Динамическая верификация гибридных систем

Николай Пакулин

Институт системного программирования РАН,
Москва, Россия
npak@ispras.ru
http://www.ispras.ru/

Аннотация В последние 10 лет практически повсеместно аналоговые контуры управления вытесняются цифровыми. В настоящее время ведутся работы по созданию цифровых систем управления критическими системами: "интеллектуальные подстанции"в электроэнергетике, "интегрированная модульная авионика"в авиации, "интеллектуальные фабрики"в промышленности и т.д.
Внедрение крупномасштабных систем с цифровыми каналами управления, отказы в которых могут повлечь тяжелые последствия, вплоть до катастрофичесx, требует создания новых методов анализа и верификации, в частности моделирования таких систем и верификации корректности и надежности гибридных систем на моделях.
В статье рассмотрены методы верификации моделей гибридных систем и предложена архитектура тестового стенда для проведения динамической верификации гибридной системы.

Keywords: гибридные системы, киберфизические системы, верификация на моделях, динамическая верификация, тестирование, UniTESK

1 Введение

В последние 10 лет практически повсеместно аналоговые контуры управления вытесняются цифровыми: цифровые каналы связывают сенсоры и актуаторы с ЭВМ, на которых выполняются управляющие программы. Получившиеся системы, состоящие из дискретных и непрерывных процессов получили название *гибридных*. Гибридные системы - это сложные взаимодействующие физические процессы, которые в реальном времени управляются сетью специализированных ЭВМ. Принципиальное отличие гибридных систем от традиционных встроенных систем заключается в масштабах - и объект управления, и управляющая система представляют собой сложные распределенные системы.

По всему миру ведутся работы по созданию цифровых систем управления критическими системами: в авиации разрабатывают и внедряют подход интегрированной модульной авионики (IMA, Integrated Modular Avionics), в электроэнергетике разрабатывают "интеллектуальные подстанции"(Smart Substation), автоматизация промышленных производств приняла новые

формы "интеллектуальных фабрик"(Smart Manufacturing) и т.д. Внедрение таких крупномасштабных гибридных систем, отказы в которых могут повлечь тяжелые последствия, вплоть до катастрофичесх, требует создания новых методов анализа и верификации, в частности моделирования таких систем и верификации корректности и надежности гибридных систем на моделях. Использование реалистичных программно-аппаратных моделей оборудования и коммуникационных сетей открывает следующие уникальные возможности в плане анализа надежности и верификации гибридных систем:

1. "Объем"верификации компонентов гибридной системы в условиях модельных испытаний существенно выше, чем при верификации на реальных объектах, поскольку имеются специальные технические возможности для поддержки дополнительных точек воздействия и точек контроля испытываемых систем.
2. Использование моделей оборудования и сетей позволяет гибко настраивать конфигурацию целевой системы для различных сценариев использования гибридной системы, состава и настроек оборудования, особенностей физических процессов. Настраиваемые модели позволяют верифицировать различные конфигурации объектов в разнообразных режимах функционирования, включая отказы и прочие нештатные ситуации.
3. Динамическая верификация модели гибридной системы позволяет проводить испытания отказоустойчивости и надежности систем в экстремальных режимах, при которых ошибки этих систем могут привести к катастрофическим последствиям. Подобные испытания невозможны на реальном оборудовании в силу разрушительности возможных последствий испытаний.

Без верификации на адекватных моделях анализ функциональной корректности и надежности гибридных систем трудно признать достоверным. Такой анализ может давать некорректные или плохо интерпретируемые результаты, так как отсутствует возможность испытывать целевую систему в условиях, приближенных к реальным. Для получения достоверных выводов о функциональной корректности и информационной безопасности необходимы реалистичные информационные потоки внутри системы, эксплуатационные настройки систем управления, работающие “по-боевому” средства обнаружения аварий и т.д.

Важность проведения всесторонних испытаний гибридных систем хорошо осознают за рубежом. С 2004 года в США функционирует государственный стенд для испытаний систем управления объектами электроэнергетики (National SCADA Test Bed)[1], созданный по инициативе Министерства энергетики США. Стенд активно используется для оценки надежности и защищенности программных комплексов и оборудования различных производителей: Siemens, ABB и других. Однако испытания проводятся на реальном электротехническом оборудовании подстанций тестовой сети Национальной лаборатории Айдахо. С одной стороны, наличие оборудования позволяет

обойти проблему адекватности моделей реальным процессам. С другой стороны, на оборудовании проводится только определенный спектр испытаний, «вблизи» от штатных режимов функционирования оборудования, не содержащих риск разрушения оборудования.

В 2008-2010 гг. в США велись работы по созданию виртуального стенда для силовой электротехнической аппаратуры [2], но опубликованы только промежуточные результаты, окончательные результаты в открытых источниках не представлены. Кроме того, в проекте развивался подход ограниченной виртуализации стенда – фактически, только программы управления и сетевые обмены выполнялись в виртуальном окружении, а объекты управления были реальными физическими приборами.

Стенды, имитирующие реальное оборудование, широко используются в аэрокосмической отрасли. В качестве примеров можно указать проекты по созданию «космических челноков» в США и КК «Буран» в СССР. Модели бортовых систем разрабатываются ведущими авиастроительными компаниями, включая Локхид-Мартин, Боинг и Аэрбас. В России в настоящее время монтируются стенды магистрального самолета нового поколения МС-21. В упомянутых стендах широко используются модели физических процессов. В частности, в ГосНИИАС разработана архитектура смешанного программно-аппаратного стенда, в котором реальное бортовое оборудование получает данные от виртуальной модели самолета через цифровые каналы связи с датчиками.

Несмотря на наличие программно-аппаратных стендов и прогресс в этой области в последние годы, недостаточно разработаны методики проведения испытаний моделей гибридных систем: критерии покрытия тестовых ситуаций, методики перебора тестовых воздействий, критерии оценки корректности поведения системы (тестовые оракулы). Особенность тестирования гибридных систем по сравнению с традиционным тестированием программного обеспечения заключается в том, что в гибридных системах гораздо многообразнее представлены входные воздействия - помимо штатных сигналов, которые сами по себе представлены непрерывными физическими величинами, гибридные системы подвержены шумам в датчиках, отказам телеметрии, резким (в течении миллисекунд) изменениям изменениям окружения (возгорание, взрыв, механическое разрушение и т.п.).

2 Методы моделирования гибридных систем

Методы моделирования гибридных систем можно условно разделить на два класса: методы, используемые в научной среде для разработки средств анализа и верификации гибридных систем, и методы, которые используются в инженерной практике для численного моделирования систем и взаимодействия между ними.

2.1 Гибридные автоматы

В настоящее время в научной среде наиболее широко используются методы, основанные на концепции гибридного автомата [3,4,5] или производные от него. В данной статье мы не будем приводить строгое определение гибридного автомата из-за его громоздкости, ограничимся качественным описанием; читатель может найти математическое определение в указанных статьях.

Говоря неформально, гибридный автомат представляет собой расширенный конечный автомат, в состояниях которого определены режимы, связанные с конечным набором дискретных и вещественных переменных. С каждым режимом определены ограничения, задающие эволюцию системы, а дуги автомата задают дискретные переходы.

В качестве примера рассмотрим термостат, обеспечивающий поддержание температуры в интервале 68-70 градусов. Эта система моделируется автоматом с двумя состояниями ВКЛ и ОТКЛ и одной вещественной переменной T, представляющей температуру. Изменение температуры в режиме ВКЛ может определяться, например, дифференциальным уравнением $\frac{dT}{dt} = k(100 - T)$, где k – константа, параметр модели. Допускается также недетерминированная спецификация режима: например, вместо конкретного дифференциального уравнения режим может задаваться неравенством $k_1 < \frac{dT}{dt} < k_2$, то есть ограничение на динамику вещественной переменной. Исходя из требований к термостату формулируется условие дискретного перехода: при $T > 70$ автомат переходит в состояние ОТКЛ. В этом состоянии динамика температуры описывается отдельным режимом, который может быть никак не связан с режимом в состоянии ВКЛ. Переход из состояния ОТКЛ в состояние ВКЛ осуществляется по достижению ограничения $T < 68$.

Определение гибридного автомата играет в теории гибридных систем ту же базовую роль, какую играют конечные автоматы в теории дискретных систем. Прежде всего, в определении гибридного автомата отсутствуют представления о структурировании системы, декомпозиции её на подавтоматы. На базе гибридных автоматов были разработаны подходы к моделированию составных систем и нотации для записи таких моделей. Можно указать методы иерархического моделирования гибридного поведения Shift [6] и Ptolemy [7], гибридные автоматы ввода-вывода [8], гибридные модули [9], подход к моделированию параллельных составных гибридных систем Charon [10]. Кроме того, развиваются новые направления в математической логике для описания непрерывных процессов, получившие название дифференциальной динамической логики (см. например [11]).

Гибридные автоматы служат основой для различных методов автоматизированного анализа и верификации моделей гибридных систем (подробнее см. раздел 3.1). По этой причине в следующем пункте 2.2 обсуждается вопрос извлечения гибридного автомата из численной модели.

2.2 Методы численного моделирования

Simulink/Stateflow Мировым лидером среди производителей программных продуктов, применяемых в практических задачах моделирования гибридных систем, является компания MathWorks. Её продукты Simulink [12] и Stateflow [13] являются стандартом де-факто для разработки моделей гибридных систем.

Simulink представляет собой среду графического программирования моделей динамических систем. Блоки моделей могут соответствовать отдельным физическим процессам, в частности поддерживается композиция моделей в модели более высокого уровня, а также математическим преобразованиям ($y = f(x)$) и операторам (прежде всего, оператору дифференцирования для графического задания обыкновенных дифференциальных уравнений). Важной отличительной чертой Simulink является интеграция с пакетом программ численного решения уравнений MATLAB. Благодаря этому в Simulink можно импортировать математические модели сложных физических процессов, описанные на MATLAB, и использовать мощные средства численного решения дифференциальных уравнений из MATLAB. На базе Simulink поставляются продукты для моделирования отдельных классов динамических систем, таких как механические системы или электротехнические. Simulink в чистом виде не поддерживает моделирование гибридных систем, так как в нем нет удобных средств описания дискретных, автоматных модулей. Для решения задачи моделирования дискретной составляющей гибридной системы используется продукт Stateflow. Эта программа предоставляет средства для описания конечных автоматов и поддерживает интеграцию с Simulink для спецификации непрерывной части модели. В качестве внутреннего представления моделей Stateflow/Simulink используется язык Matlab. Существуют аналоги Simulink/Stateflow, разработанные по модели открытого кода (open source): пакет Scicos [14], основанный на Scilab [15], открытой альтернативе MATLAB-а

Во второй половине 1990-х годов в Европе был создан новый язык для описания динамических систем, получивший название Modelica [16]. Этот язык объединил лучшие концепции из представленных в то время на рынке европейских пакетов численного моделирования. Modelica представляет собой открытый (non-proprietary), объектно-ориентированный язык, в котором эволюция переменных объекта описываются посредством уравнений. Язык позволяет специфицировать линейные, алгебраические и обыкновенные дифференциальные уравнения. На языке Modelica разработаны множество моделей, поддерживается открытый репозиторий моделей Modelica Standard Library, содержащий порядка 1280 моделей компонентов и 910 функций из различных предметных областей. Язык Modelica предназначен для свободного обмена моделями между различными пакетами численного моделирования. В настоящее время нотация Modelica поддерживается более чем двумя десятками коммерческих и открытых программных продуктов [17].

Несмотря на поддержку автоматных концепций на уровне модели, в настоящее время отсутствуют средства преобразования произвольных моделей Simulink/Stateflow в гибридные автоматы в силу богатства выразительных средств языка MATLAB и несоответствия концептуального представления автоматов в Stateflow дискретным переходам в гибридных автоматах [18,19].

Modelica не содержит отдельных средств для описания дискретных компонентов гибридной системы. Эта задача решается введением булевских или целочисленных переменных в модель. Задача преобразования модели на языке Modelica в гибридный автомат к настоящему времени не решена.

Доменно-специфичные методы моделирования. Унаследованные модели Задачи моделирования динамических систем возникли задолго до появления мощных средств численного моделирования общего назначения, таких как Simulink и Matlab. Изначально такие задачи представлялись как программы на языке программирования общего назначения (Fortran, C, Ada) и выполнялись на рабочих станциях или мэйнфреймах.

Для таких моделей задача построения аналогичного гибридного автомата может быть решена только вручную путем анализа и обратной инженерии программного модуля.

3 Методы верификации гибридных систем

Задача верификации гибридных систем имеет ряд существенных отличий от задачи верификации программного обеспечения общего назначения.

Прежде всего, при верификации необходимо учитывать непрерывный характер физических процессов, протекающих в системе, что требует разработки специальных методов анализа по сравнению с традиционным ПО, в котором внешние события поступают преимущественно в виде дискретных событий.

В силу того, что гибридные системы в подавляющем большинстве представляют собой системы «объект управления – система управления», для них актуальной является задача верификации безопасности (safety verification): верификация того факта, что при функционировании системы не может случиться «что-нибудь плохое».

3.1 Верификация гибридных автоматов

Исследования в области верификации гибридных автоматов идут преимущественно в области верификации безопасности.

При верификации безопасности гибридной системы M задаются множество начальных состояний I и множество «безопасных» состояний S (включая ограничения на переменные состояния, в том числе вещественные). Задача заключается в том, чтобы установить, что при всех возможных эволюциях системы, начинающихся в состояниях из I, система остается в «безопасных» состояниях, либо построить пример достижения «небезопасного» состояния.

Необходимо отметить, что данная задача в общем случае алгоритмически неразрешима [4,20]. Однако для простых случаев временных автоматов (timed state machine) и линейных гибридных автоматов существуют эффективные алгоритмы [21].

Один из подходов к верификации безопасности гибридных систем основывается на методе проверки на моделях (model checking). В рамках этого подхода итеративно вычисляются состояния, достижимые из текущего множества достижимых состояний, и проверяется их безопасность. Ограничения на непрерывные переменные задают криволинейный объем в многомерном пространстве. Так как анализ криволинейных поверхностей в общем случае сложен, большинство инструментов анализа используют линейные неравенства для задания ограничений на безопасные состояния. В этом случае множества допустимых значений переменных оказываются ограничены многогранниками.

Сложность методов, предложенных к настоящему времени, растет экспоненциально с увеличением числа переменных. В настоящее время наиболее совершенным инструментом этого класса считается SpaceEx [22], в котором удалось провести верификацию системы управления вертолетом, в модели которой было 28 вещественных переменных, а динамика описывалась линейными дифференциальными уравнениями [22].

Логический подход к верификации гибридных систем заключается в доказательстве теорем о том, что некоторое свойство ϕ выполняется в начальных состояниях I и во всех состояниях, достижимых из них. Основная проблема в данной области заключается в том, что современные инструменты решения булевских уравнений (SAT solvers) либо вовсе не поддерживают арифметику с вещественными числами, либо поддержка сильно ограничена (например, числа с фиксированной точкой конечной точности). В качестве примера инструмента, позволяющего доказывать безопасность гибридных систем, можно указать KeYmaera [11,23].

Интенсивные исследования ведутся в области абстракции, когда гибридный автомат сводится к более простому [24,25]: ограничения заменяются на линейные неравенства, дифференциальные уравнения $dx/dt = f(x)$ заменяются на дифференциальные неравенства $l \leq dx/dt \leq m$, где l и m – нижняя и верхняя граница f. Для повышения степени приближения поведения упрощенного автомата к поведению исходного автомата, состояния и режимы в абстрактном разбивают на более мелкие. Цель абстрагирования гибридного автомата: получить «похожий» [26] линейный гибридный автомат, для которого есть эффективные средства анализа.

3.2 Тестирование с использованием моделей

Тестирование с использованием моделей (model-based testing) объединяет методы активного изучения поведения целевой системы (тестирования), при котором используются модели тестируемой системы. Модели могут использоваться для генерации тестовых данных, оценки корректности наблюдаемого поведения, оценки полноты тестирования.

Разработаны подходы, использующие гибридные автоматы для тестирования. В частности, инструмент CHARON [27] извлекает из описания системы в виде гибридного автомата тестовые данные и генерирует код для проведения тестирования. Требования безопасности задаются в виде предикатов темпоральной логики. Для темпоральных формул строится монитор, который в динамике верифицирует наблюдаемые трассы. Инструмент использовался как для верификации собственно модели, так и реализации – робота Sony AIBO. Основной недостаток CHARON-а и подобных ему инструментов заключается в том, что модель должна быть разработана в определенном формализме. Инструменты не интероперабельны: нет возможности обмениваться моделями для проведения верификации, всякий раз необходимо переписывать модель под конкретный инструмент.

Верификация численных моделей Системы численного моделирования Simulink/Stateflow не содержат специализированных средств тестирования; методики тестирования численных моделей также не разработаны. В документации на Simulink/Stateflow предлагается использовать компонент SignalBuilder для генерации входных сигналов тестируемого компонента. Однако, при этом отсутствуют средства определения корректности наблюдаемого поведения целевой системы (оракул) и средства оценки полноты достигнутого покрытия тестовых ситуаций.

В компании Rockwell Collins совместно с университетом Иллиноиса разработан набор инструментов [28], которые преобразуют модели Simulink/Stateflow в модели на языке Lustre[29], для которых существуют эффективные методы статического анализа. Разработанная цепочка инструментов использовалась в проектах по верификации бортового авиационного оборудования.

В системе инструментов, основанных на Modelica, нет развитых средств верификации и системного тестирования [30].

Европейский институт стандартизации телекоммуникаций (ETSI) ведет работу по стандартизации расширения языка TTCN3, предназначенного для спецификации тестов для систем в непрерывными (continuous) входами и выходами [31]. Расширение позволяет записывать темпоральные предикаты на ожидаемое поведение системы с непрерывными интерфейсами, а так же задавать генераторы входных данных по сложным законам.

3.3 Системное тестирование

Помимо анализа отдельных модулей и компонентов для гибридных систем особое значение имеет анализ требований безопасности на системном уровне. На текущем уровне развития методов статического анализа модели, полученные в результате композиции моделей отдельных компонентов, не поддаются анализу из-за больших размеров. Единственным практическим средством исследования гибридных систем остается системное тестирование. В системном тестировании требования к системе формулируются

в виде некоторых предельных значений параметров (например, значения ускорений при маневрах) или средних значений (средний расход топлива).

Проведение системных тестов для больших гибридных систем требует построения специализированных тестовых стендов. Пожалуй, наибольшее распространение этот подход получил в авиастроении, где цена отказа системы чрезвычайно велика. Крупные испытательные стенды для бортовой электроники и оборудования получили даже собственное имя «Железные птицы» – Iron Bird. В рамках такого стенда объединяют реальное оборудование, управляющие программы, модели окружения (атмосфера, система позиционирования, каналы связи с наземными службами и т. д.) Часть реализаций авионических компонентов может быть заменена на модели. Общая динамика полета самолета описывается аэродинамической моделью самолета, которая служит моделью системного уровня.

Аналогичные проекты по системному тестированию реализуются также в других предметных областях: электроэнергетике, железнодорожном транспорте, космонавтике.

4 Особенности тестирования гибридных систем

Гибридные системы характерны следующими особенностями:

- Системы тесно связаны функционально. Большинство подсистем нельзя изолировать друг от друга, во время функционирования они взаимодействуют друг с другом прямо или косвенно. Это взаимодействие является существенным для корректного функционирования крупномасштабных систем, связанных посредством непрерывных физических процессов.
- Значительная часть подсистем систем основана на аналоговых физических процессах.
- Существуют требования промышленной безопасности. Условно говоря, "чтобы все работало, и не взрывалось".
- У многих подсистем есть четко выделенные режимы, в которых алгоритмы могут существенно отличаться друг от друга.

Указанные особенности гибридных систем влияют на процесс системного тестирования:

- Для тестирования гибридных систем необходимо моделировать окружение. Изолированное тестирование отдельных модулей выполняется разработчиком, при системной тестировании интересует согласованность работы отдельных систем и их влияние на поведение системы в целом. Особенность заключается в том, что окружение, скорее всего, представляет собой набор физических процессов или явлений.
- Системное тестирование проверяет выполнение интегральных характеристик функционирования системы. Соответственно, требуется верифицировать интегральное поведение совокупности систем (выдерживание курсового угла, скорости относительно воздуха и т.д.), поэтому модель поведения отдельной системы может быть не востребована при её тестировании

- Внешне наблюдаемые параметры меняются непрерывно, требуются средства для спецификации ограничений на непрерывные параметры.
- Для тестирования необходим стенд, которые предоставляет реалистичное окружение системы, включая модели или реализации связанных систем, модель системного уровня в целом, модель окружения и т.д.
- Существенный интерес представляет тестирование целевых систем в условиях отказов или ошибочных данных от связанных систем.

5 Динамическая верификация гибридных систем

В случае гибридной системы подсистемы могут представлять собой дискретные устройства, аналоговые устройства, физические процессы и явления, и гибридные подсистемы, поэтому в общем случае можно считать, что все компоненты гибридной системы являются гибридными подсистемами.

Задача стенда заключается в том, чтобы построить систему взаимодействующих компонентов, каждый из которых является некоей гибридной системой. Построение такого стенда позволяет проводить верификацию сложных систем на ранних этапах разработки, до того, как подсистемы воплощены "в металле". В частности, такой стенд позволит проводить анализ вариантов реализации системы в целом и её отдельных подсистем, проверять реализуемость системных требований и достаточность выделяемых ресурсов.

Модели подсистем могут быть реализованы с использованием различных подходов, специфицированы в различных формализмах и выполняться в разных окружениях. Задача тестового стенда - объединить модели в общую взаимодействующую систему, обеспечить возможность подавать внешние воздействия на получившуюся систему, получить внешне наблюдаемое поведение системы и обеспечить внутренние обмены данными между компонентами системы. При этом стенд может быть распределенным, то есть различные компоненты выполняются на различных компьютерных узлах, и, возможно, под управлением различных операционных систем.

При такой постановке задачи ключевым моментом является подсистема интеграции разнородных моделей в единую систему. Поддерживая интеграцию моделей подсистем в целостную систему, стенд позволяет проводить динамическую верификацию крупной гибридной системы.

“Сердцем” тестового стенда является модель системного уровня. На вход она получает информацию внешне наблюдаемых параметрах (например, тяге двигателя, положении внешних элементов конструкции), состоянии окружения (например, температура среды), и рассчитывает эволюцию системы в целом.

Ручной перебор возможных сценариев воздействия на целевую систему в случае гибридных систем ещё более трудоемкий, чем при тестировании обычных (дискретных) программных систем. Помимо дискретных входных событий такая система подвергается воздействию непрерывных физических процессов через сигналы от датчиков, а кроме дискретных выходных сообщений действует на физические процессы посредством актуаторов.

В данной работе предлагается для автоматизации тестирования гибридных систем использовать подход тестирования с использованием моделей.

Тестирование с использованием моделей (model based testing) в широком смысле охватывает любые способы использования моделей для автоматизации тестирования программных и/или аппаратных систем. Среди задач тестирования, подлежащих автоматизации, мы выделяем следующие:

- спецификация конфигурации тестируемой системы;
- оценка корректности поведения тестируемой системы;
- оценка качества тестирования;
- генерация тестовых воздействий и тестовых последовательностей;
- формирование необходимого окружения тестируемой системы.

Задача спецификации конфигурации системы заключается в задании параметров подсистем и компонентов, описании связей межу ними, и задании начального состояния. Задача оценки корректности поведения заключается в вынесении вердикта является ли наблюдаемое в ходе тестирования поведение тестируемого объекта корректным или нет. Как правило, вердикт выносится на основании проверки соответствия наблюдаемого поведения требованиям к тестируемой системе. В случае, когда эти требования представлены в виде некоторой формальной модели, проверка корректности поведения может выполняться автоматически.

Две последующие задачи напрямую связаны с целеполаганием проводимого тестирования. Примеры целей тестирования могут быть такие:

- проверка того, что тестируемая система соответствует функциональным требованиям;
- формирование заглушек для эмуляции окружения недоступного окружения тестируемой системы
- проверка корректности интеграции тестируемой системы с другими системами или аппаратурой;
- проверка системных требований к комплексу систем;
- проверка обеспечения заданных характеристик при максимальной нагрузки;
- проверка устойчивости к сбоям в аппаратуре и окружении.

Задача оценки качества тестирования заключается в количественной оценке объема проведенного тестирования относительно заданных целей. Как правило, эта задача решается путем введения прямой или опосредованной метрики, в которой происходит оценка. При наличии модели, представляющей требования к системе в целом, определение метрики на основе этой модели является широко распространенной практикой. Альтернативный подход представляют собой метрики на основе покрытия исходного кода тестируемой системы. Еще одним примером метрик, определяемых на основе моделей, являются метрики формируемые на основе выделения классов эквивалентности в модельных пространствах, представляющих в той или иной степени пространство перебора тестовых ситуаций.

Задача генерация тестов состоит в формировании сценария тестирования, обеспечивающего достижения целевого покрытия. Часто эту задачу подразделяют на генерацию единичного тестового воздействия и генерацию тестовой последовательности. В первом случае требуется подобрать параметры для единичного тестового воздействия, в то время как во втором – необходимо сформировать последовательность воздействий для достижения заданной цели. Примером использования моделей для генерации единичных воздействий является автоматическую генерацию значений параметров, попадающий в заданный класс эквивалентности, тогда как генерация тестовых последовательностей часто основывается на обходе графовых структур, FSM, LTS, etc. Генерация тестов на основе моделей может выполняться непосредственно в процессе тестирования, а может происходить заранее. Необходимость формирования окружения тестируемой системы возникает, когда при тестировании часть реального окружения тестируемой системы недоступно и, соответственно, для выполнения тестов требуется его эмуляция, которая может быть реализована посредством тестовых заглушек или симуляции поведения отсутствующих компонентов.

5.1 Архитектура тестового стенда

Как уже упоминалось выше, тестовый стенд объединяет модели подсистем в одну модель системы, а так же модели окружения в одно, целостное окружение. Модели подсистем могут разрабатываться на различных языках, в рамках различных подходов и парадигм, исполняться в различных условиях. Представляется непрактичным требовать от разработчиков моделей придерживаться единой методологии и использовать одни и те же инструменты для моделирования отдельных подсистем. Но требуется обеспечить возможность связать гетерогенные модели в одну систему.

В данной работе используется система обменов, в которой общие данные разделены на именованные переменные, причем в рамках одного вычислительного узла эти переменные разделяются посредством общей памяти (shared memory), а синхронизация значений переменных осуществляется по протоколу UDP через выделенную сеть. Запись в переменную в произвольном компоненте приводит к относительно немедленной передаче нового значения во все остальные компоненты, которые используют эту переменную. Для простоты выбрана тривиальная дисциплина разрешения коллизий: предполагается, что для каждой переменной есть ровно один компонент, который может изменять значение этой переменной. Все остальные компоненты могут только читать. Разработаны адаптеры для доступа к разделяемым переменным из языков C/C++ и Python. Ведется работа по созданию адаптеров для среды численного моделирования SciCos/SciLab.

Состав подсистем, связи между ними, начальное состояние специфицируется посредством языка описания архитектуры AADL [34]. Спецификация AADL должна быть согласована с описанием взаимосвязей между подсистемами в системе обменов данными между компонентами гибридной системы.

Каналы, которые связывают компоненты стенда между собой, представляются в стенде переменными: передача сообщения по каналу в модели AADL равносильна записи этого блока в разделяемую переменную. Нерешенным пока остается вопрос синхронизации модельного времени между компонентами стенда.

Требования системного уровня должны представляться моделью системного уровня. Эта модель используется для вынесения вердикта об интегральном поведении системы. Возможно, не так важно взаимодействие между компонентами, сколько выполнение требований о работе системы в целом. Модель программируется на языке высокого уровня (C или C++), прототипируется на динамическом языке (Python) либо разрабатывается специализированными средствами моделирования, такими как Simulink или SciLab.

Тестирование системы должно проводиться в рамках модели окружения. Так, для самолета такой моделью является атмосфера, а для трансформаторной подстанции - генераторные мощности, ЛЭП и потребители. В предлагаемом подходе окружение представляется в виде гибридной системы, которая замыкает интегральную модель целевой системы. Важно обеспечить согласованность модели окружения, системной модели и моделей компонентов. Под согласованностью понимаются следующие свойства:

- модели датчиков в компонентах целевой системы берут значения показаний (температура, давление, влажность окружающей среды и т.п.) из некоторых разделяемых переменных; соответственно, модель окружения должна записывать значения физических величин в те же самые переменные;
- физические процессы в целевой системе влияют на состояние окружения; например, температура воздуха повышается по мере работы трансформатора – модель окружения должна учитывать эти эффекты; так как обмен информацией между моделями осуществляется через разделяемые переменные, то должны быть предусмотрены соответствующие переменные во моделях, и имена переменных должны совпадать;
- физические процессы в окружении влияют на состояние целевой системы; например, трансформатор нагревается быстрее при повышении температура окружающего воздуха – модель системы должна учитывать эти эффекты; аналогично должны быть предусмотрены переменные для соответствующих параметров окружения;
- физические процессы в окружении влияют на целевую систему в целом: например, боковой ветер сносит самолёт целиком; в модели системного уровня должны быть предусмотрены подобные глобальные эффекты и переменные для получения соответствующих параметров из модели окружения.

Отдельная важная задача - проверка требований промышленной безопасности. Такие требования предполагается специфицировать как формулы темпоральной логики и проверять средствами Data Stream Mining [33].

Переменные, используемые в формулах, берутся из AADL спецификации конфигурации системы.

Задачу определения покрытия в рамках предлагаемой архитектуры предполагается решать как покрытия требований системного уровня в терминах покрытия модели системного уровня, покрытия моделей отдельных подсистем и AADL спецификации композиции подсистем.

Генерация тестовых последовательностей осуществляется средствами библиотеки PyTESK, разработанной в ИСП РАН, которая реализует алгоритмы генерации тестовой последовательности по частично заданному автомату. Специфика применения библиотеки в рамках представленной архитектуры заключается в том, что помимо дискретных событий в спецификацию тестового автомата добавляются переходы, срабатывающие по заданным ограничениям на отсчеты непрерывной физической системы. Другими словами, автомат тестирования представляет собой гибридный автомат.

6 Заключение

Крупномасштабные системы с цифровым управлением непрерывных физических процессов – гибридные или киберфизические системы – все шире используются в промышленности, транспорте, градостроительстве. По мере роста масштабов таких систем возрастает риск аварий и техногенных катастроф. Дальнейшее развитие гибридных систем требует создания метод верификации и валидации таких систем на ранних этапах проектирования.

В статье приведен обзор академических и индустриальных подходов к моделированию и верификации гибридных систем и представлен новый подход к верификации таких систем. Подход объединяет применение гибридных автоматов для моделирования теста и модели системного уровня, и инженерных (численных) моделей для моделирования компонентов системы и окружения.

Численное моделирование используется для описания физических процессов, протекающих в объектах управления, а так же в датчиках и актуаторах. Численные модели должны описывать как воздействие системы на датчики, так и эффекты от воздействий на динамические системы со стороны системы управления через актуаторы. Численные модели используются при вынесении вердикта о корректности поведения гибридной системы: например, перегрев, выгорание предохранителей и т.п. В качестве инструментов предполагается использовать свободно-распространяемые пакеты численного моделирования SciLab или Octave [32].

В данной работе не рассматриваются вопросы построения адекватных моделей физических процессов, протекающих в объектах управления гибридной системы. Мы умышленно ограничениваемся задачей тестирования модели системы в модельном окружении.

Модели событийных интерфейсов гибридной системы описывают преобразование непрерывных данных - показаний датчиков и преобразование управляющих сообщений в действия физических объектов. Традиционные

подходы к моделированию протоколов используют различные формализмы: конечные автоматы, контрактные спецификации, взаимодействующие процессы, системы переходов, темпоральные логики и т.д. В данной НИР предполагается использовать один из этих подходов (или несколько) для описания границы "физический процесс "цифровой протокол". В качестве инструментальных средств предполагается использовать средства библиотеки PyTESK.

В рамках представленного исследования ведутся работы по следующим направлениям:

- интеграция численных моделей физических процессов и дискретных моделей управляющих событий;
- спецификация тестов для гибридных систем;
- прототип архитектуры тестового стенда для проведения динамической верификации гибридной системы.

Работы ведутся при поддержке гранта РФФИ 12-01-31453 мол_а.

Список литературы

1. National SCADA Test Bed Program. http://www.inl.gov/scada/
2. David C. Bergman, Dong Jin, David M. Nicol, and Tim Yardley. The Virtual Power System Testbed and Inter-Testbed Integration. In CSET'09 Proceedings of the 2nd conference on Cyber security experimentation and test, 2009.
3. R. Alur, C. Courcoubetis, T. Henzinger, and P. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In Hybrid Systems, volume LNCS 736, pages 209–229. Springer-Verlag, 1993.
4. R. Alur, C. Courcoubetis, N. Halbwachs, T. Henzinger, P. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. Theoretical Computer Science, 138:3–34, 1995.
5. T. Henzinger. The theory of hybrid automata. In Proceedings of the 11th IEEE Symposium on Logic in Computer Science, pages 278–293, 1996.
6. A. Deshpande, A. Gollu, and P. Varaiya. SHIFT: a formalism and a programming language for dynamic networks of hybrid automata. In Hybrid Systems III, LNCS 1567. Springer, 1996.
7. J. Eker, J. Janneck, E. Lee, J. Liu, X. Liu, J. Luvig, S. Neuendorffer, S. Sachs, and Y. Xiong. Taming heterogeneity–the Ptolemy approach. Proceedings of the IEEE, 91(1):127–144, 2003.
8. N. Lynch, R. Segala, F. Vaandrager, and H. Weinberg. Hybrid I/O automata. In Hybrid Systems III: Verification and Control, LNCS 1066, pages 496–510, 1996.
9. R. Alur and T. Henzinger. Modularity for timed and hybrid systems. In CONCUR '97: Eighth International Conference on Concurrency Theory, LNCS 1243, pages 74–88. Springer-Verlag, 1997.
10. R. Alur, T. Dang, J. Esposito, Y. Hur, F. Ivancic, V. Kumar, I. Lee, P. Mishra, G. Pappas, and O. Sokolsky. Hierarchical modeling and analysis of embedded systems. Proceedings of the IEEE, 91(1), 2003.
11. A. Platzer. Differential dynamic logic for hybrid systems. J. Autom. Reasoning, 41(2):143–189, 2008.

12. Программа численного моделирования непрерывных процессов. http://www.mathworks.com/products/simulink/
13. Программа моделирования дискретных систем. http://www.mathworks.com/products/stateflow/
14. Программа визуального конструирования систем для численного моделирования. http://www.scicos.org/
15. Программа численных расчетов SciLab. http://www.scilab.org/
16. M. Tiller. Introduction to Physical Modeling with Modelica. The Springer International Series in Engineering and Computer Science, Vol. 615, 2001, 346 p.
17. Сайт пакета моделирования Modelica. http://modelica.org/tools
18. G. Karsai, J. Sztipanovits, A. Ledeczi, and T. Bapty. Model-integrated development of embedded software. Proceedings of the IEEE, 91(1):145–164, 2003.
19. A. Agrawal, G. Simon, G. Karsai. Semantic Translation of Simulink/Stateflow models to Hybrid Automata using GReAT. Proceedings of International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT) 2004. Electronic Notes in Theoretical Computer Science (ENTCS), December, 2004.
20. T. Henzinger, P. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata. In Proceedings of the 27th ACM Symposium on Theory of Computing, pages 373–382, 1995.
21. R. Alur, T. Henzinger, and P.-H. Ho. Automatic symbolic verification of embedded systems. IEEE Transactions on Software Engineering, 22(3):181–201, 1996.
22. G. Frehse, C. Le Guernic, A. Donze, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: Scalable verification of hybrid systems. In Proc. 23rd International Conference on Computer Aided Verification (CAV), LNCS 6806, pages 379–395. Springer, 2011.
23. A. Platzer. Logical Analysis of Hybrid Systems - Proving Theorems for Complex Dynamics. Springer, 2010.
24. R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas. Discrete abstractions of hybrid systems. Proceedings of the IEEE, 88(7):971–984, 2000.
25. E. M. Clarke, A. Fehnker, Z. Han, B. H. Krogh, O. Stursberg, and M. Theobald. Verification of hybrid systems based on counterexample-guided abstraction refinement. In Tools and Algorithms for the Construction and Analysis of Systems, 9th International Conference, LNCS2619, pages 192–207, 2003.
26. A. Girard and G. Pappas. Approximation metrics for discrete and continuous systems. IEEE Transactions on Automatic Control, 52(5):782–798, 2007.
27. Li Tan, J. Kim, O. Sokolsky, I. Lee. Model-Based Testing and Monitoring for Hybrid Embedded Systems. Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004, pages 487-492.
28. S. Miller. Bridging the Gap Between Model-Based Development and Model Checking. Lecture Notes in Computer Science Volume 5505, 2009, pp 443-453
29. Halbwachs, N., Caspi, P., Raymond, P., Pilaud, D.: The Synchronous Dataflow Programming Language Lustre. Proceedings of the IEEE 79(9), 1305–1320 (1991)
30. Ingela Lind Henric Andersson. Model Based Systems Engineering for Aircraft Systems – How does Modelica Based Tools Fit? Proceedings 8th Modelica Conference, 2011. pp. 856-864
31. ES 202 786. TTCN-3: Extensions: Support of interfaces with continuous signals. 2012. 45 p.
32. Программа численных расчетов Octave. http://www.gnu.org/software/octave/

33. Babcock B, Babu S, Datar M, Motwani R, Widom J (2002). Models and issues in data stream systems. In Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, PODS '02, pp. 1–16. ACM, New York, NY, USA.
34. Feiler P. H., Gluch D. P., Hudak J. J. The architecture analysis & design language (AADL): An introduction. – CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 2006. – №. CMU/SEI-2006-TN-011.