Балашов В.В.

О ПОДДЕРЖКЕ АНАЛИЗА ПОВЕДЕНИЯ ВСТРОЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

МГУ им. М.В.Ломоносова, Москва, hbd@cs.msu.su

1 ВВЕДЕНИЕ

Под *целевым свойством* поведения ВсВС будем понимать формальное представление требования к правильности функционирования ВсВС.

Целевые свойства отражают ограничения на последовательность действий или изменения состояний компонентов ВсВС, ограничения на количество вычислительных ресурсов, необходимых для функционирования программного обеспечения ВсВС, и т.д.

Важным этапом процесса разработки ВсВС как программно-аппаратной системы является анализ поведения (АП) ВсВС. Цель АП – математически строго проверить, будут ли выполнены целевые свойства поведения ВсВС при любом прогоне и любых допустимых условиях функционирования ВсВС.

Если АП не выявил нарушений целевых свойств, разработка может быть продолжена в соответствии с принятым процессом. Если в ходе АП выявлено нарушение проверяемого свойства, или метод проверки оказался неприменимым, перед разработчиком встаёт задача принятия решения о дальнейших действиях, в т.ч. о внесении изменений в ВсВС. Средства АП ВсВС развиваются в проекте ДИАНА [7] с 1984 г в рамках концепции комплексного подхода к разработке и анализу функционирования ВсВС [8],[9]. Настоящая работа является развитием данного подхода.

Актуальна задача разработки средств поддержки принятия решений разработчиком в ходе выполнения АП ВсВС. В данной работе рассмотрен *расширенный цикл* АП ВсВС, включающий в себя шаг принятия решений по результатам проверки целевого свойства поведения ВсВС. Автором подробно исследован этот шаг, предложена его модель; выделены виды решений, которые должен принимать разработчик по результатам АП. Для отдельных видов решений выдвинуты предложения по организации средств поддержки принятия решений.

2 РАСШИРЕННЫЙ ЦИКЛ АНАЛИЗА ПОВЕДЕНИЯ ВСТРОЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Для разработки средств поддержки принятия решений по результатам АП ВсВС необходимо выделить классы таких решений. В

данном разделе приведена общая последовательность шагов расширенного цикла АП BcBC. Для шага принятия решений автором на основании анализа различных подходов к АП BcBC [1], [2], [6], [8] построено дерево принятия решений. В соответствии со структурой этого дерева, по критерию вида поддерживаемых решений могут быть классифицированы подходы к поддержке АП BcBC, в т.ч. предлагаемые автором в последующих разделах.

На вход очередной итерации цикла АП BcBC поступает следующая информация:

1. представление BcBC на некотором языке описания;
2. математически строгое описание целевого свойства поведения BcBC.

Расширенный цикл АП BcBC состоит из следующих шагов:

АП.1. Выбор метода для проверки свойства.

АП.2. Перевод описания BcBC и целевого свойства в форму, пригодную для применения выбранного метода.

АП.3. Применение метода для проверки свойства.

АП.4. Анализ результатов применения метода.

АП.5. Принятие решения по итогам шага АП.4.

АП.6. Реализация принятого решения.

К результатам применения метода, анализируемым на шаге АП.4, можно отнести:

1. Заключение об успешности или неуспешности применения метода.
2. Заключение о выполненности или невыполненности свойства.
3. Оценка погрешности применения метода для конкретного случая.

Рассмотрим различные варианты результатов применения метода проверки свойства. Каждому из них соответствуют возможные решения со стороны разработчика BcBC и связанные с ними переходы на шаги цикла АП.

Метод применён успешно:

1. Выдано заключение о выполненности свойства ➔ решение: продолжать разработку в соответствии с принятым процессом. *Выход из цикла АП для данного свойства*.
2. Выдано заключение о невыполненности свойства
   a. Оценка погрешности метода достаточно низка, и можно сделать вывод о реальной невыполненности свойства ➔ решение: определить и внести в BcBC изменения, которые приведут к *проверяемой* выполненности свойства. После внесения изменений – *переход к шагу АП.2*.

b. Оценка погрешности метода такова, что из-за неё возможно ошибочное заключение ➔ решение: выбрать более точный метод вычисления тех характеристик поведения ВсВС, которые предположительно больше всего повлияли на погрешность. Переход к *шагу АП.3* для пересчёта выбранных характеристик.
c. Оценка погрешности неизвестна ➔ решение: аналогично пункту a либо аналогично пункту b.

Метод применён неуспешно:

1. Недостаточно данных для применения метода (требуются данные, не содержащиеся в имеющемся описании ВсВС) ➔ решения:
   a. Дополнить информацию о ВсВС недостающими данными. Возможно использование экспертных оценок характеристик поведения ВсВС. *Переход к шагу АП.2.*
   b. Выбрать метод, для которого достаточно имеющихся данных. *Переход к шагу АП.1.*
   c. Внести изменения в ВсВС для обеспечения применимости метода. *Переход к шагу АП.2.*
2. Недостаточно вычислительных ресурсов инструментальной машины для применения метода ➔ решения:
   a. Изменить формулировку целевого свойства с тем, чтобы снизить сложность его проверки (снижение детальности, разбиение на более простые свойства). *Переход к шагу АП.2.*
   b. Выбрать метод, менее требовательный к ресурсам. *Переход к шагу АП.1*
   c. Использовать экспертные оценки для характеристик или элементарных свойств, вычисление которых требует наибольшего количества ресурсов. *Переход к шагу АП.2.*
   d. Использовать инструментальную машину, на которой достаточно ресурсов. *Переход к шагу АП.3.*

Замечание: при неуспешном применении метода или при недостаточно точных результатах его применения возможен отказ от строго формальной проверки свойства в пользу тестирования.

В случае если свойство действительно нарушено, успешные действия по обеспечению применимости методов или повышению их точности в конечном счёте приведут к необходимости принятия решения о выборе и внесении изменений в ВсВС для обеспечения *проверяемой* при помощи имеющихся методов выполненности свойства. Это соответствует ветви 2.a для случая успешного применения метода проверки свойства.

3 ПОДДЕРЖКА ВЫБОРА ИЗМЕНЕНИЙ ДЛЯ ВНЕСЕНИЯ В ВСТРОЕННУЮ СИСТЕМУ

В случае выявления нарушения целевого свойства (ветвь дерева принятия решений 2.а для успешного применения метода), изменения для внесения в ВсВС должны выбираться так, чтобы не только обеспечить выполненность нарушенного свойства, но и не нарушить выполненность других целевых свойств ВсВС. К целевым свойствам в данном случае относятся как свойства, формально проверяемые в рамках АП, так и другие свойства, формальная проверка которых затруднительна и которые проверяются в рамках тестирования (например, свойство правильности результатов, выдаваемых реализацией вычислительного алгоритма). В связи с этим задача полностью автоматического выбора (синтеза) изменений на сегодня представляется чрезмерно сложной.

Предлагается организовать поддержку выбора изменений для внесения в ВсВС не посредством синтеза этих изменений в виде, пригодном для автоматического внесения, а посредством предоставления разработчику ВсВС рекомендаций, на основании которых он может выбрать конкретные изменения с учётом своего знания контекста разработки и взаимозависимостей требований к ВсВС.

Информация, выдаваемая системой поддержки принятия решений по факту обнаружения нарушения целевого свойства, должна содержать следующие разделы:

- *Диагностика* (описание нарушения свойства)
- *Контекст* нарушения (место в анализируемой системе, в котором локализовано нарушение)
- *Рекомендация* (указания разработчику по устранению нарушения)
- *Обоснование* диагностики и рекомендации

Эта структура информации аналогична предложенной в работе [4], посвящённой разработке программных систем-критиков, используемых для поддержки разработки программного обеспечения. Специфика систем-критиков [3], [4] состоит именно в том, что они выдают разработчику рекомендации по ведению разработки, на основании которых он может действовать с учётом своего знания совокупности требований к создаваемой системе.

4 ПРИМЕРЫ ПОДХОДОВ К ПОДДЕРЖКЕ АНАЛИЗА ПОВЕДЕНИЯ ВСТРОЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

В данном разделе рассмотрены примеры подходов к поддержке АП ВсВС в рамках модели цикла АП, введённой в разделе 2. В качестве ситуаций, в которых требуется принятие решений, выбраны:

1. Нарушение целевого свойства поведения ВсВС (ветвь дерева принятия решений 2.а для успешного применения метода)
2. Невозможность применения метода проверки свойства из-за нехватки входных данных (ветвь дерева принятия решений 1.а для неуспешного применения метода)

Другие примеры не представлены из-за ограничений по объёму.

4.1 Не выполнено свойство гарантированной планируемости набора задач

Данное свойство формулируется как "все задачи на узле с динамическим планированием задач (ДПЗ) должны выполняться в пределах своих директивных сроков при любом прогоне ВсВС". В ВсВС с ДПЗ расписание выполнения задач не строится заранее, и, следовательно, невозможно явно по расписанию и параметрам набора задач (НЗ) проверить свойство гарантированной планируемости (ГП) НЗ.

Для проверки свойства ГП в работе [1] предложены формулы, которые по параметрам НЗ (периодам, приоритетам, максимальным временам выполнения) позволяют для каждой задачи оценить сверху максимальное время отклика R_i (время от активации экземпляра задачи до завершения его выполнения). С их помощью можно проверить условие ГП, имеющее математическую формулировку: $R_i \le D_i (i = 1..N)$, где D_i – директивный срок задачи.

В работе [5] предложен подход, позволяющий в случае нарушения свойства ГП НЗ осуществить автоматический подбор изменений параметров НЗ таким образом, чтобы добиться выполненности этого свойства.

В качестве численной оценки (меры) выполненности свойства ГП можно рассматривать минимальный по всем задачам относительный запас времени до директивного срока: $Y_{rel} = \min_i (D_i - R_i) \Big/ D_i$

Для поддержки принятия разработчиком решений по результатам проверки свойства ГП может быть построено программное средство, запускаемое непосредственно после выявления нарушенности этого свойства. На вход средству поступают: параметры набора задач и рамки их допустимых изменений; значение меры выполненности свойства ГП; диапазон допустимых значений меры (в простейшем случае нижняя граница равна 0, а верхняя не определена).

Результатом работы средства будет являться: (а) *Диагностика* – сообщение о выходе значения меры ГП за пределы допустимого диапазона; (б) *Контекст* нарушения – перечень задач, для которых запас времени до директивного срока недопустимо мал или велик; (в)

Рекомендация – вычисленные по методу, изложенному в [5], изменения параметров НЗ, достаточные для достижения выполненности свойства ГП; (г) *Обоснование*: ссылка на работы, в которых обоснована методика, по которой работает средство.

Средство поддержки разработки, результат работы которого содержит перечисленные выше пункты, относится к классу программных систем-критиков, упомянутых в разделе 3.

4.2 Недостаточно данных для вычисления максимального времени выполнения фрагмента программного кода

При разработке последовательных вычислительных задач для ВсВС жёсткого реального времени важно соблюдать ограничения на максимальное время выполнения (МкВВ) этих задач на ВсВС.

Для оценки МкВВ фрагментов кода последовательной задачи существуют методы [6], позволяющие оценить МкВВ сверху. Методы оценки МкВВ осуществляют анализ графа управления задачи, содержащего информацию о временах выполнения линейных участков кода и об ограничениях на возможные пути в графе, в т.ч. на число итераций циклов.

В ряде случаев эти ограничения не могут быть получены посредством анализа кода вычислительной задачи. Например, если верхняя граница диапазона изменения счётчика цикла "for" вычисляется некоторой функцией или зависит от нелокальной для оцениваемого фрагмента переменной, метод не может определить максимальное число итераций цикла. Следовательно, метод не может дать оценку МкВВ фрагмента кода, содержащего цикл.

В подобных случаях для применения метода разработчику необходимо явно указать диапазоны изменения значений, которые не могут быть автоматически вычислены методом АП. По сути, такая информация является экспертной оценкой разработчика с учётом его знания контекста разработки. Поддержка принятия решений заключается в указании разработчику на:

1. фрагменты кода, для которых необходимо указать дополнительные данные;
2. фрагменты кода, которые могут помочь разработчику определить значения дополнительных данных (например, код функции, диапазон значений которой пользователь должен указать)

5 ЗАКЛЮЧЕНИЕ

В работе рассмотрен расширенный цикл анализа поведения (АП) ВсВС. Автором построено дерево принятия решений на шаге принятия решений по результатам АП ВсВС. Предложены методы поддержки принятия решений для отдельных методов АП и видов принимаемых решений.

В настоящее время перед автором стоят следующие задачи:

1. Дальнейшее изучение методов АП ВсВС.
2. Разработка методов поддержки принятия решений по результатам АП ВсВС в соответствии со структурой построенного дерева принятия решений.
3. Уточнение структуры дерева принятия решений.

Основой для реализации разработок является среда моделирования ВсВС ДИАНА [7]. В ней реализованы средства проверки логических свойств поведения разрабатываемых ВсВС, планируется реализация средств оценки максимального времени выполнения программ и проверки гарантированной планируемости наборов задач.

[1] I.J. Bate. Scheduling and Timing Analysis for Safety Critical Real-Time Applications. PhD Thesis, University of York, 1998.

[2] Захаров В.А., Царьков Д.В. Эффективные алгоритмы проверки выполнимости формул темпоральной логики CTL на модели и их применение для верификации параллельных программ. М.:Программирование, 1998, #4, с.3-18

[3] J.E. Robbins. Design Critiquing Systems. University of California, 1998.

[4] E. Liu. Proposal for a Software Metrics-Based Critiquing System. M.Sc. Thesis, University of Calgary, 2000.

[5] Балашов В.В. О внесении изменений во встроенную систему при нарушении директивных сроков задач. М.:"Программные системы и инструменты", 2002.

[6] S.M. Petters, G. Färber. Making Worst Case Execution Time Analysis for Hard Real-Time Tasks on State of the Art Processors Feasible. University of York, 1999.

[7] A.Bakhmurov, A.Kapitonova, R.Smelinasky. DYANA: An Environment for Embedded System Design and Analysis. Proceedings of 32-nd Annual Simulation Symposium, San Diego, California, USA, 1999.

[8] Смелянский Р.Л. Поведение программ в распределенных вычислительных системах и инструментарий для его анализа. Технология программирования МПТ, Ленинград, 1990.

[9] Молонов В.Г., Смелянский Р.Л. Комплексный подход к моделированию распределённых вычислительных систем. М.: Программирование, 1988, #1, с. 57-65