

**GLOBUS: СРЕДА МОДЕЛИРОВАНИЯ ПРИЛОЖЕНИЙ ОС РЕАЛЬНОГО  
ВРЕМЕНИ****А. Н. Лукичев (Санкт-Петербург)**

Операционные системы реального времени (ОСРВ) чаще всего применяются в области встроженных систем (ВсС). ВсС представляют собой вычислительные системы, предназначенные для управления процессами или объектами и конструктивно объединенные с ними. К системам такого рода часто предъявляются требования реального времени, то есть заданного максимального времени реакции на то или иное изменение состояния объекта, для взаимодействия с которым они предназначены.

Существующие ОСРВ опираются на понятие наихудшего времени выполнения задачи (worst-case execution time, WCET), которое сильно зависит от производительности аппаратуры (требуя от прикладного программиста детального ее знания), влияния других задач и системных сервисов, с которыми взаимодействует задача, подвергаемая оценке. WCET как верхняя граница не запрещает задаче отработать быстрее, передав результаты другой задаче или исполнительному устройству. При применении широко распространенных механизмов межзадачного взаимодействия (семафоры, очереди сообщений, «почтовые ящики»), алгоритмов планирования с вытеснением и приоритетами это приводит к тому, что в совокупности задач моменты их взаимодействия в общем случае непредсказуемы. Хуже того, непредсказуемыми становятся и моменты взаимодействия системы с объектом управления: система может ответить позже или раньше в зависимости от большого числа условий. В результате точное вычисление времени ответа и других системных характеристик становится невозможным. Часто удается найти их верхнюю границу, однако такой анализ трудоемок и должен повторяться каждый раз при внесении изменений в аппаратуру, ОСРВ или в приложение.

Работа является частью исследований, направленных на устранение указанных недостатков, и представляет результаты создания среды имитационного моделирования многозадачных приложений Globus. В Globus применен оригинальный подход к организации вычислений [1], в котором задачи ОСРВ и системные сервисы представлены относительно самостоятельными компонентами вычислительного процесса – «функциональными блоками» (ФБ). Временные характеристики задачи указываются разработчиком, а затем уточняются с учетом сведений о платформе (аппаратура и ОС). Характеристики системных сервисов привязаны к конфигурации аппаратуры и ОС и известны разработчику заранее. Применяемый способ организации вычислений позволяет проводить анализ и вычисление временных характеристик приложения на основе сведений об отдельных задачах [2]. Среда Globus основана на комплексе имитационного моделирования встроженных систем Ptolemy II. После создания и отладки модели приложения в Globus, с помощью разработанных инструментов анализа планируемости (schedulability) и трансляции модель полуавтоматически транслируется в исходный код приложения на языке С для разработанной ОС реального времени Tegra.

Исследования поддержаны грантом правительства Санкт-Петербурга в рамках конкурса субсидий в виде грантов молодым кандидатам наук, проект «Платформа жесткого реального времени для автоматизации в промышленности, робототехнике, нанотехнологиях», заявка МКН/27-05/259.

**Комплекс имитационного моделирования Ptolemy II**

Ptolemy II [3] представляет собой совокупность библиотек, графических сред и вспомогательных программ для моделирования различного рода вычислительных систем, механизмов и процессов, а также для экспериментальной апробации теоретиче-

ских исследований в области их моделирования. Комплекс разрабатывается исследовательской группой Ptolemy Project в университете Калифорнии в Беркли. Лицензия Ptolemy II (BSD) позволяет его свободное использование, модификацию и распространение, а также создание на его основе коммерческих продуктов. Комплекс основан на платформе Java, что обеспечивает его кроссплатформенность, многопоточность, наличие развитой сетевой инфраструктуры (в частности, возможно распределенное моделирование через Internet), взаимодействие моделей с устройствами, подключенными к ПК, поддержку 3D-графики и т.д.

Основное применение Ptolemy II заключается в моделировании вычислений во встроенных системах с учетом различных технологий. Комплекс поддерживает составление моделей различных типов (доменов): моделей динамических систем с непрерывным временем, с дискретными событиями (discrete-event, distributed discrete events, Wireless), синхронных моделей (synchronous-reactive, discrete time, component interaction, Giotto, timed multitasking), конечных автоматов, сетей процессов (Kahn process networks, synchronous dataflow, dynamic dataflow, heterogeneous dataflow, parameterized synchronous dataflow, communicating sequential processes) и т.д.

Ключевой особенностью Ptolemy II является возможность иерархического объединения разнородных моделей и их комплексного моделирования.

Ptolemy II позволяет создавать специализированные «конфигурации» для моделирования систем различных классов. Каждая такая конфигурация представляет собой совокупность базовой инфраструктуры моделирования, необходимых доменов и специализированной графической или текстовой среды и может рассматриваться как среда моделирования систем некоторого класса. Созданная среда моделирования Globus представляет собой одну из таких конфигураций.

### **Организация вычислений в ОСРВ Terra**

Приложение для ОСРВ Terra состоит из совокупности задач, выполняемых параллельно (псевдопараллельно). Задачи могут обмениваться данными друг с другом и с системными сервисами, представляющими собой источники данных от датчиков, модули управления исполнительными устройствами, коммуникационные и другие системные ресурсы. Событие обмена данными может приводить к активизации тех задач (или сервисов), для которых данные предназначены.

Задача Terra моделируется функциональным блоком (ФБ), отдельным компонентом вычислительного процесса, независимо от других преобразующим входные данные и генерирующим результаты за конечное ненулевое время. ФБ имеет два типа входов: информационные и управляющие (рис. 1). Появление данных на управляющем входе активизирует вычисления, в процессе которых ФБ может считать последние данные с одного или нескольких информационных входов и сгенерировать результаты на одном или нескольких выходах. Появление данных на информационном входе не активизирует вычислений. ФБ может не иметь выходов и/или информационных входов, но должен иметь хотя бы один управляющий вход.

Для управляющего входа ФБ должно быть задано время  $\gamma$ , в течение которого любые данные на нем будут гарантированно обработаны. С момента появления данных на этом входе ФБ будет готов принять новые данные через время  $\gamma$ . Для каждого выхода ФБ должно быть задано время  $\omega$ , ограничивающее интервал между генерируемыми на нем данными снизу. Для функционального блока также должна быть задана функция, позволяющая найти вектор текущих интервалов между событиями (генерации данных) на выходах, в зависимости от вектора текущих интервалов между событиями (поступления данных) на управляющих входах, причем эта функция должна быть непрерывной [1, 2]. Эти требования не ограничивают порядок обработки данных на раз-

ных управляющих входах ФБ, они могут обрабатываться с любой степенью параллельности (как крайние случаи – полностью параллельно или последовательно). То есть ФБ может моделировать как программные, так и аппаратные вычисления.

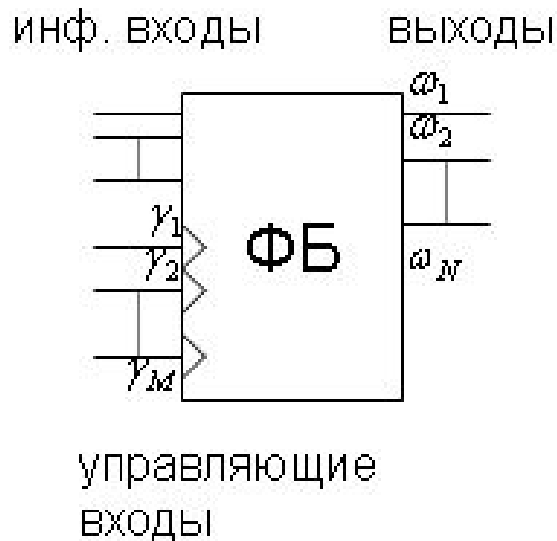


Рис. 1. Функциональный блок и его временные параметры

ФБ в модели [1] похож на функциональные блоки стандарта IEC 61499, но, в отличие от них, меньше ограничивает организацию вычислений «внутри» ФБ и взаимосвязь его входов и выходов [4].

В отсутствие влияния других ФБ, параметр  $\gamma$  управляющего входа представляет собой WCET процедуры, обрабатывающей данные на нем. После разработки и анализа планируемости приложения эта величина уточняется с учетом влияния других ФБ и аппаратной платформы и представляет собой наибольшее время отклика потока. Предельное время завершения (deadline) потока назначается разработчиком исходя из требований к проекту и должно быть не меньше времени его отклика.

Модель приложения представляет собой сеть взаимодействующих ФБ. Каждый выход ФБ в такой сети может быть соединен с одним или несколькими информационными или управляющими входами, но не может быть соединен с другим выходом.

При выполнении всех указанных и ряда дополнительных формальных требований для любой сети ФБ можно вычислить такие системные характеристики, как время ответа (реакции на входные данные) и задержка управления (время между поступлением запроса и генерацией результата вычислений), а также статически (то есть, до проведения имитационного моделирования) проверить возможность блокировок (deadlock) или зависаний (livelock) [2].

### Среда моделирования приложений Globus

Разработанная среда Globus представляет собой конфигурацию комплекса Ptolemy II с поддержкой описанного выше способа организации вычислений. Globus включает в себя графический редактор, стандартную библиотеку функциональных элементов Ptolemy II, средства протоколирования и визуализации (2D/3D) результатов моделирования.

Приложение описывается разработчиком в виде блочной диаграммы, в которой каждый блок представляет собой ФБ: задачу или системный сервис. На рис. 2 приведен пример такой диаграммы. Прямоугольниками, обведенными жирными линиями, обозначены задачи, а тонкими линиями – системные сервисы. Незакрашенные треугольни-

ки есть информационные входы ФБ. Некоторые сервисы могут не иметь на диаграмме управляющих входов (они подразумеваются по умолчанию).

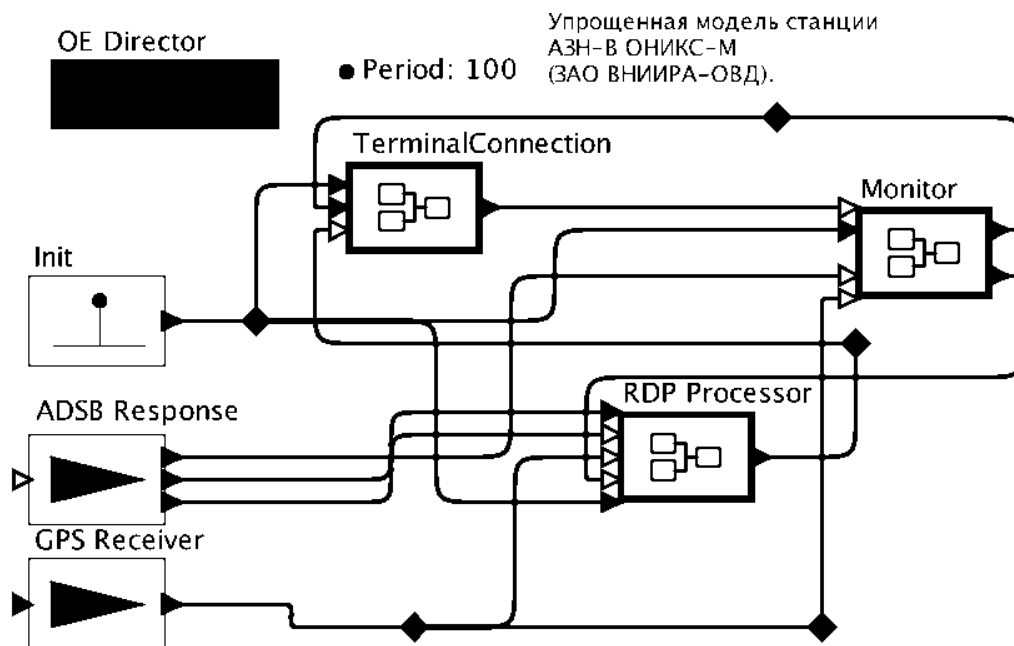


Рис. 2. Пример модели приложения в Globus

Системные сервисы имеют фиксированную функциональность (есть возможность их настройки) и берутся разработчиком из библиотеки Globus. Логика работы задачи может быть описана в виде любой другой модели, поддерживаемой Ptolemy II (дискретно-событийной, конечным автоматом, гибридной и т.д.), либо на языке Java. Помимо системных сервисов, модель может включать вспомогательные элементы (ввод данных, протоколирование, отображение результатов), но не на верхнем уровне иерархии.

ФБ обмениваются элементами данных, называемыми «токенами». Входные порты задач по умолчанию считаются управляющими, но могут быть специально помечены как информационные.

Возможность иерархического объединения разнородных моделей в Globus влечет за собой вопросы сопряжения различных способов организации вычислений между уровнями иерархии. В частности, во «внутренней» модели задачи токены могут считываться одновременно со всех входов (как, например, в синхронно-реактивных моделях), тогда как на верхнем уровне иерархии активизация ФБ произойдет столько раз, сколько токенов поступит на его управляющие входы. Если таких входов несколько, и на них одновременно поступили токены, то при первой активизации они все будут вычитаны «внутренней» моделью, и в следующий раз при активизации не будет доступно ни одного токена, что приведет к ошибке. Кроме того, данные на информационном входе ФБ должны быть доступны для чтения столько раз, сколько это необходимо ФБ и всякий раз при его активизации, что является нестандартным правилом для моделей Ptolemy II. Наконец, многие модели являются синхронными (например, конечные автоматы или синхронно-реактивные модели), то есть производят вычисления и генерируют результаты мгновенно, тогда как ФБ не может этого делать [1] и обязан генерировать данные на выходе строго позже момента активизации вычислений. Для решения этих и некоторых других проблем в Globus разработан специальный механизм транспорта токенов между портами модели, отличный от предусмотренного в Ptolemy II. В частности, даже если данные были сгенерированы на выходе ФБ раньше, чем предусмотрено его атри-

бутом  $\omega$ , для чтения соединенным с ним портам данные будут доступны только по истечении положенного времени.

Другая проблема касается синхронизации модельного времени между уровнями иерархии. «Внутренняя» модель задачи может включать компонент с непрерывным временем (например, описание системной динамики объекта управления), измерять время по тактам (синхронные модели) или не предусматривать его вообще (потокосные модели), тогда как на верхнем уровне время дискретно. Globus пользуется стандартным механизмом управления временем в Ptolemy II: модель верхнего уровня иерархии продвигает время и сообщает его моделям нижних уровней, а при «убегании» непрерывного времени вперед используется откат и пересчет «убежавшей» модели.

Globus подключает библиотеку системных сервисов, соответствующую текущей версии ОСРВ Terra на конкретной аппаратной платформе и содержащую временные характеристики сервисов. В процессе составления модели пользователь задает атрибуты управляющих входов и выходов, соответственно оценивая WCET потоков и накладывая ограничения на интенсивность генерации данных. После этого из среды возможен запуск анализатора планируемости, который уточняет атрибуты входов (с преобразованием их во время отклика) с учетом других потоков и аппаратной платформы. Проверка соблюдения предельного времени отклика в Globus производится автоматически при инициализации имитационного моделирования. Также Globus позволяет проверить возможность блокировок (или зависаний) и локализовать блокируемые задачи [2], оценить времена ответа системы (минимальные задержки на обработку поступившей информации), общую интенсивность возникновения событий в системе (временной масштаб) и локальные интенсивности в участках модели, выделить сильно и слабо связанные участки.

### Выводы

Способ организации вычислений, лежащий в основе ОС реального времени Terra, требует жесткой координации взаимодействия задач и системных сервисов, параметры которой задаются проектировщиком и не изменяются при увеличении производительности аппаратуры или изменении версии операционной системы. Таким образом, приложение будет одинаково взаимодействовать с объектом управления при смене вычислителя или после обновления ОС. Эта процедура гарантированно не приведет к блокировкам (и другим проблемам координации вычислений), независимо от сложности многозадачного приложения.

Разработанная среда моделирования Globus позволяет создавать модели многозадачных приложений для ОС реального времени Terra, проверять их корректность (соблюдение времени отклика задач, возможность планирования) и оценивать системные характеристики.

Создаваемые модели могут иметь произвольное число уровней иерархии и могут быть разнородными, то есть описание логики работы задачи может включать синхронные, потокосные компоненты, конечные автоматы, системную динамику и т.д. При этом решена проблема сопряжения разнородных компонент, частично средствами Ptolemy II, частично оригинальными методами.

Среда моделирования Globus и библиотеки системных сервисов Terra снимают с разработчика приложений требование детально разбираться в особенностях аппаратной и архитектуре программной платформы, предоставляя ему возможность сосредоточиться на требованиях к приложению. Остается только проблема экспертной оценки наибольшего времени выполнения (WCET) алгоритмов, заложенных в задачах, в зависимости от производительности аппаратуры, но эта проблема гораздо легче, так как она не учитывает взаимовлияния задач.

**Литература**

1. **Лукичев А. Н.** Денотативно-объектная модель вычислений для встроенных систем. Автореф... дисс. канд. наук. СПбГУ ИТМО, 2008 (автореферат: [http://fpro.ifmo.ru/file/dis/a\\_68.pdf](http://fpro.ifmo.ru/file/dis/a_68.pdf))
2. **Лукичев А. Н.** Вычисление временных характеристик объектно-событийных моделей встроенных систем// Научно-технический вестник СПбГУ ИТМО. Вып. 46. СПб.: СПбГУ ИТМО, 2008. С. 62–69
3. **Cristopher Brooks, Edward A. Lee, Xiaojun Liu, Stephen Neuendorffer, Yang Zhao, Haiyang Zheng.** Heterogeneous Concurrent Modeling and Design in Java (Volume 1: Introduction to Ptolemy II) / Technical Report UCB/EECS-2007-7. University of California at Berkeley, 2007.
4. **Лукичев А. Н.** Анализ организации вычислительного процесса в стандарте IEC 61499// Научно-технический вестник СПбГУ ИТМО, СПб.: СПбГУ ИТМО. 2009. № 3(61), июнь. С. 68–74.