Моделирование и анализ информационных систем. Т. 25, №2 (2018), с. 174–192
Modeling and Analysis of Information Systems. Vol. 25, No 2 (2018), pp. 174–192

Сети Петри
и временные автоматы

DOI: 10.18255/1818-1015-2018-2-174-192

УДК 519.7

О корректности моделирования модульных вычислительных систем реального времени с помощью сетей временных автоматов

Глонина А. Б., Балашов В. В.[1]

получена 2 ноября 2017

Аннотация. Рассматривается задача проверки допустимости конфигураций модульных вычислительных систем реального времени (МВС РВ). Конфигурация считается допустимой, если все работы успевают выполниться на МВС РВ в рамках своих директивных интервалов. Предложена обобщенная модель функционирования МВС РВ и метод построения на её основе модели для конкретной конфигурации. Модель представляет собой сеть временных автоматов с остановкой таймеров. По вычислению сети автоматов предлагается строить временную диаграмму (ВД) функционирования МВС РВ, необходимую для проверки допустимости. В работе обосновывается корректность предложенного подхода. Из спецификаций на МВС РВ был выделен ряд требований, применимых к моделям МВС РВ и их компонентов на выбранном уровне абстракции. Модели считаются корректными, если удовлетворяют этим требованиям. Доказано, что если все модели компонентов системы удовлетворяют соответствующим требованиям, то модель МВС РВ, построенная согласно предложенному подходу, удовлетворяет требованиям к модели системы в целом (то есть корректна), а также является детерминированной. Под детерминированностью понимается однозначность построения ВД сети автоматов, соответствующей заданной конфигурации. Это позволяет использовать для проверки допустимости конфигурации любое вычисление соответствующей сети автоматов, что крайне важно для эффективности предложенного подхода, так как число возможных вычислений сети автоматов растет экспоненциально с числом работ в системе. Выполнение требований корректности к моделям компонентов системы может быть проверено автоматически с использованием верификатора и подхода автоматов-наблюдателей. Все разработанные нами модели компонентов системы удовлетворяют соответствующим требованиям, что было доказано с помощью верификатора UPPAAL. Если пользовательские модели компонентов системы удовлетворяют требованиям корректности, то они могут быть включены в модель МВС РВ, которая при этом останется корректной и детерминированной.

Ключевые слова: моделирование, верификация, интегрированная модульная авионика, планирование вычислений

Для цитирования: Глонина А. Б., Балашов В. В., "О корректности моделирования модульных вычислительных систем реального времени с помощью сетей временных автоматов", *Моделирование и анализ информационных систем*, **25**:2 (2018), 174–192.

Об авторах: Глонина Алевтина Борисовна, orcid.org/0000-0001-8716-4128, программист, Московский государственный университет им. М.В. Ломоносова, Ленинские горы, д. 1, г. Москва, 119991 Россия, e-mail: alevtina@lvk.cs.msu.su; Балашов Василий Викторович, orcid.org/0000-0001-5211-805X, ст. науч. сотр., канд. физ.-мат. наук, Московский государственный ун-т им. М.В. Ломоносова, Ленинские горы, д. 1, г. Москва, 119991 Россия, e-mail: hbd@cs.msu.su

Благодарности:
[1] Работа выполнена при финансовой поддержке РФФИ (грант № 17-07-01566).

Введение

В настоящее время одним из перспективных типов архитектур встроенных вычислительных систем реального времени является модульная архитектура. В данной статье в качестве примера модульных вычислительных систем реального времени (МВС РВ) рассматриваются системы интегрированной модульной авионики (ИМА) [1].

МВС РВ, в том числе системы ИМА, состоят из стандартизированных вычислительных модулей, связанных между собой коммутируемой сетью с виртуальными каналами. Каждый модуль содержит набор многоядерных процессоров. В МВС РВ могут входить процессоры различных типов, различающихся производительностью.

Рабочая нагрузка представлена набором разделов. Разделом называется группа логически связанных задач, взаимодействующих посредством общей памяти. Как правило, один раздел соответствует одному приложению. Все задачи являются периодическими: за период должен выполниться один экземпляр задачи, называемый работой. Между задачами с одинаковым периодом могут существовать зависимости по данным: очередная работа задачи-получателя не может начать выполнение до тех пор, пока не получит данные от всех соответствующих работ задач-отправителей. Для каждой задачи задано время выполнения на процессоре каждого типа.

Каждый раздел привязан к вычислительному ядру в составе процессора. Несколько разделов могут быть привязаны к одному и тому же ядру. На интервале планирования для вычислительного ядра задается статическое расписание окон, то есть отрезков времени, в течение каждого из которых могут выполняться работы определенного раздела. Каждый раздел имеет собственный планировщик, управляющий постановкой на выполнение работ внутри окон раздела. Допускается использование разных алгоритмов планирования в разных разделах. У каждой работы существует директивный интервал, определяющийся периодом задачи, а также дополнительными параметрами. Работа должна завершить выполнение не позднее определенной для нее правой границы директивного интервала. Если граница директивного интервала достигается до завершения выполнения работы, то работа считается опоздавшей и не может далее выполняться на вычислительном ядре. Таким образом, время, которое опоздавшая работа находилась на вычислительном ядре, меньше заданного времени, необходимого для ее выполнения.

Конфигурация МВС РВ определяет набор ее модулей, характеристики рабочей нагрузки, привязку разделов к ядрам и расписания окон для каждого ядра. Конфигурация МВС РВ называется допустимой, если для нее в процессе функционирования МВС РВ ни одна из работ не становится опоздавшей. При проектировании МВС РВ анализируется ряд потенциальных конфигураций, для которых необходима проверка допустимости.

Одним из методов проверки допустимости является построение временной диаграммы (ВД) функционирования системы. ВД содержит события постановки на выполнение, вытеснение и завершение работ. Каждое событие характеризуется типом, идентификатором работы-источника и временной меткой. ВД может быть получена в результате прогона имитационной модели МВС РВ. В работе [2] была предложена обобщенная модель функционирования МВС РВ, а также алгоритм построения на

ее основе модели МВС РВ заданной конфигурации. В основе данной обобщенной модели лежит математический аппарат сетей временных автоматов с остановкой таймеров, позволяющий формально проверить выполнение ряда требований к моделям.

В данной статье предложен подход к обоснованию корректности и детерминированности класса моделей, синтезируемых на основе обобщенной формальной модели. С помощью данного подхода доказана корректность и детерминированность моделей, построенных авторами.

1. Обобщенная модель функционирования МВС РВ

1.1. Сети временных автоматов с остановкой таймеров

В работе [2] был сформулирован ряд требований к математическому аппарату для моделирования МВС РВ, и в соответствии с требованиями выбран аппарат сетей временных автоматов с остановкой таймеров [3].

Автомат с остановкой таймеров представляет собой конечный автомат с целочисленными переменными и таймерами. Графически такой автомат может быть представлен в виде размеченного ориентированного графа, вершины которого называются локациями, а дуги — переходами. Таймер — это специальная переменная, принимающая вещественные значения. Таймер считается активным, если его условие активности (булево выражение над множеством переменных) в текущей локации автомата истинно. В противном случае таймер считается остановленным.

Каждой локации сопоставлен инвариант — булево выражение над таймерами и переменными. Автомат может оставаться в данной локации до тех пор, пока инвариант этой локации имеет истинное значение.

Каждый переход характеризуется условием перехода, действиями над переменными и таймерами и, возможно, синхронизацией с другими автоматами. Переход активен, если автомат находится в локации, предшествующей этому переходу, значения переменных и таймеров удовлетворяют условию перехода, и инвариант локации-назначения перехода имеет истинное значение. Переход может (но не обязан) быть совершен, если он активен и может быть выполнена синхронизация. При совершении перехода меняется текущая локация автомата, происходит синхронизация и выполняются действия по изменению переменных и таймеров.

Состоянием автомата называется совокупность его текущей локации, значений переменных и таймеров. Состояние автомата может измениться не только в результате выполнения перехода, но и в результате увеличения значений всех таймеров на одно и то же значение. На рисунке 1 приведен пример автомата, моделирующего планировщик раздела.

Сеть автоматов представляет собой набор автоматов, функционирующих совместно и взаимодействующих посредством общих переменных и каналов. Набор переменных и каналов, посредством которых данный автомат взаимодействует с другими автоматами сети, называется *интерфейсом автомата*.

Канал — средство синхронизации автоматов. Существует два парных действия синхронизации: отправка и прием сигнала по каналу. Если в некоторый момент

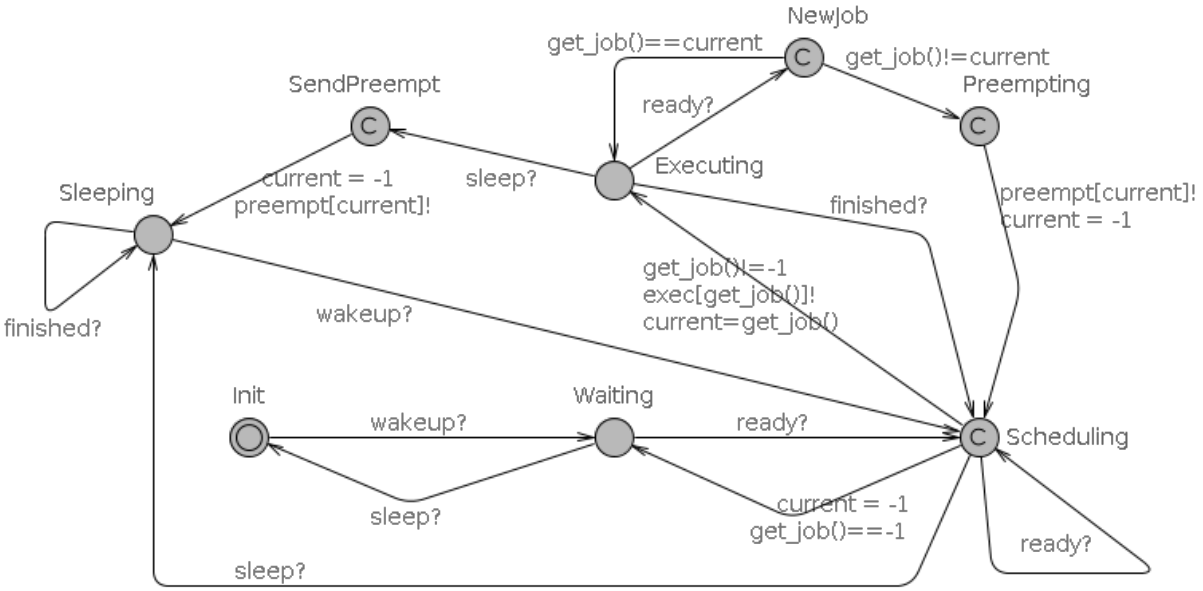

Рис. 1. Пример автомата, моделирующего планировщик раздела
Fig. 1. An automaton example. Partition scheduler model

времени в двух автоматах сети активны переходы с парными действиями синхронизации по одному и тому же каналу, то эти переходы выполняются одновременно. Каналу может быть назначен приоритет. Если в некоторый момент времени в сети автоматов возможны синхронизации по двум различным каналам, то выполняется синхронизация по каналу с высшим приоритетом.

Конечная или бесконечная последовательность состояний, соответствующая некоторому сценарию функционирования сети автоматов, называется *вычислением*.

Для описания предложенной модели первым из соавторов был введен ряд дополнительных определений [2].

Модельное время сети автоматов — значение некоторого служебного таймера, условие активности которого всегда истинно и который никогда не обнуляется.

Событие синхронизации автоматов — это тройка $\langle$ канал, набор автоматов-участников синхронизации, модельное время $\rangle$. *Временная диаграмма сети автоматов* — набор событий синхронизации, соответствующий некоторому вычислению данной сети.

Параметризованный автомат — это автомат, в арифметических и логических выражениях которого наряду с числами и переменными используются параметры с незаданными значениями.

Базовый тип автоматов определяется только интерфейсом. Параметризованный автомат *реализует* базовый тип автоматов, если интерфейс параметризованного автомата содержит все каналы и переменные интерфейса базового типа, а также, возможно, другие каналы и переменные.

Введенные уровни абстракции автоматов соответствуют в рамках предлагаемой схемы моделирования уровням абстракции компонентов МВС РВ. Базовому типу автоматов соответствует базовый тип компонентов. Базовый тип компонента МВС РВ определяет сервис, предоставляемый компонентом системы. Примером базово-

го типа компонента является планировщик раздела. Параметризованному автомату соответствует конкретный тип компонента. Конкретный тип компонента МВС РВ определяет детали реализации базового типа. Примером конкретного типа компонента является планировщик раздела, работающий по алгоритму планирования с фиксированными приоритетами и вытеснением. Экземпляру автомата соответствует экземпляр компонента системы.

Набор базовых типов автоматов назовем *обобщенной сетью автоматов*. Набор параметризованных автоматов — *параметризованной сетью автоматов*.

1.2. Структура обобщенной модели функционирования МВС РВ

В работе [2] А. Глониной предложена обобщенная модель функционирования МВС РВ, представляющая собой обобщенную сеть временных автоматов с остановкой таймеров, состоящую из следующих базовых типов автоматов:

- CS, моделирующий планировщик ядра;
- TS, моделирующий планировщик раздела;
- T, моделирующий функциональную задачу;
- L, моделирующий виртуальный канал (ВК).

Структура предложенной обобщенной сети автоматов приведена на рисунке 2.

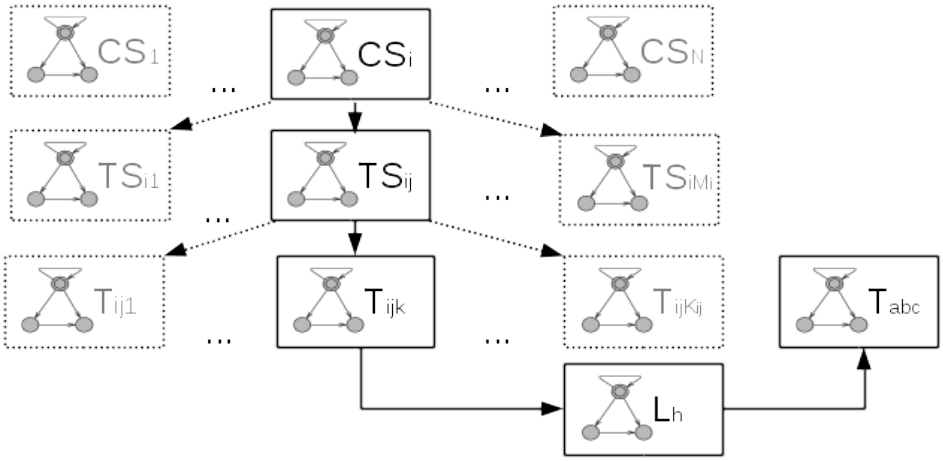

Рис. 2. Структура предложенной обобщенной сети автоматов
Fig. 2. The proposed general stopwatch automata network structure

Полное описание перечисленных базовых типов и сети приведено в [2]. Далее будут рассматриваться временные диаграммы сетей автоматов, представляющие собой наборы событий синхронизации, поэтому укажем каналы синхронизации, входящие в интерфейсы перечисленных базовых типов автоматов (далее под типом будем понимать тип автоматов):

- *exec*, *preempt* — каналы для взаимодействия базовых типов T и TS, синхронизация по которым моделирует соответственно постановку работы задачи на выполнение и ее вытеснение; каждый автомат, реализующий базовый тип T, использует собственные каналы *exec* и *preempt*; автомат, реализующий базовый тип TS, использует по n таких каналов, где n — количество задач раздела, планировщик которого моделируется этим автоматом;

- *ready*, $finished$ — каналы для взаимодействия базовых типов T и TS, синхронизация по которым моделирует соответственно готовность к запуску и завершение работы некоторой задачи раздела; автомат, реализующий базовый тип TS, использует по одному каналу *ready* и $finished$; все автоматы, реализующие базовый тип T и моделирующие задачи одного раздела, используют один и тот же канал *ready* и один и тот же канал $finished$;

- *wakeup*, *sleep* — каналы для взаимодействия базовых типов TS и CS, синхронизация по которым моделирует соответственно открытие и закрытие окна некоторого раздела; каждый автомат, реализующий базовый тип TS, использует собственные каналы *wakeup* и $finished$; автомат, реализующий базовый тип CS, использует по n таких каналов, где n — количество разделов вычислительного ядра, планировщик которого моделируется этим автоматом;

- *send*, *receive* — каналы для взаимодействия базовых типов T и L, синхронизация по которым моделирует соответственно отправку и прием сообщения; каждому автомату, реализующему базовый тип T, соответствует по одному каналу *send* и *receive*; при этом несколько автоматов, реализующих базовый тип L, могут использовать один и тот же канал *send* или *receive* (так моделируется отправка и получение работой задачи сообщений по нескольким виртуальным каналам).

Первым из соавторов была разработана параметризованная сеть автоматов, содержащая реализации перечисленных базовых типов. Эти реализации представляют собой модели планировщика ядра, функциональной задачи и виртуального канала, а также модели трех часто используемых в МВС РВ планировщиков разделов: с фиксированными приоритетами и вытеснением [4], с фиксированным приоритетами без вытеснения [5], планировщик, работающий по алгоритму EDF [6]. Одна из моделей планировщиков раздела приведена на рисунке 1. Также в [2] приведен предложенный А. Глониной алгоритм построения модели МВС РВ с конкретной конфигурацией на основе параметризованной модели по описанию конфигурации. Показано, что для заданных параметризованной модели и конфигурации модель МВС РВ строится однозначно.

Кроме того, в [2] описан разработанный первым автором алгоритм однозначного построения ВД МВС РВ по ВД ее модели: из ВД модели выбираются события синхронизации по каналам *exec*, *preempt* и $finished$, и каждому из выбранных событий однозначно ставится в соответствие событие ВД МВС РВ. Тип события в ВД МВС РВ (постановка на выполнение, вытеснение или завершение работы) определяется каналом синхронизации, идентификатор работы-источника — автоматом-участником синхронизации, временная метка — модельным временем.

Полученная ВД МВС РВ используется для проверки критерия допустимости конфигурации: зная для некоторой работы временные метки её постановок на выполнение, вытеснение и завершение, можно определить суммарное время её выполнения на вычислительном ядре. Если для каждой работы это время равно заданному в конфигурации времени выполнения работы на ядре процессора соответствующего типа, то конфигурация допустима. Иначе, если для некоторой работы суммарное время выполнения на ядре меньше указанного в конфигурации, то работа была снята с ядра по причине достижения границы директивного интервала, и, следовательно, конфигурация недопустима.

2. Корректность проверки допустимости конфигураций МВС РВ

Из раздела 1 следует, что для проверки допустимости некоторой конфигурации достаточно получить соответствующую ей ВД МВС РВ на интервале планирования. Следовательно, для обоснования корректности проверки допустимости достаточно обосновать корректность получаемых согласно предложенному методу ВД МВС РВ.

При проектировании МВС РВ отсутствует возможность экспериментального сравнения ВД, получаемых при прогоне моделей, с ВД функционирования целевых систем. Поэтому корректность всего класса ВД, получаемых согласно предложенному методу, необходимо доказать формально.

Введем понятия корректности модели МВС РВ и корректности ВД модели МВС РВ. Спецификации и стандарты на МВС РВ содержат ряд требований корректности к функционированию системы. ВД моделей содержат события, соответствующие событиям, происходящим в МВС РВ. Из спецификаций МВС РВ с архитектурой ИМА [4, 7] нами были выделены требования, применимые к ВД моделей, то есть представимые в виде ограничений на порядок происходящих в моделях событий и на длительность интервалов между ними. Кроме того, были выделены применимые к ВД моделей требования детерминированности, основанные на спецификациях МВС РВ и, как правило, использующиеся на этапе проектирования (например, в работах [8–11]). Примеры требований приведены ниже, в разделах 2.1.2. и 2.2. ВД модели МВС РВ является корректной, если для нее выполнены все выделенные требования. Модель МВС РВ будем считать корректной, если все возможные ее ВД корректны. Далее под требованием к модели будем понимать требование к ее ВД. Таким образом, модель МВС РВ корректна, если она удовлетворяет всем выделенным требованиям, то есть для всех ее ВД все выделенные требования выполняются.

Напомним, что модель МВС РВ с заданной конфигурацией строится автоматически по описанию этой конфигурации, на основе параметризованной сети автоматов, реализующей обобщенную модель МВС РВ. Обоснование корректности всех моделей, синтезируемых согласно предложенному методу, выполняется следующим образом. Для каждого базового типа автомата, формирующего обобщенную модель МВС РВ (то есть входящего в состав этой модели), на основе спецификаций МВС РВ [4,7,12] выделяется набор требований, которым должны удовлетворять все параметризованные автоматы, реализующие данный базовый тип. Далее доказывается,

что если для всех параметризованных автоматов, формирующих параметризованную модель МВС РВ, выполняются требования, сформулированные для соответствующих базовых типов, то параметризованная модель МВС РВ удовлетворяет всем выделенным требованиям корректности к модели в целом, а также является детерминированной. Под детерминированностью параметризованной модели понимается то, что ее экземпляру, построенному для заданной конфигурации, однозначно соответствует ВД. Детерминированность модели позволяет использовать любое вычисление сети автоматов для построения ВД.

Таким образом, после выполнения указанных доказательств для обоснования корректности ВД некоторой параметризованной сети автоматов, реализующей предложенную обобщенную сеть, достаточно проверить выполнение требований к формирующим её параметризованным автоматам. Проверка выполнения требований к параметризованным автоматам может быть проведена автоматически с помощью верификатора UPPAAL [13]. Для всех построенных автоматов такая проверка была выполнена. Чтобы при добавлении в параметризованную модель нового автомата все ее ВД оставались корректными, достаточно убедиться (с использованием верификатора), что все требования к соответствующему базовому типу автомата для данного автомата выполняются.

2.1. Проверка выполнения требований к моделям компонентов МВС РВ

Для каждого базового типа компонента, которому соответствует базовый тип автоматов, был выделен ряд требований. Для того чтобы любая модель МВС РВ, построенная согласно предложенному методу, была корректной, требуется, чтобы все входящие в нее параметризованные автоматы удовлетворяли требованиям, выделенным для соответствующего базового типа. Кроме того, для каждого конкретного типа компонента могут быть сформулированы дополнительные требования, которым должны удовлетворять параметризованные автоматы, моделирующие этот тип компонента.

2.1.1. Подход на основе автоматов-наблюдателей к верификации параметризованных автоматов

Выделенные требования к параметризованным автоматам представляют собой ограничения на последовательности событий синхронизации и на интервалы между ними. Поэтому для проверки их выполнения был выбран подход на основе автоматов-наблюдателей (observer automata) [14]. Опишем кратко этот подход.

Для проверки некоторого требования к параметризованному автомату строится сеть автоматов, состоящая из данного автомата и автомата-наблюдателя. Автомат-наблюдатель представляет собой автомат, имеющий интерфейс для взаимодействия с исходным автоматом. Во всех локациях автомата-наблюдателя активны переходы с действиями синхронизации, парными ко всем возможным действиям синхронизации исходного автомата. Автомат-наблюдатель имеет некоторую «плохую» локацию, такую что любая не удовлетворяющая требованию последовательность синхронизаций гарантированно приводит автомат-наблюдатель в «плохую» локацию.

Таким образом, проверка выполнения требования сводится к проверке недостижимости этой «плохой» локации. Отметим, что в общем случае задача проверки достижимости в сетях автоматов с остановкой таймеров алгоритмически неразрешима [3], однако в [15] и [16] показано, что для частного случая, соответствующего рассматриваемым в данной работе моделям (детерминированность планировщиков, использование заранее известных оценок времени выполнения работ и времени передачи сообщений), эта проблема разрешима. Кроме того, при использовании верификатора UPPAAL гарантируется [17], что процесс верификации завершается всегда (возможно, с неопределенным результатом). Таким образом, завершение верификации с результатом «плохая локация недостижима» говорит о выполнении требования, завершение верификации с результатом «плохая локация достижима» — о невыполнении требования, завершение верификации с неопределенным результатом — о том, что модель не может быть использована, так как нет подтверждения выполнения для нее требования корректности. В случае завершения верификации с неопределенным результатом, параметризованный автомат необходимо перестроить так, чтобы результат верификации был определен. Согласно [15] и [16], для рассматриваемого класса МВС РВ такое перестроение возможно.

Параметризованный автомат должен удовлетворять требованиям корректности при всех возможных значениях своих параметров. Поэтому в автомате-наблюдателе происходит недетерминированная инициализация параметров исходного автомата, и, таким образом, при верификации рассматриваются все возможные значения параметров из множества допустимых значений. Если хотя бы для одного набора значений параметров «плохая» локация достижима, то исходный автомат считается некорректным.

Кроме того, на порядок событий в исходном автомате могут влиять значения переменных, изменяемых другими автоматами (множество переменных для взаимодействия с другими автоматами определяется интерфейсом соответствующего базового типа автомата). Поэтому в автомате-наблюдателе происходит недетерминированное изменение значений таких переменных. Если хотя бы для одной последовательности значений переменных «плохая» локация достижима, то исходный автомат считается некорректным.

2.1.2. Проверка требований корректности к моделям компонентов МВС РВ

Рассмотрим пример построения автомата-наблюдателя для проверки выполнения приведенного ниже требования к параметризованным автоматам, реализующим базовый тип TS («планировщик раздела»). Один из таких параметризованных автоматов, моделирующий планировщик с фиксированными приоритетами и вытеснением, изображен на рисунке 1.

1. Планировщик раздела может поставить работу на выполнение только в рамках окна этого раздела.

Переформулируем это требование в терминах синхронизаций с участием автомата, моделирующего планировщик раздела. Окнами раздела являются интервалы между синхронизациями по каналам *wakeup* и *sleep*. Постановке на выполнение работы i-й задачи соответствует синхронизация по каналу $exec[i]$. Таким образом,

синхронизации по каналам $exec[i]$ должны происходить только между синхронизациями по каналам $wakeup$ и $sleep$. Последовательности синхронизаций, удовлетворяющие данному требованию, представляют собой циклически повторяющиеся последовательности вида:

1) некоторое (возможно, нулевое) количество синхронизаций по любым каналам, кроме $wakeup$ и $exec[i]$ (события вне окна раздела);

2) синхронизация по каналу $wakeup$ (открытие окна);

3) некоторое (возможно, нулевое) количество синхронизаций по любым каналам (в том числе, возможно, $exec[i]$), кроме $sleep$ (события внутри окна);

4) синхронизация по каналу $sleep$ (закрытие окна).

Один цикл соответствует одному окну раздела.

Любая последовательность синхронизаций с участием модели планировщика раздела, для которой нарушается требование 1, имеет вид:

1) некоторое (возможно, нулевое) количество последовательностей вида, описанного выше (корректная отработка нескольких окон);

2) некоторое (возможно, нулевое) количество синхронизаций по любым каналам, кроме $wakeup$ и $exec[i]$ (события вне очередного окна);

3) синхронизация по каналу $exec[i]$ (постановка некоторой работы на выполнение вне очередного окна);

Автомат-наблюдатель для проверки сформулированного требования приведен на рисунке 3.

При переходах между локациями $Init1$–$Init5$ происходит недетерминированная инициализация параметров автомата: выбирается количество задач (согласно стандарту ARINC 653 [4], раздел не может содержать более 64 задач), для каждой задачи выбирается приоритет (уникальность приоритета обеспечивается функцией $check\_prio$) и правая граница директивного интервала (левая граница не используется в моделях планировщиков). Если для верифицируемого параметризованного автомата известно, что в его системе переходов какие-то из параметров не используются, то соответствующий этап инициализации целесообразно пропустить для ускорения последующей верификации. Например, разработанная первым из соавторов модель планировщика с фиксированными приоритетами не использует границы директивных интервалов задач (за принудительное завершение опоздавших работ отвечают модели задач), а модель планировщика, работающего по стратегии EDF, не использует приоритеты.

Помимо локаций $Init1$–$Init5$, в автомате-наблюдателе присутствуют локации $Inactive$ и $Active$, соответствующие неактивному (вне окон раздела) и активному (внутри окон) состояниям планировщика, а также «плохая» локация $ERROR$. Из локаций $Inactive$ и $Active$ имеются переходы по всем определенным для базового типа TS действиям синхронизации. Также интерфейс базового типа TS содержит набор переменных $is\_ready[i]$. Поэтому из локаций $Inactive$ и $Active$ имеются переходы с недетерминированным изменением этих переменных. Переход из локации

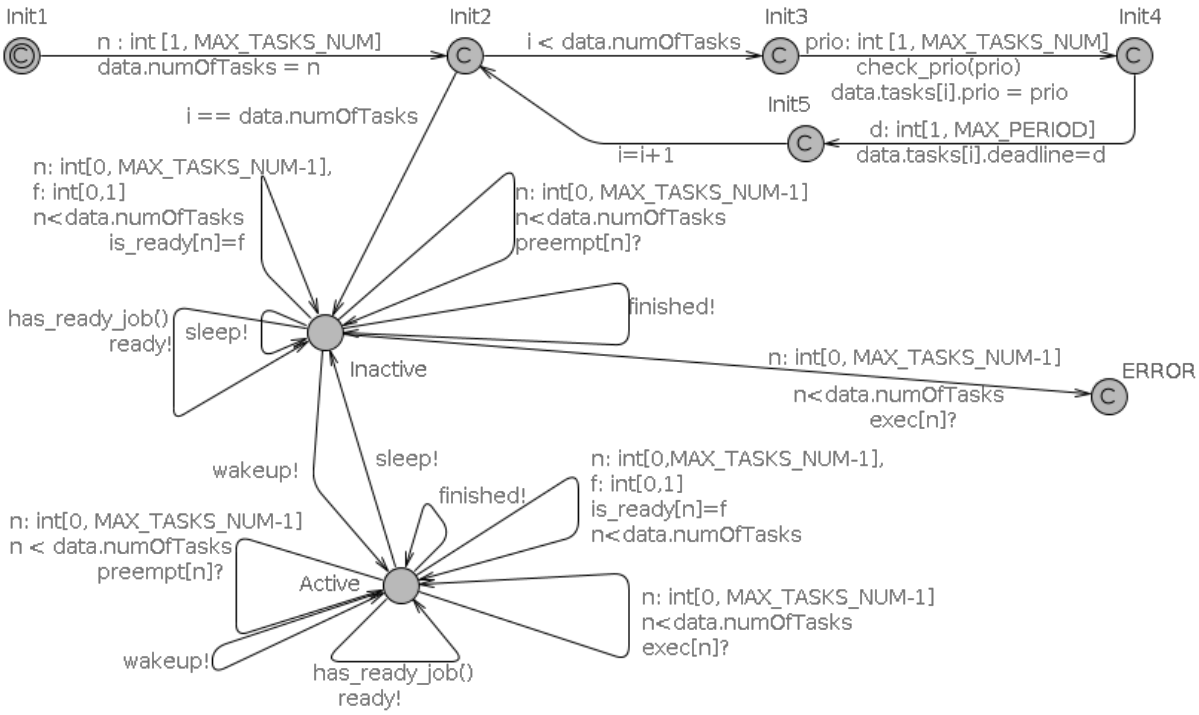

Рис. 3. Пример автомата-наблюдателя
Fig. 3. An observer automaton example

Inactive в локацию *Active* осуществляется при выполнении синхронизации по каналу *wakeup*, из локации *Active* в локацию *Inactive* — при выполнении синхронизации по каналу *sleep*. Если текущей локацией является *Active* и происходит синхронизация по каналу $exec[i]$, то текущей локацией остается *Active*. Если текущей локацией является *Inactive* и происходит синхронизация по каналу $exec[i]$, то это соответствует описанной выше некорректной последовательности синхронизаций и, следовательно, выполняется переход в локацию $ERROR$.

Для каждой из разработанных моделей планировщиков раздела была построена сеть автоматов, состоящая из данной модели (то есть параметризованного автомата) и описанного автомата-наблюдателя. Для всех моделей планировщика раздела для проверки сформулированного требования используется один и тот же автомат-наблюдатель, так как этому требованию должны удовлетворять все реализации базового типа TS. С помощью верификатора UPPAAL было доказано, что локация $ERROR$ во всех построенных сетях автоматов недостижима.

Ниже приведен ряд требований к реализациям базовых типов обобщенной модели и использующихся в дальнейших рассуждениях. Для всех разработанных параметризованных автоматов выполнение соответствующих требований было доказано согласно описанному выше методу.

Требования к реализациям базового типа TS («планировщик раздела»):

2. Для фиксированного состояния очереди готовых работ выбор работы для постановки на выполнение осуществляется однозначно.

3. Если в некоторый момент времени данный раздел активен и появляется одна новая готовая работа, то планировщик обрабатывает это событие мгновенно.

4. Если в некоторый момент времени данный раздел активен и появляется новая готовая работа, то выбор, должна ли она быть поставлена на выполнение, осуществляется однозначно.

5. События открытия и закрытия окон планировщик обрабатывает мгновенно.

6. Планировщик может поставить на выполнение только готовую работу.

7. Работа может быть поставлена на выполнение только вследствие следующих событий: открытие окна раздела, появление новой готовой работы, завершение некоторой работы. При этом постановка на выполнение происходит мгновенно.

8. Если открывается окно раздела и очередь готовых работ раздела не пуста, то на выполнение мгновенно ставится некоторая новая работа.

9. Если завершилась некоторая работа раздела, очередь готовых работ раздела не пуста и в данный момент не происходит закрытие окна раздела, то на выполнение мгновенно ставится некоторая новая работа.

10. Любая работа может быть вытеснена только в двух случаях: либо при закрытии окна данного раздела, либо при появлении новой готовой работы.

11. Если в некоторый момент времени на вычислителе выполняется работа раздела и появляется новая готовая работа, которая должна быть поставлена на выполнение, то текущая работа мгновенно вытесняется.

12. Если в некоторый момент времени на вычислителе выполняется работа раздела и происходит закрытие текущего окна этого раздела, то данная работа мгновенно вытесняется либо завершается.

Требования к реализациям базового типа T («функциональная задача»):

1. Любая работа может быть поставлена на выполнение, только если она является готовой и планировщик раздела об этом оповещен.

2. Очередная работа задачи становится готовой, как только становятся выполненными два условия: модельное время достигает левой границы директивного интервала; получены сообщения от всех соответствующих работ-отправителей (если задача зависит по данным от других задач).

3. Каждая работа задачи должна завершиться, либо когда время ее выполнения на вычислительном ядре достигнет WCET, либо при достижении правой границы своего директивного интервала. Работа не может завершиться, если ни одно из этих двух условий не выполнено.

4. Если задача имеет выходные данные, то каждая ее работа должна послать сообщения через все выходные виртуальные каналы строго в момент завершения работы при условии завершения не позднее правой границы своего директивного интервала.

5. Очередная работа задачи может быть поставлена на выполнение только в рамках своего директивного интервала.

Требования к реализациям базового типа CS («планировщик ядра»):

1. Для любого окна верно, что планировщик ядра оповещает планировщик раздела, соответствующего данному окну, об открытии и закрытии окна строго в моменты, определенные в расписании.

Требования к реализациям базового типа L («виртуальный канал»):

1. Задержка на передачу сообщения через виртуальный канал строго равна значению, указанному в конфигурации системы.

2. Для любого сообщения верно, что получатель оповещается о его получении ровно один раз, в момент окончания передачи.

Помимо приведенных выше требований, также было доказано выполнение порядка десяти требований к реализациям базовых типов автоматов.

Кроме требований, которым должны удовлетворять все реализации соответствующих базовых типов автоматов, существуют требования, специфичные для отдельных параметризованных автоматов (то есть моделей конкретных типов компонентов МВС РВ). В качестве примера одного из таких требований к параметризованным автоматам приведем следующее требование к модели планировщика с фиксированными приоритетами и вытеснением:

На выполнение всегда ставится работа с максимальным приоритетом.

Данное требование специфично для этой реализации планировщика раздела.

На основе требований к конкретным типам компонентов МВС РВ авторами был сформулирован ряд требований, специфичных для реализованных параметризованных автоматов. Их выполнение было доказано с использованием подхода автоматов-наблюдателей. При включении в модель МВС РВ пользовательских параметризованных автоматов к ним также могут быть сформулированы дополнительные требования, проверка выполнения которых осуществляется аналогично.

Отметим, что требования мгновенности реакции на некоторые события можно заменить на требования реакции на соответствующие события с фиксированной задержкой. Например, требование 3 к модели планировщика можно заменить на следующее: «Если в некоторый момент времени данный раздел активен и появляется одна новая готовая работа, то планировщик обрабатывает это событие за N тактов, где значение N задано в конфигурации». Однако в большинстве работ, например в [8–11], данной задержкой пренебрегают, так как работы одного раздела выполняются в общем адресном пространстве и величина задержки мала. Кроме того, модель предполагает, что между окнами разделов могут быть интервалы, не принадлежащие никаким разделам, поэтому такие интервалы могут использоваться для моделирования задержек, необходимых для переключения контекста разделов и инициализации окон.

2.2. Корректность модели в целом

Помимо требований корректности к моделям отдельных компонентов МВС РВ существуют и требования корректности к модели МВС РВ в целом. Проверить выполнение этих требований автоматизированно с помощью верификатора невозможно, так как в общем случае неизвестно количество автоматов в параметризованной сети автоматов, и задача верификации алгоритмически неразрешима [18]. Поэтому выполнение таких требований было доказано нами путем строгих логических рассуждений на основе информации о структуре модели, а также на основе выполнения требований к моделям компонентов МВС РВ.

Приведем пример доказательства выполнения следующего требования:

1. Для любых двух задач верно, что если одна задача зависит по данным от другой, то время начала выполнения каждой работы задачи-получателя данных не меньше времени завершения соответствующей работы задачи-отправителя данных плюс время передачи данных.

Доказательство выполнения данного требования основано на выполнении требований к реализациям базовых типов L и T. Из выполнения требования 2 к модели задачи следует, что время начала выполнения очередной работы не меньше времени получения сообщения от соответствующей работы-отправителя. Из выполнения требования 1 к модели виртуального канала следует, что время получения сообщения равно времени его отправки плюс значение задержки, указанное в конфигурации. Из выполнения требования 4 к модели задачи следует, что отправка сообщения происходит строго в момент завершения работы отправителя. Таким образом, из выполнения перечисленных требований к моделям компонентов следует выполнение указанного требования к модели в целом.

Приведем еще несколько требований к модели в целом:

2. Для любого ядра системы верно, что в каждый момент времени на нем выполняется не более одной работы.

Доказательство выполнения данного требования основано на выполнении требований к реализациям базовых типов CS и TS.

Доказательство выполнения следующих двух требований основано на структуре модели в целом, а именно на информации о приоритетах каналов синхронизации.

3. В момент закрытия окна работа не может быть поставлена на выполнение.

4. Если момент закрытия окна раздела совпадает с моментом завершения выполняющейся на вычислительном ядре работы, то сначала происходит завершение работы, а затем закрытие окна, то есть завершение работы не переносится на начало следующего окна данного раздела.

Нами было доказано выполнение еще семи требований к модели в целом, помимо перечисленных выше.

2.3. Детерминированность модели

Назовем модель МВС РВ с заданной конфигураций (то есть экземпляр сети автоматов) детерминированной, если этому экземпляру сети автоматов однозначно соответствует ее ВД. Докажем детерминированность модели от противного.

Предположим, что модель не является детерминированной, то есть для экземпляра сети автоматов, соответствующей некоторой конфигурации, могут быть получены две разные ВД. Упорядочим события этих двух ВД по времени. Так как события в системе могут происходить одновременно, то одному значению модельного времени соответствует некоторое множество событий. Пусть t_i — наименьшее значение модельного временем, которому в двух ВД соответствуют различные множества событий. Это означает, что как минимум одно событие синхронизации присутствует в множестве событий для этого значения времени одной ВД (будем считать эту ВД первой) и отсутствует в соответствующем множестве другой (будем считать эту ВД второй).

Рассмотрим все возможные варианты каналов данной синхронизации (для краткости далее под «событием X» будем понимать событие синхронизации по каналу X). Покажем, что для каждого канала предположение не верно, то есть если событие синхронизации по каналу присутствует в первой ВД, то оно присутствует и во второй ВД.

1. $wakeup_j$ или $sleep_j$. Из выполнения требования 5 к моделям планировщиков

раздела и требования 1 к моделям планировщиков ядра следует, что полученные ВД соответствуют конфигурациям, различающимся расписаниями окон для ядра j-го раздела. Это противоречит тому, что ВД являются результатом выполнения одной и той же модели.

2. $finished_j$. Возможны два варианта:

   - согласно первой ВД некоторая работа выполнялась на вычислительном ядре ровно WCET единиц времени; до момента t_i ВД совпадают, поэтому из отсутствия во второй ВД события $finished_j$ следует, что работа выполнялась на вычислительном ядре более WCET единиц времени, что противоречит требованию 3 к модели задачи;

   - согласно первой ВД для некоторой работы в момент t_i наступила правая граница директивного интервала; значение этой границы зависит только от номера работы и характеристик соответствующей задачи, поэтому и согласно второй ВД момент t_i является правой границей директивного интервала той же работы; однако, согласно второй ВД, в момент t_i эта работа не завершается, что противоречит требованию 3 к модели задачи.

3. $send_j$. Из выполнения требования 4 к модели задачи следует, что в первой ВД в момент t_i присутствует событие $finished_k$ и t_i не превышает правой границы директивного интервала завершившейся работы. Из рассуждений предыдущего пункта следует, что событие $finished_k$ присутствует и во второй ВД с такой же временной меткой. Следовательно, согласно второй ВД, нарушено требование 4 к модели задачи, поэтому данный вариант не возможен.

4. $receive_j$. Из наличия данного события в первой ВД и выполнения требований 1 и 2 к модели j-го ВК следует, что в первой ВД присутствует событие $send_j$ с временной меткой $t_i - d_j$, где d_j — задержка на передачу сообщения через j-й ВК. Возможны два варианта:

   - $d_j > 0$; следовательно, событие $send_j$ присутствует и во второй ВД, так как до момента t_j ВД совпадают; отсутствие события $receive_j$ в момент t_i в таком случае говорит о невыполнении требований 1 либо 2 к модели виртуального ВК, поэтому данный вариант невозможен;

   - $d_j = 0$; если событие $send_j$ присутствует и во второй ВД в момент t_i, то это свидетельствует о невыполнении требований 1 либо 2 к модели ВК (аналогично варианту $d_j > 0$), что невозможно; следовательно, событие $send_j$ не присутствует во второй ВД, но это противоречит рассуждениям предыдущего пункта и поэтому невозможно.

5. $ready_j$. Событие соответствует тому, что некоторая работа стала готовой к выполнению согласно первой ВД. Из выполнения требования 2 к модели задачи следует, что t_i не меньше левой границы директивного интервала работы и в первой ВД присутствуют события $receive_{k_1}$, ..., $receive_{k_n}$ с временными метками, не большими t_i. Из совпадения ВД до момента t_i и рассуждений

предыдущего пункта следует, что все эти события $receive_{k_1}$, ..., $receive_{k_n}$ присутствуют также и во второй ВД. Таким образом, отсутствие во второй ВД в момент t_i события $ready_j$ свидетельствует о нарушении требования 2 к модели задачи и поэтому невозможно.

6. $exec_j$. Событие соответствует тому, что некоторая работа поставлена на выполнение. Из выполнения требования 3 к модели в целом следует, что t_i не является моментом закрытия окна раздела. Из выполнения требования 6 к модели планировщика раздела и требования 5 к модели задачи следует, что данная работа готова и t_i принадлежит ее директивному интервалу. Из совпадения ВД до момента t_i и из рассуждений предыдущего пункта следует, что согласно обеим ВД на момент t_i есть как минимум одна готовая работа. Из выполнения требования 7 к модели планировщика раздела следует, что имеет место один из трех вариантов:

   - в первой ВД в момент t_i присутствует событие $wakeup_k$. Из рассуждений пункта 1 следует, что во второй ВД также присутствует это событие с такой же временной меткой. Из выполнения требований 2 и 8 к модели планировщика следует, что в момент t_i во второй ВД присутствует событие $exec_j$, что противоречит введенному предположению;

   - в первой ВД в момент t_i присутствует событие $finished_k$. Из рассуждений пункта 2 следует, что во второй ВД также присутствует это событие с такой же временной меткой. Из выполнения требований 2 и 9 к модели планировщика следует, что в момент t_i во второй ВД присутствует событие $exec_j$, что противоречит введенному предположению;

   - в первой ВД в момент t_i присутствует событие $ready_k$. Из рассуждений пункта 5 следует, что во второй ВД также присутствует это событие с такой же временной меткой. Из выполнения требований 2 и 4 к модели планировщика следует, что в момент t_i во второй ВД присутствует событие $exec_j$, что противоречит введенному предположению;

7. $preempt_j$. Событие соответствует вытеснению с вычислительного ядра некоторой работы. Согласно требованиям 10, 11 и 12 к модели планировщика раздела имеет место один из двух вариантов:

   - В первой ВД в момент t_i присутствует событие $sleep_k$. Согласно рассуждениям пункта 1, во второй ВД также присутствует событие $sleep_k$ с той же временной меткой. Из выполнения требования 3 к модели в целом следует, что событие $exec_j$, предшествующее $preempt_j$, имеет временную метку, строго меньшую t_i. Поэтому событие $exec_j$ присутствует и во второй ВД. Следовательно, согласно требованию 12 к модели планировщика раздела, в момент t_i во второй ВД имеется событие $preempt_j$, что приводит к противоречию.

   - В первой ВД в момент t_i присутствует событие $ready_k$. Согласно рассуждениям пункта 5, это же событие присутствует и во второй ВД. Из

выполнения требований 4 и 7 к модели планировщика раздела следует, что во второй ВД в момент t_i присутствует событие $exec_h$. Это же событие присутствует и в первой ВД (см. предыдущий пункт). Из совпадения ВД до момента t_i следует, что согласно обеим ВД на данном вычислительном ядре выполняется одна и та же работа. Поэтому из выполнения требования 11 к модели планировщика следует, что во второй ВД, как и в первой, должно присутствовать событие $preempt_j$, что противоречит предположению.

Таким образом, доказано от противного, что предположение о существовании двух различных ВД, соответствующих одному экземпляру рассматриваемой сети автоматов, ложно. Следовательно, представленная в виде этой сети модель является детерминированной.

Заключение

В работе предложен подход к обоснованию корректности моделей МВС РВ, синтезируемых на основе обобщенной модели, введенной первым из соавторов в [2]. Выбранный для моделирования математический аппарат сетей временных автоматов с остановкой таймеров позволяет проверить с помощью верификатора выполнение ряда требований корректности к параметризованным автоматам, моделирующим отдельные компоненты системы. Для разработанных параметризованных автоматов такая проверка была выполнена. Посредством строгих логических рассуждений, основанных на выполнении требований к моделям компонентов МВС РВ, а также на информации о структуре модели системы в целом, доказано выполнение ряда требований корректности к моделям МВС РВ, построенных согласно предложенному методу, а также их детерминированность. С помощью данного подхода доказана корректность всех моделей, синтезируемых с использованием разработанных параметризованных автоматов, моделирующих компоненты МВС РВ. Также показано, что для сохранения корректности и детерминированности параметризованной модели МВС РВ при включении в неё пользовательской модели компонента системы достаточно проверить выполнение требований корректности к добавляемой модели компонента. Эта проверка производится автоматически с использованием верификатора.

Предложенные методы и средства реализованы программно, интегрированы с САПР планирования вычислений в МВС РВ [8] и успешно апробированы на данных, близких к реальным. Результаты экспериментов приведены в [2].

Отметим, что предложенный подход к построению моделей и доказательству их корректности может быть использован для моделирования вычислительных систем других типов (например, с федеративной архитектурой), что является одним из возможных направлений дальнейших исследований.

Список литературы / References

[1] Watkins C. B., Walter R., "Transitioning from Federated Avionics Architectures to Integrated Modular Avionics", *Proceedings of the 26th IEEE/AIAA Digital Avionics Systems Conference*, 2007, 2.A.1-1–2.A.1-10.

[2] Glonina A. B., Bahmurov A. G., "Stopwatch Automata-Based Model for Efficient Schedulability Analysis of Modular Computer Systems", *Parallel Computing Technologies*, LNCS, **10421**, Springer, 2017, 289–300.

[3] Cassez F., Larsen K, "The Impressive Power of Stopwatches", *CONCUR 2000 — Concurrency Theory*, LNCS, **1877**, Springer, 2000, 138–152.

[4] *Avionics Application Software Standard Interface. Arinc Specification 653*, Aeronautical Radio, 1997.

[5] Marouf M., Sorel Y., "Scheduling Non-Preemptive Hard Real-Time Tasks with Strict Periods", *Proceedings of the 16th IEEE Conference on Emerging Technologies & Factory Automation (ETFA)*, 2011, 1–8.

[6] Mallachiev K. M., Pakulin N. V., Khoroshilov A. V., "Design and Architecture of Real-Time Operating System", *Proceedings of ISP RAS*, **28**:2 (2016), 181–192.

[7] *RTCA: DO-297: Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations*, Radio Technical Commission for Aeronautics, 2005.

[8] Balashov V. V., Balakhanov V. A., Kostenko V. A., "Scheduling of Computational Tasks in Switched Network-Based IMA Systems", *Proceedings of International Conference on Engineering and Applied Sciences Optimization*, 2014, 1001–1014.

[9] Третьяков А., "Автоматизация построения расписаний для периодических систем реального времени", *Труды Института системного программирования РАН*, **22** (2012), 375–400; [Tretyakov A., "Automation of Scheduling for Periodic Real-Time Systems", *Proceedings of ISP RAS*, **22** (2012), 375–400, (in Russian).]

[10] Wang D., Han J., Ma D., Zhao X., "Studying on ARINC653 Partition Run-time Scheduling and Simulation", *Proceedings of World Academy of Science, Engineering and Technology*, 2012, 1583–1587.

[11] Lee Y. H., Kim D., Younis M., Zhou J., "Scheduling Tool and Algorithm for Integrated Modular Avionics Systems", *Proceedings of the 19th IEEE Digital Avionics Systems Conference*, 2000, 1C2/1–1C2/8.

[12] *Aircraft Data Network Part 7. Avionics Full Duplex Switched Ethernet (AFDX) Network*, Aeronautical Radio, 2005.

[13] "UPPAAL Home", http://www.uppaal.org/.

[14] Andre E., "Observer Patterns for Real-Time Systems", *Proceedings of the 18th International Conference on Engineering of Complex Computer Systems*, 2013, 125–134.

[15] Abdeddaim Y., Maler O., "Preemptive Job-Shop Scheduling using Stopwatch Automata", *Tools and Algorithms for the Construction and Analysis of Systems*, LNCS, **2280**, Springer, 2002, 113–126.

[16] Krcal P., Yi W., "Decidable and Undecidable Problems in Schedulability Analysis Using Timed Automata", *Tools and Algorithms for the Construction and Analysis of Systems*, LNCS, **2988**, Springer, 2004, 236–250.

[17] David A., Illum J., Larsen K., Skou A., "Model-based Framework for Schedulability Analysis Using Uppaal 4.1", *Model-Based Design for Embedded Systems*, 2009, 93–119.

[18] Apt K. R., Kozen D. C., "Limits for Automatic Verification of Finite-State Concurrent Systems", *Information Processing Letters*, **22**:6 (1986), 307–309.

Glonina A. B., Balashov V. V., "On the Correctness of Real-Time Modular Computer Systems Modeling with Stopwatch Automata Networks", *Modeling and Analysis of Information Systems*, **25**:2 (2018), 174–192.

DOI: 10.18255/1818-1015-2018-2-174-192

Abstract. In this paper, we consider a schedulability analysis problem for real-time modular computer systems (RT MCS). A system configuration is called schedulable if all the jobs finish within their deadlines. The authors propose a stopwatch automata-based general model of RT MCS operation. A model instance for a given RT MCS configuration is a network of stopwatch automata (NSA) and it can be built automatically using the general model. A system operation trace, which is necessary for checking the schedulability criterion, can be obtained from the corresponding NSA trace. The paper substantiates the correctness of the proposed approach. A set of correctness requirements to models of system components and to the whole system model were derived from RT MCS specifications. The authors proved that if all models of system components satisfy the corresponding requirements, the whole system model built according to the proposed approach satisfies its correctness requirements and is deterministic (i.e. for a given configuration a trace generated by the corresponding model run is uniquely determined). The model determinism implies that any model run can be used for schedulability analysis. This fact is crucial for the approach efficiency, as the number of possible model runs grows exponentially with the number of jobs in a system. Correctness requirements to models of system components models can be checked automatically by a verifier using observer automata approach. The authors proved by using UPPAAL verifier that all the developed models of system components satisfy the corresponding requirements. User-defined models of system components can be also used for system modeling if they satisfy the requirements.

Keywords: modeling, model checking, integrated modular avionics, scheduling

On the authors:
Alevtina B. Glonina, orcid.org/0000-0001-8716-4128, programmer,
Lomonosov Moscow State University,
1 Leninskie Gory, Moscow 119991, Russia, e-mail: alevtina@lvk.cs.msu.su

Vasily V. Balashov, orcid.org/0000-0001-5211-805X, Ph.D. in Mathematics, senior research fellow
Lomonosov Moscow State University,
1 Leninskie Gory, Moscow 119991, Russia, e-mail: hbd@cs.msu.su

Acknowledgments:
This work was supported by RFBR (grant No 17-07-01566).