

УДК 004.94

ОТОБРАЖЕНИЕ УСЛОВИЙ ДИНАМИКИ ФУНКЦИОНИРОВАНИЯ ОБЪЕКТА УПРАВЛЕНИЯ НА МОДЕЛЬ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

В.К. Погребной, А.В. Погребной, Д.В. Погребной*

Институт «Кибернетический центр» ТПУ

*ООО «Контек-Софт», г. Томск

E-mail: pogrebnoy@tpu.ru

Предложена методика отображения условий динамики функционирования объекта управления на модель системы реального времени, представленную модульной структурой на языке структурного моделирования в форме графа потока данных. Это существенно повысило конструктивность модели и расширило возможности по установлению зависимостей между параметрами динамики модели и характеристиками проектируемой системы. Разработаны формальные методы оценки влияния параметров динамики модели на суммарные объемы потребления вычислительных ресурсов системы – процессорное время, память, пропускная способность сети.

Ключевые слова:

Модель системы реального времени, граф потока данных, модульная структура, прикладная функция, программная нагрузка, процесс, условия динамики, пропускная способность сети.

Key words:

Realtime system model, dataflow graph, modular architecture, applied function, software load, process, dynamic conditions, network bandwidth.

Одним из наиболее важных свойств моделей, используемых при проектировании систем реального времени (СРВ), является их конструктивность [1]. Например широко известные модели в виде сетей Петри не обладают таким свойством, а методы моделирования на основе сетей массового обслуживания и марковских цепей позволяют оценить только частные характеристики систем – производительность и надежность. В разрабатываемой технологии модульного проектирования СРВ на основе активных моделей [2] удается совместить преимущества имитационных моделей с конструктивностью аналитических. Технология включает разработку и визуальное представление на языке структурного моделирования модульной структуры модели, аналитические методы разработки и оптимизации модели, моделирование работы системы на функциональном уровне и в динамике.

В статье рассматривается одна из основных задач первого этапа технологии – отображение условий динамики функционирования объекта управления на модульную структуру модели СРВ и анализ влияния значений параметров динамики на объемы вычислительных ресурсов, необходимых для ее работы. Задача решается в условиях, когда модульная структура модели построена и в соответствии с визуальными возможностями языка структурного моделирования представлена в форме графа потока данных (ГПД). Решение данной задачи повышает конструктивность модели, и как показано в статье, позволяет установить зависимости между параметрами динамики и объемами требуемых вычислительных ресурсов. Эти результаты создают условия для решения задачи второго этапа технологии по доопределению и оптимизации параметров модели, который завершается построением ее исходного варианта, содержащего модель архитектуры и топологии сети вычислительной си-

стемы и согласованную с ней модель программной нагрузки СРВ. Последующие этапы имитационного моделирования СРВ на функциональном уровне и в динамике предполагают преобразование исходного варианта модели в соответствующие активные формы [2] и осуществляются путем выполнения полученных активных моделей на виртуальной машине моделирования.

1. Определение модульной структуры модели СРВ

Модульная структура соответствует начальной стадии модельного представления СРВ и включает две взаимосвязанные части – модульную структуру совокупности прикладных функций (программную нагрузку СРВ), реализующих алгоритмы управления объектом и модульную структуру алгоритма функционирования объекта управления.

Модульная структура объекта управления МС ОУ объединяет имитаторы агрегатов (модули группы А) [3] способные формировать показания датчиков с этих агрегатов в зависимости от текущего состояния агрегата и входных управляющих воздействий, поступающих с программной нагрузки СРВ. При разработке МС ОУ в качестве имитаторов агрегатов могут использоваться их имитационные модели, разработанные непосредственно в ходе проектирования агрегатов. Модули – имитаторы агрегатов оформляются в библиотеку, которая охватывает оборудование конкретной предметной области и при проектировании новых СРВ в данной области пополняется вновь разработанными модулями.

Построение модульной структуры программной нагрузки (МС ПН) осуществляется путем декомпозиции прикладных функций (ПФ) на фрагменты (модули группы F) [3], которые выбираются из библиотеки функциональных модулей или про-

граммируются и оформляются для занесения в библиотеку. Декомпозицию выполняет разработчик проекта, руководствуясь рядом критериев, которые определяются общими представлениями об алгоритмах программной нагрузки и разрабатываемой СРВ в целом. Среди таких критериев выделяются следующие:

- фрагменты ПФ (модули) должны быть функционально законченными с явно выделенными входами и выходами;
- уровень декомпозиции ПФ на модули должен обеспечивать максимально возможное распараллеливание модульной структуры, сопоставимое с числом процессоров (станций) вычислительной системы;
- создание благоприятных условий для возможности использования элементов структурного управления при выполнении модели на виртуальной машине моделирования и в последующем при организации функционирования программного обеспечения СРВ;
- введение составных модулей, объединяющих несколько модулей в один и используемых как средство для улучшения визуального восприятия модульной структуры модели и как возможность распределить модули и данные, входящие в составной модуль, на одну станцию.

Совместная работа МС ОУ и МС ПН важна на этапах моделирования. Это позволяет добиться высокой адекватности модельного представления СРВ и имитации ее функционирования. Что касается рассматриваемых задач отображения условий динамики и анализа их влияния на объемы потребления вычислительных ресурсов, то здесь все решения принимаются относительно МС ПН. Поэтому при решении данных задач мы будем иметь дело с МС ПН, рассматривая ее как визуальное представление в форме ГПД совокупности совместно работающих ПФ.

Каждая ПФ реализует определенную задачу, которую должна выполнять СРВ в соответствии с техническим заданием. На вход ПФ поступает информация с датчиков и других источников, а на выходе формируются управляющие воздействия для исполнительных механизмов агрегатов объекта управления и других потребителей, в том числе и ПФ. В составе модели СРВ алгоритм каждой ПФ, как правило, представляется разбитым на совокупность связанных между собой фрагментов (модулей), образующих модульную структуру. Пример модульной структуры модели программной нагрузки в форме ГПД, содержащей множество модулей (переходов) $F=\{f_m\}$, $m=1,2,\dots,17$ и множество данных (позиций) $D=\{d_i\}$, $i=1,2,\dots,28$, представлен на рисунке. Модель объединяет пять ПФ с множествами модулей (f_2, f_3, f_9) , $(f_1, f_3, f_{10}, f_{14})$, (f_6, f_{11}, f_{13}) , (f_7, f_{13}, f_{16}) , $(f_4, f_8, f_{12}, f_{17})$. При этом модули f_3 и f_9 являются составными (переходы изображены двойными линиями), то есть каждый из них объединяет нес-

колько модулей. Пунктирными линиями на рисунке выделены совокупности модулей и данных, распределенных на станции, номера которых указаны в соответствующих совокупностях. Для данного примера минимальное число станций, необходимое для подключения датчиков и исполнительных механизмов, принято равным шести. Поэтому анализ способности данной совокупности станций выполнить программную нагрузку может привести лишь к необходимости увеличения их числа.

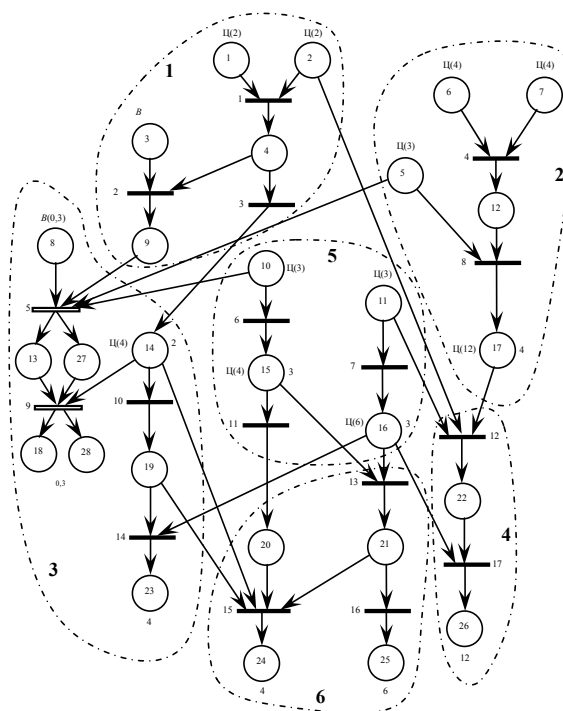


Рисунок. Пример модульной структуры модели программной нагрузки в форме ГПД

2. Динамика совместной работы ПФ

ГПД представим двудольным графом $R=(H,S)$, где H – множество вершин, а S – множество дуг. Множество H включает два непересекающихся подмножества D и F , $D=\{d_q\}$, $q=1,2,\dots,Q$, $F=\{f_m\}$, $m=1,2,\dots,M$. Дуги множества S не могут связывать вершины внутри множеств D или F . Для любой дуги $s \in S$ возможен вариант $s=(d_q, f_m)$, либо $s=(f_m, d_q)$. Вершины $d_q \in D$ и $f_m \in F$ могут быть связаны дугой только в одном направлении. Множество D соответствует множеству данных в ГПД, а множество F – множеству модулей. Множество дуг S соответствует информационным связям – входным, если дуга (d_q, f_m) , и выходным, если дуга (f_m, d_q) . Множество вершин инцидентных только входным дугам обозначим множеством входных данных X^0 , а множество вершин инцидентных только выходным дугам обозначим множеством выходных данных Y^0 .

Множество входных данных X^0 (информация с датчиков и от других источников) в зависимости от условий поступления разбивается на четыре подмножества: X_{II}^0 – циклическое поступление, X_B^0 –

вероятностное поступление, X_d^0 – детерминированная последовательность моментов поступления, X_y^0 – поступление при выполнении заданного условия. Для каждого входа $d_q \in X_d^0$ задается время цикла поступления $\delta_{II}(d_q)$ из множества $\{\delta_{II}\}$. Аналогично входы $d_q \in X_B^0$, $d_q \in X_d^0$, $d_q \in X_y^0$ сопровождаются указанием величин $\delta_B(d_q)$, $\delta_d(d_q)$, $\delta_y(d_q)$ из соответствующих множеств $\{\delta_B\}$, $\{\delta_d\}$, $\{\delta_y\}$. Величины $\delta_B(d_q)$, $\delta_d(d_q)$, $\delta_y(d_q)$ задают максимально допустимые интервалы времени от момента поступления входных данных d_q до момента получения соответствующих выходных данных. Если ограничения на время получения выходных данных (управляющих воздействий) не накладываются, то величины $\delta_B(d_q)$, $\delta_d(d_q)$, $\delta_y(d_q)$ не указываются.

Выходные данные $d_q \in Y^0$ со станций поступают на исполнительные механизмы или устройства отображения информации с временами циклов $\mu_{II}(d_q)$ из множества величин $\{\mu_{II}\}$ для $d_q \in Y_{II}^0$, а для $d_q \in Y_B^0$, $d_q \in Y_d^0$, $d_q \in Y_y^0$ с указанием величин интервалов времени $\mu_B(d_q)$, $\mu_d(d_q)$, $\mu_y(d_q)$ из соответствующих множеств значений $\{\mu_B\}$, $\{\mu_d\}$, $\{\mu_y\}$.

Таким образом, в каждой ПФ входам $d_q \in X^0$ соответствуют величины $\delta(d_q)$, а выходам $d_q \in Y^0$ – величины $\mu(d_q)$. В модульной структуре одной ПФ как правило соблюдается условие $\delta_{II}(d_q) \leq \mu_{II}(d_q)$. Если циклы поступления $\delta_{II}(d_q)$ меньше циклов обновления $\mu_{II}(d_q)$, то исключается возможность представления работы ПФ одним вычислительным процессом. Например, для ПФ с модулями $\{f_1, f_3, f_{10}, f_{14}\}$ $\delta_{II}(d_1) = \delta_{II}(d_2) = 2$ такта, а $\mu_{II}(d_{23}) = 4$ такта, то есть имеет место несогласованность условий поступления входов и обновления выходов. В этом случае разработчик проекта вынужден выполнить согласование, разбивая ПФ на два процесса, работающих с циклами 2 и 4 такта. Разбиение может быть произведено через один из модулей f_3, f_{10}, f_{14} . При необходимости аналогичные преобразования выполняются и для других ПФ. В рассматриваемом примере ГПД согласование потребовалось выполнить в четырех ПФ через модули $f_{10}, f_{11}, f_{13}, f_{12}$. Сведения о полученных при этом процессах приведены в табл. 1.

Таблица 1. Состав и параметры процессов

Процессы	Модули	Входы запуска	Информационные входы	Выходные данные	Циклы запуска и обновления
1	f_1, f_3	d_1^*	$d_1(2), d_2(3)$	d_{14}	2
2	f_{10}, f_{14}	d_{14}^*	$d_{14}(4), d_{16}(6)$	d_{23}	3
3	f_2, f_5, f_9	d_3^*, d_8^*	$d_4, d_5(3), d_{10}(3), d_{14}(4)$	d_{18} d_{28}	0,3 –
4	f_4, f_8	d_6^*	$d_5(3), d_6(4), d_7(4)$	d_{14}	4
5	f_{12}, f_{17}	d_{17}^*	$d_{11}(3), d_{17}(12), d_2(2), d_{16}(6)$	d_{26}	12
6	f_6	d_{10}^*	$d_{10}(3)$	d_{15}	3
7	f_{11}, f_{15}	d_{15}^*	$d_{14}(4), d_{21}, d_{15}(4), d_{19}, d_{14}(4), d_{21}$	d_{24}	4
8	f_7	d_{11}^*	$d_{11}(3)$	d_{16}	3
9	f_{13}, f_{16}	d_{16}^*	$d_{15}(3), d_{16}(6)$	d_{25}	6

В табл. 1 звездочками помечены входные данные, которые задают цикличность запуска процессов. У некоторых данных, например $d_5(3)$ в 4-ом процессе, в скобках указаны их циклы поступления. Эти данные, в отличие от данных со звездочкой, не задают цикличность запуска процессов.

В результате согласования условий поступления входов и обновления выходов модель программной нагрузки СРВ представляется совокупностью параллельных взаимодействующих процессов. В общем случае процесс можно определить как модуль или совокупность взаимодействующих модулей ГПД, имеющих согласованную стратегию запуска, условий поступления входов и обновления выходов и выполняющих функционально законченную операцию за время не превышающее заданную величину. Из этого определения следует, что каждый модуль ГПД может быть объявлен процессом. Но при этом число взаимодействующих динамических объектов (процессов) в модели существенно возрастет. Если каждый процесс, содержащий несколько модулей, свернуть в составной модуль, то модель упрощается, но ее адекватность заметно снижается, т. к. взаимодействия в такой модели рассматриваются не на уровне модулей исходного ГПД, а на уровне составных модулей.

Поэтому в дальнейших исследованиях динамики совместной работы процессов, они рассматриваются как автономные динамические объекты, а взаимодействия между ними, связанные с передачей данных, сохраняются на уровне модулей исходного ГПД независимо от наличия составных модулей и процессов, объединяющих несколько модулей.

3. Анализ влияния условий динамики на потребление вычислительных ресурсов

К основным вычислительным ресурсам, потребляемым при выполнении ГПД, относятся:

- процессорное время, которое может быть затрачено на выполнение модулей ГПД;
- объем памяти в запоминающих устройствах станций;
- пропускная способность сети.

Определение объемов вычислительных ресурсов осуществляется относительно одного цикла моделирования. Величина данного цикла принимается такой, чтобы цикл выполнения любого из процессов ГПД укладывался в цикл моделирования целое число раз. Минимальное значение цикла моделирования определяется как наименьшее общее кратное для значений циклов выполнения процессов.

Анализ требуемого процессорного времени. Цикл выполнения процесса p , $p=1,2,3,\dots,P$, обозначим величиной T_p , а цикл моделирования – величиной T . Тогда общий ресурс процессорного времени для вычислительной системы, содержащей S_0 процессоров (станций), составит $S_0 T$. Частота ρ_m , с кото-

рой модуль f_m выполняется в одном цикле моделирования, определяется из выражения $\rho_m = T/T_p(m)$, где $T_p(m)$ – цикл выполнения процесса p , к которому принадлежит модуль f_m . Время выполнения модуля f_m обозначим величиной τ_m .

В принятых обозначениях необходимым условием проверки способности процессоров совокупности C_0 своевременно выполнить программную нагрузку является следующее:

$$\sum_{m=1}^M \rho_m \tau_m \leq C_0 T. \quad (1)$$

Заметим, что в выражении (1) учитываются все модули f_m , в том числе и те, которые принадлежат процессам, запускаемым по правилам B, D, Y . Для таких модулей величина ρ_m принимается равной 1. При этом соотношение частот между циклически выполняемыми модулями и модулями, которые запускаются по правилам B, D, Y , можно регулировать путем кратного увеличения цикла моделирования.

Если условие (1) не выполняется, то взяв отношение $\left(\sum_{m=1}^M \rho_m \tau_m\right)/T$ и округлив его в большую сторону, получим минимальное число процессоров (станций) C^* необходимое для выполнения программной нагрузки. В этом случае величина $C^* > C_0$, но это не означает, что число станций вычислительной системы непременно будет увеличено до C^* . Сокращение объемов потребления процессорного времени может быть достигнуто, в частности, за счет изменения динамических характеристик и, соответственно, снижения частот ρ_m , а также поиска возможностей по снижению величин τ_m .

В рассматриваемом примере ГПД для $d_q \in X_{II}^0$ циклы поступления $\delta_{II}(d_q)$ заданы из множества $\{\delta_{II}\} = (2, 3, 4, 6, 12)$ и указаны в скобках при записи условий поступления для соответствующих d_q . Множество X_B^0 включает позиции d_3 и d_8 с указанием ограничения на время обработки $\delta_B(d_8) = 0, 3$; $X_D^0 = X_Y^0 = \emptyset$. После выполнения согласований циклов поступления и обновления и формирования процессов времени циклов обновления $\mu_{II}(d_q)$ для выходов $d_q \in Y^0$ образуют множество $\{\mu_{II}\} = \{\delta_{II}\} = (2, 3, 4, 6, 12)$. Для некоторых позиций в результате согласования одновременно указываются величины $\delta_{II}(d_q)$ и $\mu_{II}(d_q)$, например, $\delta_{II}(d_{15}) = 4$, $\mu_{II}(d_{15}) = 3$ или $\delta_{II}(d_{16}) = 6$, $\mu_{II}(d_{16}) = 3$. В этом случае состояния данных в соответствующих позициях накапливаются и потребляются по определенным правилам. Частота ρ_m и время выполнения τ_m модулей f_m приведены в табл. 2.

Таблица 2. Параметры модулей

f_m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ρ_m	6	1	6	3	1	4	4	3	1	3	3	1	2	3	3	2	1
τ_m	0,5	1	1	1,5	0,5	0,5	0,5	2	1,5	2	2	5	2	1	1	2	6

Значения частот ρ_m получены исходя из цикла моделирования, принятого равным величине $T=12$, как наименьшее общее кратное циклов по-

ступления входных данных и циклов обновления выходных.

Для указанных выше значений параметров ГПД условие (1) выполняется ($63,5 \leq 72$) и отношение

$$\sum_{m=1}^M \rho_m \tau_m / T = 63,5/12 \cong 5,3$$

указывает требуемое число станций по данному ресурсу.

Округляя полученное значение в большую сторону до целого значения, получим, что для выполнения ГПД требуется не менее 6 станций. При этом 6 станций обладают некоторым резервом, так как округление произведено в большую сторону. Кроме того, модули, работающие по правилу B без ограничения на время получения данного d_{28} (модули f_2, f_5, f_9), запускаются далеко не в каждом цикле моделирования, а при запуске по правилу B (модули f_5, f_9) с ограничением $\mu_B(d_{18}) = 0,3$ такта предусматривается, что эти модули должны срабатывать значительно быстрее. В данном случае модули f_5, f_9 являются составными (на рисунке изображены двойными линиями) и поэтому получение выхода d_{18} может осуществляться модулями скрытой части ГПД с суммарным временем работы не более 0,3 такта.

Анализ требуемой памяти. Сопоставление объема памяти, требуемого для хранения модулей и данных ГПД, с выделенной для этой цели памятью запоминающих устройств станций, производится при решении задачи распределения модулей и данных по станциям. При этом должно учитываться то обстоятельство, что некоторым позициям соответствует несколько состояний данных, которые нужно хранить одновременно. Число таких состояний определяется разработчиком проекта с учетом динамики работы ГПД. Кроме того, разработчик проекта может принять решение о хранении нескольких копий состояний данного d_q на разных станциях.

Таким образом, общий объем памяти необходимой для хранения всех данных ГПД должен удовлетворять условию:

$$\sum_{q=1}^Q \xi_q c_q p_q \leq C_0 P. \quad (2)$$

Здесь ξ_q – число копий состояний данного d_q ; c_q – число состояний данного d_q , которое необходимо хранить исходя из условий динамики и правил селекции состояний при передаче их между модулями; p_q – объем памяти, занимаемой одним состоянием данного d_q ; P – объем памяти в запоминающем устройстве станции, отведенный для хранения данных.

Заметим, что условие (2) записано для случаев, когда число процессоров C_0 совпадает с числом станций и в каждой из них величина P принята одинаковой.

Анализ требуемой пропускной способности сети. Объем данных, передаваемых по дугам ГПД (d_q, f_m) и (f_m, d_q), вычисляется для одного цикла моделирования T и принимается в качестве весов дуг. Ориентацию дуг (d_q, f_m) или (f_m, d_q) в данном случае можно не

учитывать, поэтому их веса будем определять относительно неупорядоченной пары вершин (d_q, f_m) . Вес r_{qm} дуги, связывающей пару вершин (d_q, f_m) , зависит от размера памяти p_q , занимаемой данным d_q , частоты передачи данного d_q по дуге (d_q, f_m) и числа присоединяемых состояний. Частота передачи данного по дуге (d_q, f_m) , инцидентной модулю f_m , равна частоте выполнения этого модуля ρ_m в цикле моделирования. Правила присоединения состояний и их селекции определяют число состояний c_{qm} данного d_q , передаваемых по дуге (d_q, f_m) . В принятых обозначениях веса r_{qm} определяется по выражению:

$$r_{qm} = a_{qm} c_{qm} p_q \rho_m. \quad (3)$$

Здесь $a_{qm}=1$, если данное d_q инцидентно модулю f_m , $a_{qm}=0$, в противном случае.

Таким образом, общий объем передаваемых по сети данных за один цикл моделирования для w -го варианта распределения модулей и данных по станциям составит величину r_w ,

$$r_w = \sum_{(d_q, f_m) \in S_w} r_{qm}. \quad (4)$$

Множество S_w включает дуги (d_q, f_m) , у которых данные d_q и модули f_m в варианте распределения w оказались на разных станциях.

Если пропускную способность магистрали сети обозначить величиной φ , определяющей объем данных, передаваемых за один такт моделирования, то для успешной работы локальной сети, построенной на базе одной магистрали, должно выполняться условие:

$$(r_w / \varphi) k_\varphi \leq T. \quad (5)$$

Здесь k_φ – коэффициент, учитывающий факторы снижения значения величины φ в реальной сети. Если условие (5) не выполняется, то это означает, что сеть на одной магистрали с данным параме-

тром φ не работоспособна и необходимо принятие решений по уменьшению r_w за счет изменения величин в выражениях (3), (4) или увеличению значения параметра φ .

Число магистралей в сети с неизменным параметром φ ориентировочно можно определить исходя из отношения $(r_w / \varphi) k_\varphi / T$. Полученный результат округляется в большую сторону и принимается как минимально возможное число магистралей в сети. При этом предполагается, что будет выбрана такая структура сети, в которой значительная часть передач данных между станциями сможет осуществляться по разным магистралям параллельно.

Выводы

1. Методика отображения условий динамики функционирования объекта управления на модель системы реального времени, позволяет представить ее в виде совокупности совместно работающих динамических объектов (процессов). Такое представление повышает конструктивность модели и создает возможности для разработки формальных методов, связывающих параметры динамики, заложенные в условия функционирования объекта управления, с параметрами варианта системы, способного успешно осуществлять функции управления данным объектом.
2. Показана возможность разработки формальных методов на примере ряда задач анализа влияния условий динамики на потребление вычислительных ресурсов, реализующих прикладные функции системы. Задачи включают анализ требуемых объемов процессорного времени, памяти запоминающих устройств станций, пропускной способности сети вычислительной системы.
3. Погребной В.К. О построении активных моделей распределенных систем реального времени // Известия Томского политехнического университета. – 2008. – Т. 312. – № 5. – С. 78–84.

Поступила 05.11.2009 г.

СПИСОК ЛИТЕРАТУРЫ

1. Альянах И.Н. Моделирование вычислительных систем. – Л.: Машиностроение, 1988. – 223 с.
2. Погребной В.К. Визуальный уровень представления алгоритмов функционирования распределенных систем реального времени на языке структурного моделирования // Известия Томского политехнического университета. – 2009. – Т. 314. – № 5. – С. 140–146.