ISSN 0868–5886 *НАУЧНОЕ ПРИБОРОСТРОЕНИЕ, 2011, том 21, № 3, с. 125–129*

ОБРАБОТКА И ПРЕДСТАВЛЕНИЕ ДАННЫХ

УДК 577.152.311-82

© Н. В. Краснов, А. В. Небылов, А. В. Самокиш

МЕТОД ФОРМАЛИЗАЦИИ ОПИСАНИЯ ПОВЕДЕНИЯ СИСТЕМ УПРАВЛЕНИЯ В ХОДЕ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Рассмотрен метод формализации исходных данных, описывающих поведение систем управления (СУ), путем получения структуры Крипке и набора темпоральных формул на основе набора прецедентов работы системы. Процесс формализации рассмотрен на примере описания работы автоматизированного анализатора уровня активности холинэстераз плазмы и эритроцитов крови человека "Гранат-4". Полученное описание может служить основой как для реализации программного обеспечения СУ, так и для анализа и исследования особенностей работы СУ с помощью методов Model Checking, формальных методов верификации или средств построения и контроля сценариев проверки.

Кл. сл.: формализация, верификация, структура Крипке, темпоральная логика, холинэстераза, система переходов, прецедент

ВВЕДЕНИЕ

Сложность проектирования современных систем управления (СУ) постоянно растет. Современный этап развития в области проектирования систем управления, в том числе программного обеспечения (ПО) СУ сложных аналитических комплексов, характеризуется тем, что ПО начинает вносить основной вклад в стоимость разработки. Процесс создания ПО СУ требует использования методов, позволяющих получить готовый результат, отвечающий ряду жестких требований.

Ошибки в ПО СУ и микропроцессоров могут привести к дорогостоящему затягиванию этапов динамической комплексной отладки и испытаний, а также к неожиданным отказам системы во время эксплуатации [1]. Такие ошибки обусловлены, прежде всего, логической сложностью комплекса программ. Верификация ПО дает возможность обнаруживать ошибки на ранних стадиях разработки, что позволяет считать ее основным методом повышения качества программных систем.

Формализация спецификации является важным элементом подготовки проведения верификации ПО СУ. Форма представления требований, на которых будет строиться спецификация, должна обеспечивать возможность их автоматического анализа. В связи с этим в процессе определения требований необходимо использовать формальные методы описания. Однако в настоящее время переход к формальному описанию вызывает значительные затруднения у большинства разработчиков. Изложение функциональных требований к программному обеспечению, как правило, осуществляется в повествовательной форме. Требования технического задания на разработку ПО СУ носят декларативный характер и могут рассматриваться как перечень решаемых задач, логика реализации которых должна быть изложена в дополнительных документах. Такая форма представления значительно затрудняет анализ требований [2]. Таким образом, на текущем этапе развития средств проектирования СУ задача формализации исходных данных является крайне актуальной.

ПОСТАНОВКА ЗАДАЧИ

В настоящей работе представлен метод формализации исходных данных с помощью формул темпоральной логики. Особенности применения метода проиллюстрированы на примере разработки ПО СУ для прибора "Гранат-4".

Прибор "Гранат-4" разрабатывался как автономный автоматизированный анализатор уровня активности холинэстераз плазмы и эритроцитов крови человека (каждая отдельно) для контроля безопасности условий работы персонала в условиях возможности воздействия антихолинэстеразных веществ. Принцип действия прибора состоит в фотометрическом измерении кинетических параметров ферментативных реакций, катализируемых холинэстеразами крови в растворах, приготовленных по методике проб, в течение времени, заданного программой прибора, с последующим расчетом характеристик, отнесенных к стандартным условиям, откорректированных на температуру в условиях измерения (температура контролируется встроенным датчиком).

Алгоритм работы прибора (т. е. его ПО) должен автоматически обеспечивать контроль исправности каждый раз при включении прибора, измерение характеристических параметров, расчет характеристических величин, отнесенных к стандартным условиям, занесение рассчитанных значений в память (для заданного количества проб) и автоматически исключать возможность неправильного заполнения и переполнения памяти. Под управлением с ручной 3-кнопочной клавиатуры прибор должен обеспечивать возможности выбора режима работы согласно типу пробы (плазма или кровь), изменения, при необходимости, номера пробы (автоматически должен быть предложен следующий, по нарастанию, свободный номер), выбор режима работы по вариантам: работа, просмотр, выключение. При установке режимов должна быть предусмотрена возможность сброса неправильных установок по выбору режима. Результаты измерений проб типа плазма—кровь для каждого пациента должны быть занесены в память (с одинаковыми номерами) таким образом, чтобы при их обработке (с использованием установленных в память калибровочных данных) получались значения характеристических параметров, отнесенных к стандартным условиям для конкретного пациента.

МЕТОД ФОРМАЛИЗАЦИИ

В предлагаемом методе основой описания логики работы ПО СУ является UML-формализм "прецедент" (use case). Прецедент описывает некоторый целостный фрагмент поведения системы не вдаваясь при этом в особенности внутренней структуры объекта. Набор прецедентов должен как можно более полно описывать поведение проектируемой системы управления. В предлагаемом подходе прецедент описывается с помощью условных начального и конечного состояний системы и условия перехода между ними. Следует отметить, что определение прецедентов производится вручную, все остальные шаги (включая описание структуры Крипке и темпоральных формул) выполняются автоматически в рамках средства проектирования СУ arKItect®, разрабатываемого в Международном институте передовых аэрокосмических технологий Санкт-Петербургского государственного университета аэрокосмического приборостроения.

В результате анализа требований к работе аналитического прибора "Гранат-4" были описаны следующие прецеденты, определяющие логику работы:

1. Если прибор был выключен, то он переходит в режим проверки.
2. Если проверка не была пройдена, прибор переходит в состояние "Неисправен".
3. Если проверка прошла успешно, прибор переходит в основной режим работы.
4. Из основного меню можно перейти в режимы:
   – просмотра результатов,
   – режим измерения параметра плазмы,
   – режим измерения параметра крови,
   – выключить прибор.
5. Если в режиме просмотра результатов доступны измерения параметров и крови, и плазмы, то производится расчет общего коэффициента.
6. Из режима измерения параметра прибор переходит в режим расчета параметра и далее в режим индикации результата.
7. По окончании измерений прибор переходит в состояние выбора режима работы.
8. В режимах измерения параметров проводится проверка возможности начать новое измерение, которая в случае отсутствия свободных ячеек памяти переводит прибор в режим просмотра результатов.

Описание представленных условий работы с помощью прецедентов определяет состояния системы и переходов между ними, далее этот набор автоматически объединяется и представляется в виде системы переходов ST (transition system). В ходе описания прецедентов работы прибора "Гранат-4" была построена система переходов со следующим набором состояний.

S_0: Прибор выключен
S_1: Проверка работоспособности
S_2: Прибор неисправен
S_3: Прибор в состоянии выбора режима работы
S_4: Прибор в состоянии проверки результатов работы
S_5: Просмотр результатов
S_6: Проверка состояния памяти результатов вычисления параметра плазмы
S_7: Измерение параметра плазмы
S_8: Расчет параметра плазмы с учетом температуры воздуха
S_9: Индикация рассчитанного параметра плазмы
S_{10}: Проверка состояния памяти результатов вычисления параметра крови
S_{11}: Измерение параметра крови
S_{12}: Расчет параметра крови с учетом температуры воздуха
S_{13}: Индикация рассчитанного параметра плазмы

Для дальнейшей формализации необходимо построить структуру Крипке, с помощью которой будут формализованы требования к СУ посредством формул темпоральной логики. Обычно [3] структура Крипке, соответствующая системе пе-

реходов, строится путем отбрасывания действий на переходах. Это делается по той причине, что в некоторых случаях проверка требований к системе производится с игнорированием конкретных действий окружения, которые ведут к нарушению этих требований.

В рассматриваемом случае ставится задача максимально полного описания системы, и, таким образом, действия окружения следует ввести в структуру Крипке. Предлагается следующий подход к созданию структуры Крипке-М по имеющейся системе переходов ST.

$S_0{}^{\mathrm{M}} = S_0{}^{\mathrm{ST}}$ — множество начальных состояний структуры Крипке соответствует множеству начальных состояний системы переходов;

$S^{\mathrm{M}} = S^{\mathrm{ST}} + R^{\mathrm{ST}}$ — множество состояний структуры Крипке строится следующим образом: для каждого состояния системы переходов S^{ST} создается состояние структуры Крипке S^{M} и для каждого перехода системы переходов R^{ST} создается состояние структуры Крипке S^{M};

R_{M} — множество переходов структуры Крипке строится в соответствии с созданными состояниями и системой переходов: для каждого R^{SP} (соединяющего $S^{\mathrm{SP}}{}_1$ и $S^{\mathrm{SP}}{}_2$) создается переход в структуре Крипке, ведущий из $S^{\mathrm{M}}{}_1$ (созданный для $S^{\mathrm{SP}}{}_1$) в $S^{\mathrm{M}}{}_2$ (созданный для перехода R^{ST}) и далее переход из $S^{\mathrm{M}}{}_2$ в $S^{\mathrm{M}}{}_3$ (созданный для $S^{\mathrm{SP}}{}_2$);

AP — набор атомарных предикатов заполняется автоматически предикатами, соответствующими условиям переходов в ST (каждому уникальному переходу соответствует свой предикат, если условия повторяются, то и предикат повторяется), и атомарными предикатами, соответствующими состояниям системы (каждому состоянию соответствует свой атомарный предикат, эти наборы предикатов могут быть изменены позднее).

L ($S \rightarrow 2^{\mathrm{AP}}$) — функция пометок: т. к. каждому состоянию системы Крипке соответствует состояние или переход ST, а им соответствуют уникальные предикаты (по построению), то построение функции пометок очевидно.

При построении структуры Крипке были определены атомарные предикаты двух типов — описывающие состояния системы переходов и описывающие сами переходы. Для разграничения введем следующие обозначения:

$@S$ — атомарный предикат (или набор, если структура Крипке была модифицирована пользователем), описывающий некое состояние системы;

q, p, ..., r — предикаты описывающие действия внешней среды (условия переходов в системе переходов); тогда основное требование, описанное в построенной структуре Крипке, — это причинно-следственная связь событий — будет описана следующим образом:

$G(@S_1 \wedge p \Rightarrow XF(@S_2))$.

Если система находилась в состоянии $@S_1$ и возникло событие p, то в течение некоторого времени система должна перейти в состояние $@S_2$. Эта запись соответствует описанию прецедента, где $@S_1$ и $@S_2$ определяют начальное и конечное состояния, а p — условие перехода между ними. Например: если прибор находится в режиме проверки ($@S_1$) и проверка прошла успешно (p) то прибор переходит в состояние выбора режима работы ($@S_3$).

По полученной структуре Крипке можно построить вычисления (трассы) перехода от одного состояния системы к другому и в дальнейшем проверить выполнимость этих трасс на реальных системах.

Построенная структура Крипке содержит формализованные проверки достижимости состояния $@S$:

$F@S$ — когда-нибудь в будущем состояние $@S$ будет достигнуто (при построении определенной трассы);

$(\neg@S_2)U(@S_1 \wedge \neg@S_2)$ — состояние $@S_1$ будет достигнуто до достижения $@S_2$ (определяется по построенной трассе);

$(\neg@S_2 \wedge \neg@S_1)U@S_1$ — состояние $@S_2$ не будет достигнуто раньше состояния $@S_1$ (определяется по построенной трассе).

Построенная система также специфицирует отсутствия события $@S$ на определенных трассах:

$G\neg@S$ — состояние никогда не будет достигнуто;

$F@S_1 \Rightarrow (\neg@S_2)U@S_1$ — состояние $@S_2$ не наступит до состояния $@S_1$;

$G(@S_1 \Rightarrow G\neg@S_2)$ — состояние $@S_2$ не наступит после события $@S_1$.

Приведем список атомарных предикатов, выделенных для данной системы. Для упрощения схемы введем предикаты только для состояний системы переходов:

q_1: прибор выключен
q_2: прибор в состоянии проверки
q_3: прибор неисправен
q_4: прибор в состоянии выбора режима работы
q_5: прибор в состоянии проверки результатов
q_6: прибор в состоянии просмотра результатов
q_7: прибор в состоянии проверки памяти для значений параметра плазмы
q_8: прибор в состоянии измерения параметра плазмы
q_9: прибор в состоянии расчета параметра плазмы
q_{10}: прибор в состоянии индикации рассчитанного параметра плазмы

q_{11}: прибор в состоянии проверки памяти для значений параметра крови

q_{12}: прибор в состоянии измерения параметра крови

q_{13}: прибор в состоянии расчета параметра крови

q_{14}: прибор в состоянии индикации рассчитанного параметра крови

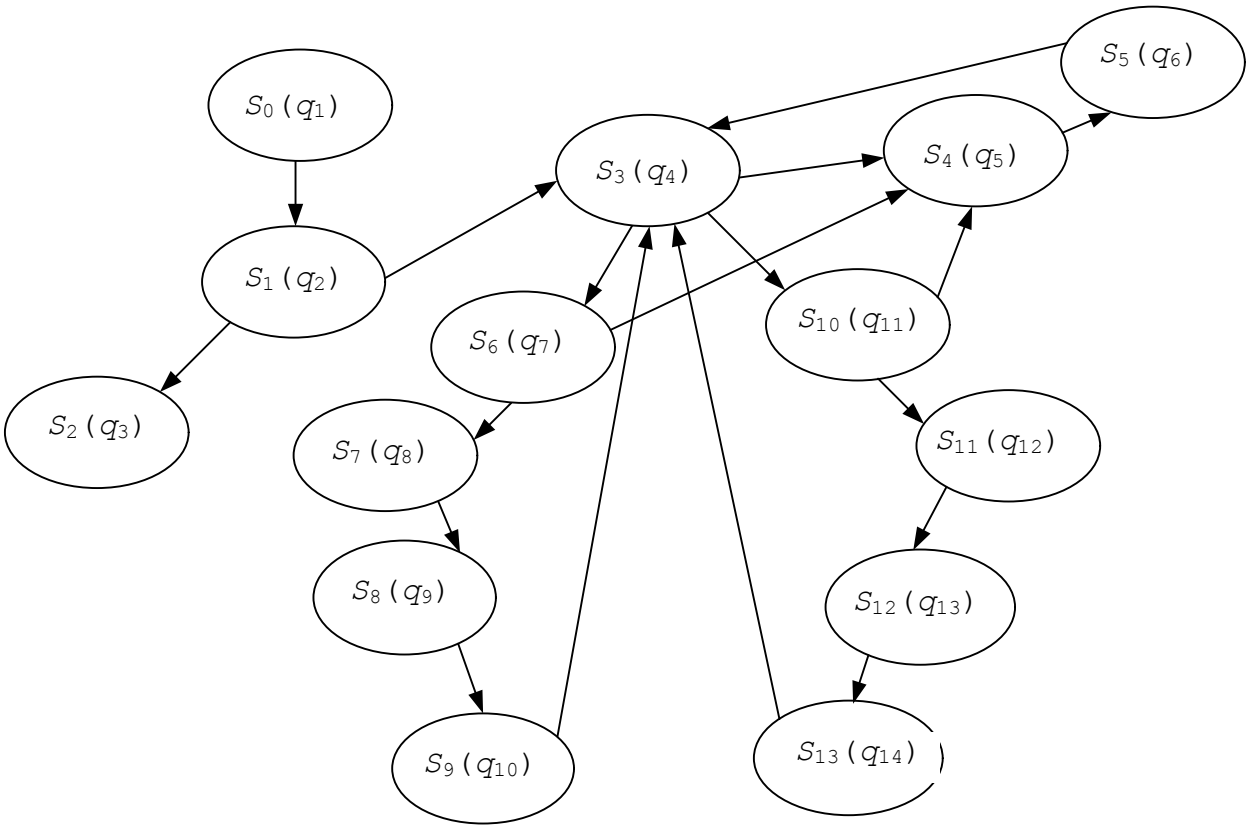
На рисунке представлена построенная струтура Крипке с указанием атомарных предикатов для состояний системы (атомарные предикаты, описывающие условия переходов, опущены для краткости записи). Полученная структура Крипке содержит требования к работе СУ, формализованные с помощью темпоральной логики. Запишем некоторые правила с помощью формул (полное перечисление всех темпоральных формул, описанных в данной структуре Крипке, неуместно: этими формулами оперируют автоматические средства верификации, например Model Checking).

1. $G(@S_1 \wedge q_3 \Rightarrow XF(@S_2))$ — если прибор находился в состоянии проверки и оказался неработоспособен, то он переходит в состояние "неисправен".

2. $G(@S_4 \wedge q_6 \Rightarrow XF(@S_5))$ — если прибор находился в состоянии проверки результатов работы и проверка прошла успешно, то прибор переходит в состояние индикации рассчитанных параметров.

3. $G(@S_2 \Rightarrow G\neg@S_3)$ — если прибор находится в состоянии "неисправен", то он не сможет перейти в состояние "выбор режима работы".

4. $F@S_7 \Rightarrow (\neg@S_9)\,U@S_7$ — состояние "просмотр результатов измерения параметра плазмы" не наступит, до того как будет выполнено "измерение параметра плазмы".



Структура Крипке, описывающая режимы работы прибора "Гранат-4"

ВЫВОДЫ

В результате обработки комплекса исходных данных, описывающих поведение СУ, и представления их с помощью отдельных предикатов была получена спецификация — формализованное описание поведения СУ с помощью формул темпоральной логики и структуры Крипке. Такое описание может служить основой как для реализации ПО СУ, так и для анализа и исследования особенностей работы СУ с помощью методов Model Checking, формальных методов верификации или средств построения и контроля сценариев проверки. Для аналитического прибора "Гранат-4" было построено формальное описание логики работы ПО, позволившее упростить разработку и построить сценарии проверки, максимально полно проверяющие возможные события в течение одной сессии работы с прибором.

Работа выполнена в Санкт-Петербургском государственном университете аэрокосмического приборостроения при поддержке Института аналитического приборостроения РАН (ИАП РАН)

СПИСОК ЛИТЕРАТУРЫ

1. RTCA/DO-178B Software Considerations in Airborne Systems and Equipment Certification. Radio Technical Commission for Aeronautics, 1992.
2. *Шишов С.А.* Лекции по дисциплине "Системное автоматизированное проектирование". Московский государственный институт (технический университет), 1996.
3. *Кулямин В.В.* Методы верификации программного обеспечения.
URL: (http://www.sci-innov.ru/icatalog_new/?entry_id=62322).

Институт аналитического приборостроения РАН, г. Санкт-Петербург (*Краснов Н.В.*)

Международный институт передовых аэрокосмических технологий Санкт-Петербургского государственного университета аэрокосмического приборостроения, г. Санкт-Петербург (*Небылов А.В., Самокиш А.В.*)

Контакты: *Самокиш Андрей Владимирович*,
andrew_samokish@yahoo.com

Материал поступил в редакцию 19.04.2011.

FORMALISATION METHOD FOR DECRIPTION OF EMBEDDED SYSTEM BEHAVIOR DURING SOFTWARE DEVELOPMENT

N. V. Krasnov[1], A. V. Nebylov[2], A. V. Samokish[2]

[1]*Institute for Analytical Instrumentation of RAS, Saint-Petersburg*
[2]*International Institute for Advanced Aerospace Technologies*
(*Saint-Petersburg State University of Aerospace Instrumentation*), *Saint-Petersburg*

This article analyzes the formalization method for embedded system initial data behavior description using temporal formulas and Kripke structure. The formalization process is exemplified by the description of automatic cholinesterase activity level analyzer for human plasma and blood "Granat-4". The obtained formalized description can be used for software development, system analysis with Model Checking, formal verification and preparing test planes.

Keywords: formalization, verification, Kripke structure, temporal logic, cholinesterase, transition system, use case