новые задачи несут с

собой необходимость моделирования систем на разнообразных, не предугаданных ранее уровнях детальности. Возникают новые требования по масштабируемости, производительности, времени отклика. Универсальное решение представляет собой компромисс между выразительной мощностью и эффективностью, и в перспективе развития требует несоразмерного объёма усилий для обновления системы и поддержания её в консистентном состоянии.

В итоге для нужд каждого крупного проекта в среду выполнения вносились правки, несовместимые с её применением в других проектах, и в настоящий момент существует несколько активно используемых вариантов одного и того же продукта.

В данной работе мы решили пойти по другому пути – разработать архитектуру среды выполнения имитационных моделей как коллекции логических блоков. Каждый из таких блоков инкапсулирует в себе часть функциональности среды прогона имитационных моделей и предоставляет другим блокам интерфейс к этой функциональности. Набор блоков был разработан таким образом, чтобы различия в требованиях к среде прогона, предъявляемых различными задачами ИМ инкапсулировались в отдельных блоках, с сохранением общей структуры среды прогона. Тогда, обеспечив возможность компоновки среды прогона из нужных экземпляров различных блоков, мы сможем создавать среду выполнения ИМ, настроенную на решение узко поставленной задачи ИМ, без необходимости прибегать к компромиссу в части производительности системы и не расходуя усилия разработчиков на создание и поддержку новой среды выполнения ИМ.

В основу работы лёг сравнительный анализ вариантов среды выполнения ДИАНА, используемых в нескольких проектах: 1) функциональном моделировании бортовой системы морского базирования [3], 2) стенда полунатурного моделирования бортовых авиационных комплексов [4], 3) моделирования производительности набора команд нейропроцессора [5] и 4) моделирования производительности бортовой автомобильной информационной системы. Также были рассмотрены среды выполнения моделей нескольких известных систем моделирования: AutoMod, SLX, Extend, SIMAN V, ProModel, GPSS/H.

По результатам проведённого исследования был разработан набор блоков, инкапсулирующих в себе различия рассмотренных систем. В работе подробно описываются функции каждого предложенного блока и интерфейсы их взаимодействия между собой. На основе предложенных блоков была построена обобщённая схема среды выполнения. Она призвана координировать взаимодействие составляющих блоков между собой. В работе приводится описание её функциональности с чётким разделением функций между отдельными компонентами.

В дополнение к обобщённой схеме был разработан механизм настройки на основе набора конфигурационных параметров. Параметры позволяют адаптировать логику работы обобщённой схемы под требования конкретной

задачи моделирования. В рамках данной работы детально описываются свойства среды выполнения в зависимости от значений каждого параметра. С помощью данных параметров в рамках обобщённой схемы СВ можно воспроизвести логику большей части рассмотренных в ходе исследования сред выполнения.

Работа состоит из трёх частей. В первой части приводится описание обобщённой схемы СВ в терминах выделенных логических блоков. Здесь речь идёт о разделении между блоками функциональности СВ и их взаимодействии между собой, затрагиваются общие принципы работы предлагаемой СВ.

Далее подробнее рассматриваются возможные отличия блоков, вписывающихся в предлагаемую модель. Здесь же прослеживается зависимость свойств собранной СВ от выбора определённого набора составляющих блоков. В этой части выделяются и описываются инструменты настройки обобщённой СВ.

В заключительной части рассматриваются промежуточные результаты, достигнутые на данном этапе разработки, и перспективы дальнейшего развития в данном направлении.

### **1. Обобщённая СВ**

Обобщённая схема СВ предложена на основании проведённого сравнительного анализа СВ различных вариантов системы ДИАНА [2-5] и нескольких известных систем моделирования: AutoMod, SLX, Extend, SIMAN V, ProModel, GPSS/H [6-7].

В работе используются термины из статьи [8], часть которых была обобщена в процессе анализа. Базовыми понятиями обобщённой СВ являются *событие* – сигнал, оповещающий СВ об изменении состояния ИМ, *логический объект* (ЛО), способный планировать события, и *ресурс*, предоставляющий ЛО некоторый сервис. Ресурсами могут служить модельное время, ячейки области памяти (переменные), буфер обмена информацией, семафор и тому подобное. ЛО могут откладывать наступления события, пока не будет выполнено некоторое условие, зависящее от состояния набора ресурсов. Такое условие называется *условием задержки*. В результате наступления события СВ может совершить ряд действий, называемый *обработкой события*. Модельное время, в которое производилась обработка последнего события, называется *горизонтом событий*.

В приведённых терминах выполнение ИМ принимает вид двухфазного цикла:

1. *Фаза обработки событий*. Здесь обрабатываются события с выполненными условиями задержки. Может случиться так, что во время обработки события станут выполнимыми условия задержки других событий. Тогда они будут обработаны в течение этой же фазы. Соответствующую череду обработок событий и появлений новых готовых к обработке событий будем называть циклом обработки

событий.

2. *Фаза изменения времени.* На данной фазе модельное время продвигается до следующего запланированного события. Все операции с событиями происходят в блоках внутренней обработки. Каждый такой блок представляет собой контейнер для событий, сгенерированных ЛО. Любому блоку внутренней обработки можно индивидуально назначить свой планировщик, который будет соответствующим образом ранжировать элементы блока. В обобщённой СВ могут содержаться несколько типов блоков внутренней обработки:
  - a. Блок текущих событий (БТС). Содержит события, готовые к обработке.
  - b. Блок будущих событий (ББС). Здесь хранятся события с условием задержки, включающим в себя только модельное время. Как правило, это список, отсортированный по возрастанию модельного времени. Как только выполняется условие задержки, событие автоматически переносится средой выполнения в БТС и обрабатывается. Модельное время продвигается дискретными шагами именно по временам наступления событий из ББС. Такой подход гарантирует, что время, на которое запланировано событие, не может быть меньше горизонта событий.
  - c. Блок задержанных событий (БЗС). Здесь содержатся события с более сложными условиями задержки, не только от модельного времени, но и от других ресурсов. Как и в ББС, при выполнении условия задержки событие переносится в БТС. Существует два механизма обработки задержек: *пассивное и активное ожидание*. Будем соответственно выделять два подтипа БЗС. В СВ могут одновременно существовать блоки обоих подтипов.

Проверка условий задержки для событий из БЗС активного ожидания производится на каждой итерации цикла обработки событий. При пассивном ожидании к каждому событию из БЗС помимо условия задержки прикрепляется список всех влияющих на него ресурсов. К ресурсам прикрепляются списки зависимых событий, условия которых проверяются при освобождении соответствующих ресурсов. Тогда среди событий с выполненными условиями задержки планировщик БЗС выбирает следующих обладателей ресурса, которые выводятся из состояния задержки и направляются на обработку.

Разработчик модели может задать произвольное число *управляемых блоков (УБ)* – элементов СВ, инкапсулирующих в себе ЛО и ресурсы. Каждый ЛО или ресурс может быть прикреплен к произвольному числу УБ или не прикреплен ни к одному из них. Ресурсы, прикрепленные к УБ, называются приватными. Они доступны только для ЛО из того же УБ. Остальные ресурсы называются глобальными.

Координирует работу частей СВ на всём протяжении выполнения ИМ *диспетчер*. Прежде всего, он производит начальную настройку модели: им

создаются, связываются и инициализируются объявленные ЛО, ресурсы, УБ и блоки внутренней обработки. После этого ИМ запускается на выполнение. На каждой итерации цикла обработки событий диспетчер отслеживает изменения в состоянии системы. Он сам продвигает модельное время системы и инициирует завершение процесса моделирования, когда достигается временной порог, или не остаётся готовых к обработке событий. Схема обобщённой СВ изображена на Рисунке 1.

Теперь рассмотрим подробнее внутреннее устройство ЛО. Каждому ЛО соответствует специальный контейнер для сгенерированных им событий, называемый *локальной очередью событий* (ЛОС).

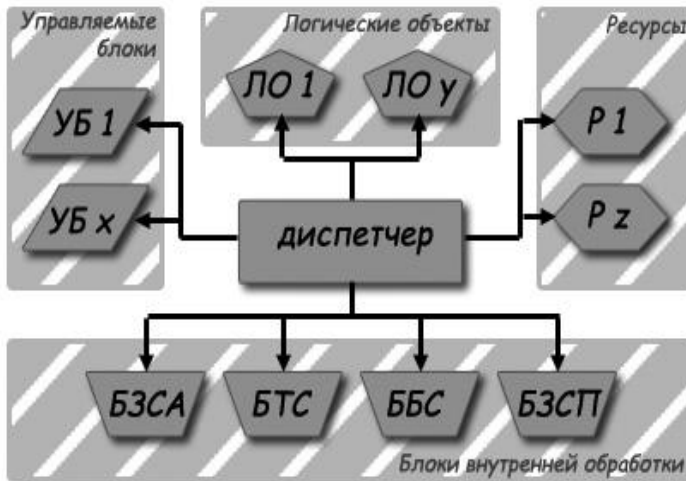


Рисунок 1. Схема обобщённой СВ ДСИМ.

Изначально сгенерированные ЛО события попадают в ЛОС. Затем они могут быть перемещены в блоки внутренней обработки. При этом число созданных данным ЛО событий, находящихся в текущий момент модельного времени в блоках внутренней обработки, называется *уровнем активности* (УА) этого ЛО, а максимально возможный УА – его *пределом активности* (ПА). УА изменяется СВ автоматически: по завершении обработки, событие удаляется из содержащего его блока внутренней обработки, а УА соответствующего ЛО уменьшается; при перемещении события из ЛОС УА, наоборот, увеличивается. С помощью ограничения ПА можно добиться, чтобы не все запланированные события сразу поступали в блоки внутренней обработки. Это позволяет ЛО «убежать» во времени вперёд относительно модели, внешне сохраняя хронологию событий.

ЛО может пребывать в двух состояниях:

1. *Активное*. ЛО находится в этом состоянии, если в данный момент модельного времени обрабатывается соответствующее ему событие.

В активном состоянии ЛО может планировать новые события и изменять состояние модели, в частности ПА любого ЛО, принадлежащего тому же УБ, включая свой собственный. Если ЛО находится в активном состоянии, то иногда говорят, что он обрабатывается.

2. *Спящее* – состояние бездействия ЛО. Спящий ЛО может быть пробуждён только в том случае, если разница его ПА и УА станет положительной. Если ЛО находится в спящем состоянии, то он не участвует в обработке событий, что может увеличить производительность СВ.

## **2. Настройка обобщённой СВ ДСИМ**

В ходе проведённого обзора СВ ДСИМ были выделены признаки, отличающие различные СВ ДСИМ друг от друга. Этим признакам соответствуют конфигурационные параметры. Выбор значения того или иного параметра придаёт обобщённой СВ свойств одной из рассмотренных СВ.

1. *Флаг приоритетов.* В каждом блоке внутренней обработки существует свой планировщик. Он специальным образом ранжирует элементы блока и, тем самым, контролирует порядок их обработки. Для некоторых блоков ранжирование предопределено и не играет особой роли: это ББС и БЗС для пассивного ожидания. Для других блоков такие планировщики могут играть важную роль. Если флаг поднят, то к атрибутам события добавляется приоритет, и планировщик может его использовать, в противном случае – приоритеты событий в модели не выделяются.

Заметим, что в общем случае решение проблемы взаимодействия событий, происходящих одновременно, не может быть решено простым введением приоритетов. В обобщённой СВ оно достигается с помощью многократного повторения цикла обработки событий.

2. *Флаг выделения текущих событий.* Если этот флаг поднят, то система разделяет между собой БТС и ББС. В противном случае создаётся только один блок будущих событий, который совмещает в себе свойства обоих блоков. Такое разделение особенно удобно, если события, происходящие одновременно, могут взаимодействовать между собой. В таком случае обычно вводятся приоритеты событий.

3. *Переключатель задержек.* В зависимости от значения переключателя, СВ собирается следующим образом:

- БЗС не выделяются,
- выделяется только БЗС активного ожидания,
- выделяется только БЗС пассивного ожидания,
- выделяются оба подтипа БЗС.

Если БЗС активного ожидания не выделяется, то соответствующие ему события хранятся и обрабатываются в БТС (ББС, если БТС не выделяется).

4. *Переключатель списков событий.* При пассивном ожидании к ресурсам прикрепляются списки событий. Это может быть проделано

несколькими способами:

Каждому ресурсу соответствует единственный список.

Нескольким ресурсам сопоставляется единственный список.

Единственному ресурсу сопоставляется несколько списков.

Объединение двух предыдущих пунктов.

В большинстве систем этот переключатель принимает первое значение.

Общие списки событий могут быть полезны, если на уровне логики модели выделяется несколько ресурсов, но на самом деле эти ресурсы связаны. Кроме того, общие списки позволяют реализовать так называемое гибридное ожидание, представляющее собой некоторый компромисс между рассмотренными видами ожиданий. Несколько списков событий для одного ресурса удобно использовать, если зависимые события нужно ранжировать между собой. Заметим, что того же эффекта можно добиться, используя планировщик и приоритеты событий.

*5. Максимально допустимое число событий в ЛОС ЛО.* Многие системы (в частности, системы [2-5]) используют в своей работе несколько нитей. Операция переключения контекста процесса довольно затратная, поэтому в целях улучшения производительности её стараются производить как можно реже. Рассмотрим механизм обработки события. ЛО получает управление и производит планирование событий, пока прямо или косвенно не будет совершён блокирующий вызов. Запланированные до блокировки ЛО события хранятся в ЛОС ЛО. В целях недопущения чрезмерного разрастания ЛОС ЛО существует специальный параметр-ограничитель. Если число событий в ЛОС достигло предельного значения, то любая попытка запланировать событие приведёт к блокировке ЛО.

В принципе возможна ситуация, в которой модель будет выполняться в потоке одного процесса. Тогда локальное кэширование событий может оказаться неоправданным.

*6. Предел активности ЛО.* В большинстве СВ предел активности ЛО равен числу событий в его ЛОС. Однако некоторые СВ (например, СВ варианта системы ДИАНА для моделирования бортовой сети автомобиля) устроены так, что в обработке участвуют только несколько первых событий ЛОС ЛО. Это позволяет уменьшить число событий в блоках внутренней обработки и в некоторых случаях может увеличить производительность.

ПА в обобщённой СВ может меняться в двух измерениях. Он может иметь общее значение для всей СВ или устанавливаться индивидуально для каждого ЛО. В обоих случаях ПА может быть установлен статически или динамически меняться. В данном контексте «статически» означает, что предел может принимать единственное значение помимо нуля (это значение позволяет заблокировать ЛО). То есть, для конкретного логического объекта предел всегда может быть обнулён, каков бы ни был его тип.

### **3. Выводы**

Как правило, СВ ДСИМ разрабатывается для решения определённого круга задач моделирования. Со временем круг решаемых задач меняется, что

приводит к необходимости вносить в СВ изменения. Один из вариантов решения этой проблемы – разработка универсальной СВ. Как показала практика, этот вариант неудачен, ибо универсальная среда должна 1) заранее учитывать все будущие задачи ИМ, что невозможно и 2) приносит производительность моделирования в ущерб универсальности.

Предлагаемая в данной статье архитектура СВ ДСИМ построена по модульному принципу. Набор модулей разработан с таким расчётом, чтобы при изменении задачи ИМ изменения в СВ касались одного блока. Интерфейс между блоками разработан с таким расчётом, чтобы позволить замену одного логического блока на другой без модификации всех остальных.

Таким образом, для построения СВ с необходимыми свойствами достаточно дописать экземпляр нужного блока. С ростом числа различных экземпляров блоков обобщённой СВ доля повторно используемого кода будет увеличиваться, а затраты на разработку новых СВ сведутся к настройке конфигурации существующей СВ. Это делает возможным быстрое создание высокопроизводительной СВ с нужными свойствами.

### **Литература**

1. *Молонов В.Г., Смелянский Р.Л.* Комплексный подход к моделированию распределённых вычислительных систем. Программирование N.1, 1988 С.57-67

2. *A.Bakhmurov, A.Kapitonova, R.Smeliansky.* "DYANA: An Environment for Embedded System Design and Analysis", 5-th International Conference TACAS'99, Amsterdam, The Netherlands, March 22-28, 1999. Springer (LNCS Vol.1579), pp.390-404

3. *В.В. Балашов, А.Г. Бахмуров, Д.Ю. Волканов, Р.Л. Смелянский, М.В. Чистолитов, Н.В. Юценко, Г.Т. Мамонтов, П.В. Юхта.* Опыт применения программной среды ДИАНА для моделирования и интеграции бортовых вычислительных систем. Рефераты докладов XXVI конференции памяти выдающегося конструктора гироскопических приборов Н.Н. Острякова - С.-Петербург:ЦНИИ «Электроприбор», 2008. – С.60-61.

4. *V.V. Balashov, A.G. Bakhmurov, M.V. Chistolinov, R.L. Smeliansky, D.Yu. Volkanov, N.V. Youshchenko.* A Hardware-in-the-Loop Simulation Environment for Real-Time Systems Development and Architecture Evaluation. In Proc. of the Third International Conference on Dependability of Computer Systems DepCoS-RELCOMEX 2008, Szklarska Poreba, Poland, June 26-28 2008.

5. *Бахмуров А.Г., Егисанетов Э.Г., Новиков О.В., Прус В.В., Савенков К.О., Смелянский Р.Л.* Инструментальная поддержка процесса разработки ПО для спецвычислителей на основе процессора L1879BM1. Методы и средства обработки информации. Труды второй Всероссийской научной конференции. - М.: Издательский отдел факультета вычислительной математики и кибернетики МГУ им. М.В. Ломоносова, 2005. - С.450-456

6. *TJ Schriber, DT Brunner.* "Inside discrete-event simulation software: how it works and why it matters". Proceedings of the Winter Simulation Conference, 2005, pp. 11 pp.+

7. *TJ Schriber, DT Brunner,* "Inside discrete-event simulation software: how it works and why it matters". Proceedings of the Winter Simulation Conference, 1996, pp. 11 pp.+

8. *Sanchez, P.J.* "Fundamentals of simulation modeling". Proceedings of the Winter Simulation Conference, 2007, 9-12 Dec. 2007 Page(s):54 – 62.