УДК 004.052

О.М. ТАРАСЮК, А.В. ГОРБЕНКО, В.С. ХАРЧЕНКО

КОМПЛЕКСИРОВАНИЕ ФОРМАЛЬНЫХ МЕТОДОВ РАЗРАБОТКИ И АНАЛИЗА НАДЕЖНОСТИ EVENT-B И FME(C)A

Abstract. *The paper analyses existing obstacles problems and potentialities of applying Event-B formal technique when developing fault-tolerant computing systems. We discuss an integration of the Event-B and technique of failure modes and effect analysis FME(C)A to provide an approach for identification of possible failures, estimation of their criticality, as well as optimal choice and formal proving of fault-tolerant and recovery techniques. The basic procedures of transition from Event-B model of correct system to Event-B model of fault-tolerant system is also given.*
Key words: *formal methods, Event-B, FME(C)A, fault-tolerant systems.*

Анотація. *Аналізуються існуючі проблеми й можливості застосування формальних методів під час створення відмовостійких комп'ютерних систем. Розглянуто задачу інтеграції формального методу розробки Event-B та методу аналізу надійності FME(C)A для виявлення можливих відмов, оцінки їхньої критичності, а також оптимального вибору та формального доказу коректності засобів відновлення й забезпечення відмовостійкості. Запропоновано процедуру переходу від Event-B моделі коректної системи до Event-B моделі коректної відмовостійкої системи.*
Ключові слова: *формальні методи, Event-B, FME(C)A, відмовостійкі системи.*

Аннотация. *Анализируются существующие проблемы и возможность применения формальных методов при разработке отказоустойчивых компьютерных систем. Рассматривается интеграция формального метода разработки Event-B и метода анализа надежности FME(C)A для выявления возможных отказов, оценки их критичности, а также оптимального выбора и формального доказательства корректности средств восстановления и обеспечения отказоустойчивости. Предложена процедура перехода от Event-B модели корректной системы к Event-B модели корректной отказоустойчивой системы.*
Ключевые слова: *формальные методы, Event-B, FME(C)A, отказоустойчивые системы.*

1. Введение

В настоящее время большой интерес специалистов в области разработки компьютерных систем и программного обеспечения, в первую очередь, для критических приложений, вызывают так называемые формальные методы – группа методов, предназначенных для формализации процесса разработки, такие как VDM [1], Z [2], B [3] и Event-B [4], Model Checking [5] и др. Эти методы являются частным случаем математически ориентированных приемов спецификации требований, разработки и верификации программного и аппаратного обеспечения и систем в целом. Одним из наиболее известных формальных методов, используемых для спецификации требований и разработки систем, является метод Event-B, основанный на применении аксиом и правил вывода для доказательства корректности функционирования системы.

Вследствие высоких требований к опыту и квалификации разработчиков, а также их специальной математической подготовке, формальные методы применяются для разработки сложных систем с высокими требованиями, прежде всего, к таким составляющим гарантоспособности, как надежность (безотказность) и функциональная безопасность. Как правило, при проектировании таких систем для снижения вероятности возникновения отказов также широко используются методы повышения надежности, основанные на использования различных видов избыточности и способов резервирования, специальных механизмов отказоустойчивости и отказобезопасности. Под отказобезопасностью понимают такой вид отказоустойчивости, который обеспечивает способность системы парировать опасные отказы и/или не допускать их катастрофические последствия [6].

Однако при попытках применения метода Event-B для разработки отказоустойчивых систем возникает ряд проблем технологического и методического характера. Прежде всего, это связано с

ISSN 1028-9763. Математичні машини і системи, 2010, № 2

противоречием, возникающим из-за особенностей самого метода. Event-B предназначен для создания корректных систем, т.е. «идеальных» систем, в то время, как отказоустойчивая система допускает некорректное поведение (возникновение отказов в системе), которое должно быть парировано.

Тем не менее актуальность рассматриваемой проблемы подчеркивается многочисленными примерами, когда дефекты именно средств отказоустойчивости и восстановления после отказов, а также ошибки в обработке исключительных ситуаций, привели в конечном итоге к возникновению и развитию катастрофической ситуации, а не её предупреждению [7].

Так, по результатам расследования катастрофы ракетоносителя Arian-5, произошедшей 4 июня 1996 года, комиссия Европейского космического агентства установила, что её причиной стало некорректно обработанное программное исключение, вызванное при переполнении во время преобразования данных из 64-разрядного формата с плавающей точкой в 16-разрядное целое и приведшее к отказу основного и резервного каналов навигационной системы [8].

Другим показательным примером служит каскадный отказ энергосистем Канады и США, произошедший 14 августа 2003 года, который выявил существенные недостатки в механизмах отказоустойчивости и восстановления после отказов [7].

По заключению экспертов Ростехнадзора, одной из причин катастрофы на Саяно-Шушенской ГЭС в России также является отказ многоуровневой системы автоматической защиты [9].

Таким образом, практика проектирования и применения систем для критических приложений указывает на необходимость разработки методов и технологий, которые бы, с одной стороны, минимизировали риски отказов, обусловленных как физическими, так и проектными дефектами систем и средств обеспечения их отказоустойчивости, а, с другой, обеспечили бы верифицируемость принимаемых решений с использованием формальных (математически обоснованных) методов.

Целью данной статьи является решение трех взаимосвязанных задач:

– во-первых, анализ возможностей (особенностей) применения формального метода Event-B для разработки отказоустойчивых и отказобезопасных компьютерных систем;

– во-вторых, аксиоматизация понятий полноты и минимальности множеств инвариантов (ключевых элементов Event-B) и ошибок, связанных с их использованием;

– в-третьих, разработка предложений по комплексированию формальных методов разработки и анализа отказов гарантоспособных систем на примере Event-B и FME(C)A.

Решение первой задачи направлено на создание систем, свободных от дефектов проектирования, и, как следствие, сокращение затрат на обеспечение устойчивости к отказам, обусловленным данным классом дефектов. Кроме того, применение Event-B предоставляет потенциальную возможность математически обосновывать (доказывать) корректность не только базовых функций системы, но и средств и механизмов отказоустойчивости и отказобезопасности, используемых для парирования физических отказов элементов.

Вторая задача связана с необходимостью формализации процесса оценки «качества» системы инвариантов, используемых в нотации Event-B. Речь идет о том, какая полнота и

достоверность контроля корректности функционирования обеспечивается при используемом множестве инвариантов для специфицированного множества отказов.

Решение третьей задачи является, на наш взгляд, естественным дополнением к первым двум, поскольку обеспечение полноты контроля и диагностирования, а также повышение других показателей надежности (гарантоспособности) требует детального анализ отказов, вызванных различными причинами. Для этого широко применяются специальные формальные и частично формализованные методы анализа надежности (FTA, RBD, FME(C)A и др.), которые могут хорошо комплексироваться на всех этапах жизненного цикла с формальными методами специфицирования и разработки.

Статья структурирована следующим образом: во втором и третьем разделах дается краткое описание формального метода Event-B и особенностей его применения для диагностирования отказов. Анализ ошибок, связанных с использованием инвариантов, а также понятия полноты и минимальности их множеств, представлены в четвертом и пятом разделах соответственно. Шестой раздел посвящен разработке общего подхода и процедуры совместного использования формальных методов Event-B и FME(C)A, седьмой – разработке элементов синтеза модели Event-B отказоустойчивой системы. В восьмом разделе обсуждаются полученные результаты и формулируются направления дальнейших исследований.

2. Формальный метод Event-B

2.1. Сущность Event-B

Метод Event-B[1] появился в результате эволюции классического метода B [3], основанного на использовании нотации абстрактной машины AMN (Abstract Machine Notation) для формальной разработки программного обеспечения. В свою очередь, метод B, предложенный Жаном-Раймондом Абриалем, базируется на использовании Z-нотации и позволяет получить программный код путем разработки и поэтапной детализации формальной спецификации дискретных систем.

Метод B был успешно использован при разработке многих критических компьютерных систем, наиболее известными из которых являются автоматическая система управления линией №14 Парижского метро (1998 г.), автоматическая система управления экспрессом Парижского аэропорта (2006 г.). Event-B формализует процесс описания свойств и динамического поведения систем, а также обеспечивает контроль за соблюдением этих свойств в процессе функционирования на основе механизма предусловий. В качестве математического аппарата используются логика предикатов, Булева алгебра и теория множеств.

При использовании формального метода Event-B спецификация системы представляется в виде её формальной модели (машины), основными элементами которой являются:

– набор системных переменных (variables), конкретное значение которых отражает определенное состояние системы (state);

– контекст (context), представленный в виде набора системных констант;

– инварианты (invariants) – набор свойств (условий), истинность которых должна всегда соблюдаться в процессе функционирования системы;

[1] http://www.event-b.org.

– события (events), возникающие внутри системы или за её пределами и переводящие систему из одного состояния в другое путем выполнения системой определенных операций, изменяющих значение системных переменных, в качестве реакции на каждое конкретное событие;

– набор предусловий (guards) для каждого события, запрещающих его возникновение, если в результате реакции системы на это событие произойдет нарушение инвариантов.

2.2. Принципы Event-B

Ключевыми принципами формального метода Event-B являются следующие.

Рефайнмент (refinement) или детализация, предполагающая поэтапный переход от более абстрактной модели системы к более конкретной. Детализация выполняется путем добавления новых событий или изменения существующих, добавления новых операций, переменных, предусловий, инвариантов или констант.

Автоматическое доказательство, выполняемое путем перебора всех событий и инвариантов, и математического доказательства того, что при возникновении каждого события с учетом механизма предусловий не происходит нарушения ни одного инварианта. При этом на каждом новом этапе детализации выполняется доказательство только новых теорем.

Применение Event-B позволяет повысить качество требований к системе (снизить вероятность дефектов в требованиях), а также значительно сократить или исключить полностью дефекты проектирования, гарантируя корректное поведение системы в рамках используемой системы инвариантов.

3. Применение Event-B для диагностирования отказов

При проектировании отказоустойчивых систем разработчики рассматривают некоторую группу отказов, для которых предусматриваются действия, парирующие эти отказы, не допуская некорректного поведения системы.

Однако в рамках спецификации Event-B, определяющей именно корректное поведение системы, каждый отказ должен приводить к нарушению одного или нескольких инвариантов или к блокировке модели, т.е. возникновению такой ситуации, которая в принципе не должна быть возможна при описании системы с помощью нотации Event-B.

Эта особенность может быть использована для построения системы диагностирования отказов и непредусмотренных ситуаций, основанной на выполнении модели Event-B параллельно с функционированием самой системы. Тогда возникновение в системе события, которое невозможно (запрещено из-за невыполнения предусловий (guards)) в рамках модели Event-B, или же блокировка модели (когда ни одно из событий не может возникнуть из-за невыполнения предусловий (guards)) будут свидетельствовать о наличии отказа, вследствие которого произошло некорректное поведение системы.

Могут быть использованы два подхода к построению подобной системы диагностирования. Первый из них базируется на контроле нарушения исключительно инвариантов модели Event-B. Отказ обнаруживается в тот момент времени, когда в системе происходит событие, нарушающее целостность системы инвариантов.

Однако, как показывает практика построения моделей Event-B реальных систем управления, возможны ситуации, когда отказ в системе не приводит к немедленному нарушению инвариантов, т.е. не проявляется сразу же после возникновения. Более того, при определенных условиях использования системы отказ может не проявляться бесконечно долго (так называемый скрытый отказ). В этом случае имеет место позднее или отстроченное обнаружение отказов.

Второй подход заключается не только в контроле сохранения инвариантов модели Event-B, но и в контроле согласованности значений переменных состояния (хотя эта ситуация также может контролироваться введением специального инварианта). Следовательно, факт расхождения переменных состояния системы и модели Event-B так же, как и нарушение инвариантов модели, будет свидетельствовать о возникновении отказа в системе или же о некорректном (непредусмотренном) варианте её использования.

4. Ошибки инвариантов первого и второго рода

Построение системы диагностирования, основанной на совместном выполнении модели Event-B, оправдано, если существует уверенность в корректности самой модели, полноте и отсутствии ошибок в системе инвариантов.

Об ошибках первого рода (в классической диагностике – «риск поставщика» [10]) будет свидетельствовать тот факт, когда нарушение инвариантов происходит в процессе нормального функционирования системы. Это означает наличие избыточных или так называемых «слишком сильных» инвариантов, которые ошибочно ограничивают область допустимого применения системы.

К ошибкам второго рода («риск заказчика») можно отнести ситуации, когда:

– наблюдаемый отказ элемента системы или некорректное использование системы не приводит к нарушению инвариантов модели, т.е. система инвариантов является неполной;

– наблюдаемый отказ элемента системы или некорректное использование системы не приводит к расхождению значений переменных состояния системы и модели, т.е. модель является неполной или некорректной;

– при корректном поведении системы происходит расхождение значений переменных состояния, т.е. модель системы является некорректной.

Причинами возникновения ошибок инвариантов первого и второго рода являются ошибки разработчиков, которые не учли все варианты использования системы, допустили логические ошибки при её проектировании или же не зафиксировали все ключевые свойства системы с помощью инвариантов. Кроме того, средства автоматического доказательства теорем, используемые для проверки правильности модели, не являются идеальными. Иногда инварианты могут быть доказаны только «вручную», что также не исключает ошибок.

С точки зрения Event-B, система является корректной, если при ее функционировании не нарушается ни один из инвариантов, что может быть доказано математически (т.е. можно говорить лишь о корректности в рамках используемой системы инвариантов).

Таким образом, можно утверждать, что при отсутствии ошибок первого и второго рода нарушение инвариантов может быть вызвано некорректным поведением системы, обусловленным

физическим отказом компонентов, возникновением других непредусмотренных событий или нецелевым использованием системы.

5. Полнота и минимальность системы инвариантов

5.1. Основные понятия

Исходя из приведенных рассуждений, можно сделать некоторое обобщение. Суть его состоит с том, что для проектируемой системы следует определить понятия полного и минимального множеств инвариантов.

Полное множество инвариантов (ПМИ), $MI_{ПМИ} = \{I_i\},\ i = 1,..., n_{ПМИ}$ – это такая совокупность инвариантов, при которой доказуемо контролируются все специфицированные свойства системы и обнаруживаются их возможные нарушения (отказы).

Следует подчеркнуть, что о полноте множества инвариантов, как и в классической диагностике, можно говорить только для допустимых, т.е. описанных свойств и их нарушений. Инварианты, фиксирующие свойства, не относящиеся к допустимым, будем называть недопустимо избыточными (НДИ), $MI_{НДИ} = \{I_j\},\ j = 1,..., n_{НДИ}$, $MI_{НДИ} \not\subset MI_{ПМИ}$.

Минимальным множеством инвариантов (ММИ), $MI_{ММИ} = \{I_к\},\ к = 1,..., n_{ММИ}$, $MI_{ММИ} \subseteq MI_{ПМИ}$ назовем такую их совокупность, удаление из которой хотя бы одного инварианта приводит к нарушению свойства полноты.

Минимальных множеств инвариантов может быть несколько. Подмножество (подмножества) инвариантов, представляющих разность между ПМИ и ММИ, назовем допустимо избыточными (ДИИ), $MI_{ДИИ} = \{I_i\},\ I = 1,..., n_{ДИИ}$, $MI_{ДИИ} = MI_{ПМИ} \setminus MI_{ММИ}$.

При наличии полного множества инвариантов задача нахождения минимальных множеств решается с использованием булевой матрицы $\sum = \|\sigma_{ij}\|$, столбцы которой соответствуют множеству специфицированных свойств системы и их возможным нарушениям (отказам), а строки – инвариантам, образующим полное множество.

В ячейках матрицы $\sum$ указывается признак – булева переменная σ_{ij}, которая равна 1(0), если j-ое свойство системы или его нарушение гарантированно контролируется (не контролируется) инвариантом i.

5.2. Доказательство полноты и минимальности множеств инвариантов

Доказательством полноты с использованием матрицы $\sum$ является наличие в каждом столбце хотя бы одной единицы: $\forall i : \exists j,\ \sigma_{ij} = 1$.

Соответственно доказательством неполноты является наличие хотя бы одного столбца с нулевыми элементами: $\exists i : \forall j,\ \sigma_{ij} = 0$.

С помощью матрицы $\sum$ удобно также формально задавать условия минимальности и неминимальности множеств инвариантов, обладающих свойством полноты. Кроме того, с ее

помощью легко решается задача поиска всех минимальных множеств инвариантов как задача покрытия. Аналогичными являются задачи поиска тупиковых (минимальных) форм функций по импликантной таблице или синтеза минимальных тестов по таблице функций неисправностей [11].

Исходя из рассмотренных понятий, можно сказать, что наличие недопустимо избыточных инвариантов приводит к увеличению рисков ошибок первого рода, а нарушение свойства их полноты – рисков ошибок второго рода.

5.3. Метрики качества системы инвариантов

Для оценки качества некоторого множества инвариантов MI_X целесообразно ввести специальные метрики, примеры которых приведены ниже:

– метрика полноты: $\mu_{ПМИ} = \left|MI_{ПМИ} \cap MI_X\right| / \left|MI_{ПМИ}\right|$;

– метрика минимальности (допустимой избыточности): $\mu_{ММИ} = \left|MI_X \setminus MI_{ММИ}\right| / \left|MI_{ММИ}\right|$;

– метрика недопустимой избыточности: $\mu_{НДИ} = \left|MI_X \setminus MI_{ПМИ}\right| / \left|MI_{ПМИ}\right|$ и др.

6. Совместное применение Event-B и формальных методов анализа надежности

6.1. Анализ видов, причин и последствий отказов FME(C)A

Очевидно, что для построения высоконадежных (гарантоспособных) систем корректность спецификации и её представление соответствующей нотацией является необходимым, но недостаточным условием. Корректность предполагает отсутствие отказов, обусловленных дефектами проектирования (design faults).

Для защиты же от отказов, обусловленных дефектами или старением физических элементов (physical faults), а также внешними воздействиями, в том числе информационными (intrusions), необходимо наличие средств отказоустойчивости (fault- and intrusion-tolerance), т.е. средств, обеспечивающих реализацию всех составляющих операционного цикла обнаружения, локализации, парирования отказов и восстановления вычислительного (управляющего) процесса.

Необходимым условием построения эффективных отказоустойчивых систем является анализ возможных видов, причин и последствий отказов, вызванных различными причинами. В этом контексте актуальным представляется совместное применение формальных методов разработки систем и формальных методов анализа их надежности. Особый интерес представляет совместное применение Event-B и метода анализа видов и последствий критических отказов FME(C)A и его модификаций [12, 13]. Такой вывод логичен с учетом определенных выше понятий полноты и минимальности множеств инвариантов, поскольку FME(C)A- и IME(C)A-таблицы представляют собой систематизированную информацию о возможных отказах.

Сущность FME(C)A-анализа состоит в [14]:

– определении уровня, на котором проводится анализ (формировании иерархии элемент-система), и множества элементов, отказы которых анализируются с точки зрения влияния на работоспособность системы;

– определении видов отказов каждого из элементов;

– анализе последствий отказов каждого из них для работоспособности системы;

– определении вероятности и тяжести последствий этих отказов (по качественной или количественной шкале оценивания);

– построении двух- (вероятность – тяжесть последствий) или трехмерной (вероятность – тяжесть последствий – время восстановления) матрицы критичности, в которой каждый из элементов размещается в определенной ячейке матрицы в соответствии с его критичностью;

– определении множества средств, которые могут снизить вероятность и тяжесть последствий отказов.

6.2. Процедура комплексирования Event- и FME(C)A

Иерархический подход к анализу видов причин и последствий отказов, предложенный в [15], может быть совмещен с процедурой детализации (refinement), являющейся основой Event-B метода. В этом случае для абстрактной модели Event-B формируется абстрактная FMEA-таблица. В процессе выполнения процедуры детализации, когда выполняется очередной переход от более абстрактной к более конкретной модели системы, соответственно выполняется и детализация FMEA-таблицы. Таким образом, имеем иерархию FMEA-таблиц, соответствующую иерархии моделей Event-B системы.

В свою очередь, операции декомпозиции (decomposition) модели Event-B будет соответствовать операция разбиения более абстрактной FMEA-таблицы на несколько FMEA-таблиц следующего уровня детализации. Такой подход позволяет решить проблему размерности и сложность FMEA-анализа для многокомпонентных иерархических систем, а также рассматривать дополнительное свойство прослеживаемости (трассируемости) FMEA-анализа, что особенно важно для независимой экспертизы и верификации.

Подобный иерархический подход может быть распространен и для совместного использования Event-B с другими методами формального анализа надежности, например, FTA, HAZOP. Отметим, что последний из указанных методов предназначен для контроля последствий выхода технологических параметров производственных процессов за допустимые пределы, что является прямым аналогом наиболее распространенного вида инвариантов Event-B, накладывающих ограничение на область допустимых значений переменных состояния системы.

6.3. Автоматическая генерация видов отказов

В рамках предложенного подхода к совместному использованию формальных методов разработки и анализа надежности предлагается дополнить метод Event-B процедурой генерации видов отказов, который может быть реализован автоматически.

В данном случае отказом будет считаться событие, приводящее к нарушению хотя бы одного инварианта. Таким образом, для каждого инварианта модели Event-B может быть получен набор потенциальных отказов. Затем эти отказы могут быть включены в FMEA-таблицу для дальнейшего анализа их причин, последствий и критичности. Одним из критериев включения потенциального отказа в FMEA-таблицу является его критичность. С другой стороны, наличие информации, содержащейся в независимо построенной FMEA-таблице, дает возможность верифицировать полноту множества инвариантов.

Критичность – это интегральная характеристика, учитывающая вероятность возникновения отказа, тяжесть его последствий, время неработоспособности и другие составляющие [16]. Критичность конкретного отказа может быть определена с помощью матрицы критичности (при качественном анализе) или с помощью вероятностного моделирования с использованием структурных схем безопасности [5] или других методов, относящихся к технологии PSAM (Probabilistic Safety Assessment And Management) [17] при количественном анализе.

Для сокращения размерности FMEA-таблицы при выполнении очередного шага детализации из нее могут быть удалены наименее критические отказы. Однако информация об этих отказах будет сохранена в более абстрактной FMEA-таблице и при необходимости (при пересмотре значений характеристик критичности отказа или изменении граничного уровня критичности (снижении диагонали критичности)) эти отказы могут быть восстановлены для последующего анализа.

После выполнения заключительного шага детализации FMEA-таблица будет содержать финальный набор отказов, критических для разрабатываемой системы.

В соответствии с [16] следующими этапами FMEA-анализа будет выполнение операций формирования списка методов и средств снижения критичности отказов (обеспечения отказоустойчивости) и решение оптимизационной задачи их выбора (рис. 1).

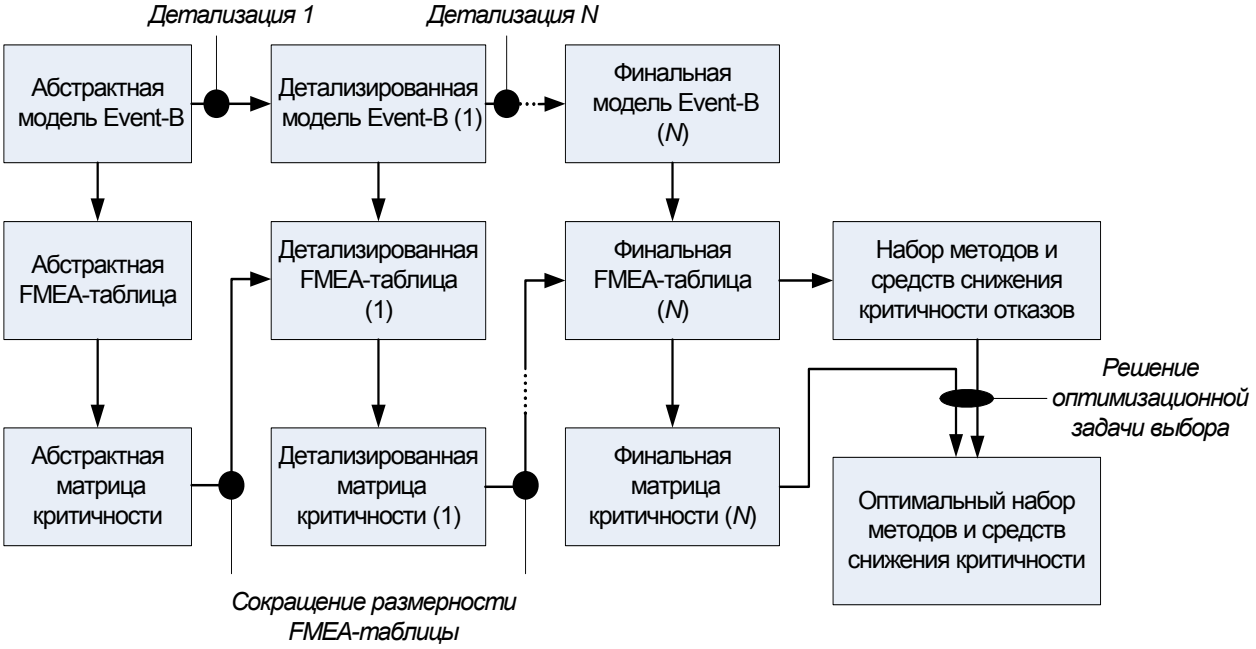

Рис. 1. Совместное использование формальных методов Event-B и FMEA

7. Разработка модели Event-B отказоустойчивой системы

Для перехода от модели Event-B корректной системы к модели отказоустойчивой системы необходимо, во-первых, ввести в корректную модель события отказов, внутри которых будут соответствующим образом изменены системные переменные. Эти события должны соответствовать критическим отказам из FMEA-таблицы. Очевидно, что наличие таких событий-отказов приведет к нарушению инвариантов корректной модели.

Во-вторых, в набор состояний отказоустойчивой модели Event-B должны быть введены корректирующие события, реагирующие на возникновение событий-отказов, внутри которых будут определены действия по парированию этих отказов.

Поскольку разные отказы могут приводить к нарушению одних и тех же инвариантов, а также в результате одного отказа может нарушаться более одного инварианта, то целесообразно определить корректирующие события не для каждого события-отказа, а для события-нарушения инварианта.

В-третьих, в качестве условия срабатывания корректирующих событий должны быть записаны инварианты корректной модели (вернее их логическое отрицание), а инварианты самой системы должны быть изменены таким образом, чтобы учитывать наличие средств отказоустойчивости и выполнение действия по коррекции системы. Это может быть выполнено с помощью добавления соответствующих дизъюнктов к исходным инвариантам корректной системы.

Таким образом, процесс совместного применения формальных методов Event-B и FMEA может быть детализирован, как это показано на рис. 2.

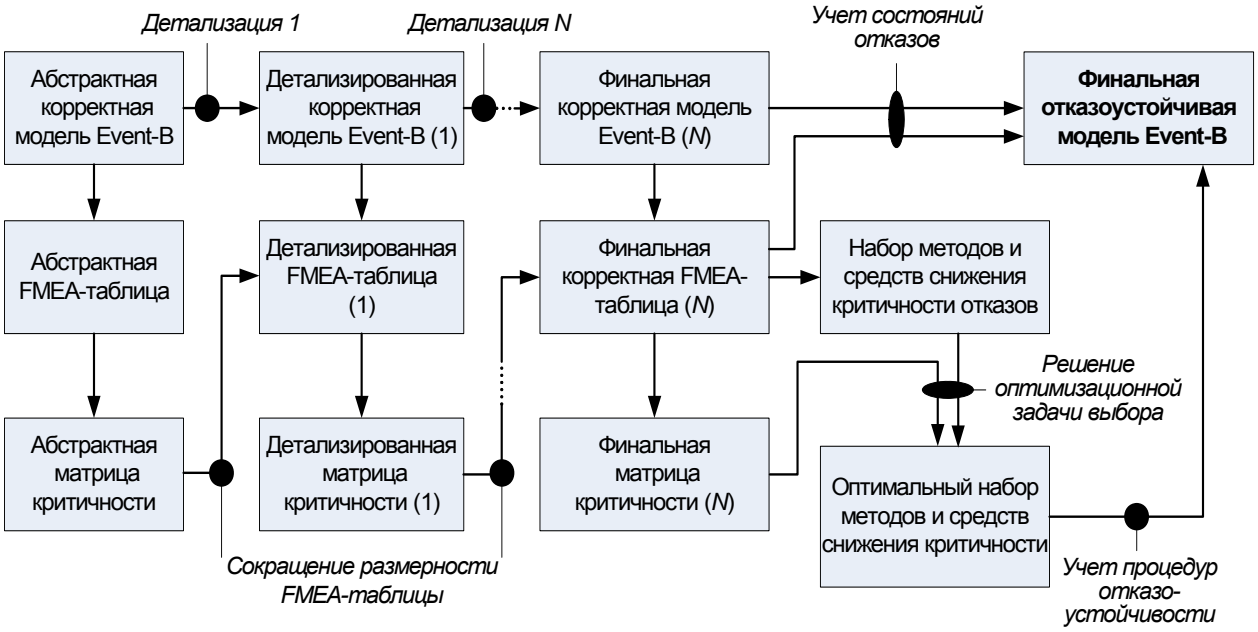

Рис. 2. Применение формальных методов Event-B и FME(C)A
для разработки отказоустойчивых систем

Необходимо отметить, что возможны два подхода к построению модели Event-B отказоустойчивой системы.

Первый из них предполагает получение сначала финальной версии модели Event-B корректной системы, после чего выполняется переход к отказоустойчивой модели путем учета отказов из финальной FMEA-таблицы, а также отобранных средств и процедур обеспечения отказоустойчивости, как это показано на рис. 2.

В рамках второго подхода каждая процедура детализации включает операции учёта новых событий-отказов, оптимальный выбор и учет средств обеспечения отказоустойчивости (рис. 3).

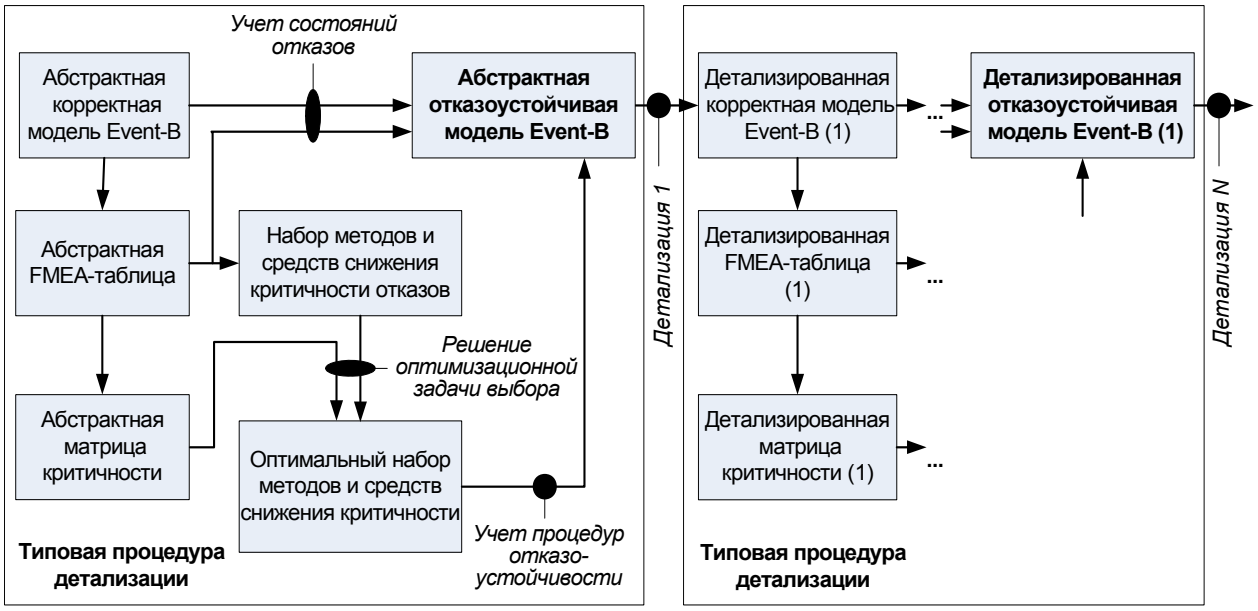

Рис. 3. Процедура детализации отказоустойчивой модели Event-B

Такой подход позволяет получать корректную модель отказоустойчивой системы на всех промежуточных этапах детализации, однако является более трудоемким.

8. Заключение

8.1. Обсуждение результатов

В данной работе проанализированы реальные ограничения и возможности расширения границ применения формального метода Event-B для разработки отказоустойчивых систем. Введение понятий полноты и минимальности множеств инвариантов, анализ ошибок первого и второго рода, которые могут иметь место при их разработке и использовании, а также введение соответствующих метрик и показателей обеспечивают формирование доказательной базы для оценивания реального уровня надежности (а при определенных условиях и гарантоспособности) систем, проектируемых с использованием метода Event-B и других формальных методов.

Следует учитывать, что матрица $\sum$ может быть преобразована из булевой в матрицу, элементы которой принимают значения из диапазона от нуля до единицы, характеризуя достоверность «покрытия» (контроля), обеспечиваемого соответствующим инвариантом.

Одним из важных результатов является определение условий и базовых процедур перехода от модели Event-B корректной системы к модели отказоустойчивой системы, что дает возможность уменьшить пространство проектных действий, не верифицируемых формальными, математически доказательными процедурами.

Совместное использование формального метода Event-B и методов анализа отказов, в первую очередь, анализа видов и последствий критических отказов FME(C)A и его модификаций позволяет расширить конструктивное использование формальных методов, распространив их возможности на системы, критичные к отказам, обусловленным как проектными, так и физическими дефектами, а также дефектами взаимодействия (информационного и физического).

Более того, при определенных условиях такое комплексирование может быть использовано для создания так называемых resilient systems [18] – систем, устойчивых не только к отказам, а и к

изменениям требований и параметров внешней среды (системам, способным эволюционировать в реальном времени – real-time evolvable systems). В этом случае расширяется множество допустимых событий и соответствующих им инвариантов.

8.2. Направления дальнейших исследований

Таким образом, к основным направлениям дальнейших исследований и разработок в данном направлении следует отнести:

– детализацию вариантов и процедур комплексирования метода Event-B и методов анализа отказов (FME(C)A, HAZOP, FTA и их модификаций);

– развитие оценочной базы – метрик множеств инвариантов для расчета показателей надежности и гарантоспособности;

– создание соответствующей инструментальной поддержки, интегрирующей существующие средства формальной разработки и анализа (платформа Event-B «Rodin» [19], утилиты FME(C)A-анализа [15] и др.).

СПИСОК ЛИТЕРАТУРЫ

1. Proof in VDM: A Practitioners Guide / [Bicarregui J., Fitzgerald J., Lindsay P. et al.] – Berlin: Springer-Verlag, 1994. – 362 p.
2. Diller A. Z: An Introduction to Formal Methods / Diller A. – Wiley, 1994. – 354 p.
3. Abrial J.-R. The B-Book: Assigning Programs to Meanings / Abrial J.-R. – Cambridge University Press, 1996. – 853 p.
4. Abrial J.-R. Modeling in Event-B: System and Software Engineering / Abrial J.-R. – Cambridge University Press, 2009. – 586 p.
5. Clarke E. Model Checking / Clarke E., Grumberg O., Peled D. – The MIT Press, 1999. – 314 p.
6. Отказобезопасные информационно-управляющие системы на программируемой логике / Под ред. В.С. Харченко, В.В. Скляра. – Харьков: Нац. аэрокосмический ун-т «ХАИ», НПП «Радий», 2008. – 380 с.
7. Romanovsky A. A Looming Fault Tolerance Software Crisis? / A. Romanovsky // ACM SIGSOFT Software Engineering Notes. – 2007. – Vol. 32, Is. 2. – P. 1 – 4.
8. Тэллес М. Наука отладки / М. Тэллес, Ю. Хсих; пер. с англ. – М.: Кудиц-образ, 2003. – 560 с.
9. Зыков С. Защита отказала. Авария на Саяно-Шушенской ГЭС длилась больше часа / С. Зыков // Российская газета (Центральный выпуск). – 26 августа 2009 г. – № 4982. – Режим доступа: http://www.rg.ru/2009/08/26/avaria-kutin.html.
10. Основы технической диагностики / [В.В. Карибский, П.П. Пархоменко, Е.С. Согомонян, В.Ф. Халчев]. – М.: Энергия, 1976. – Кн. 1. – 464 с.
11. Яблонский С.В. Введение в дискретную математику / Яблонский С.В. – М.: Наука, 1979. – 271 с.
12. F(I)MEA-Technique of Web Services Analysis and Dependability Ensuring / A. Gorbenko, V. Kharchenko, A. Furmanov [et al.]; A. Romanovsky [et al.] (eds.) // Rigorous Development of Complex Fault-Tolerant Systems, LNCS 4157. – Berlin, Heidelberg: Springer-Verlag, 2006. – P. 153 – 167.
13. Extended Dependability Analysis of I&C Systems by FME(C)A-technique: Models, Procedures, Application / E. Komari, V. Kharchenko, E. Babeshko [et al.] // Proc. 4th Int. Conf. on Dependable Computer Systems (DepCoS'09). – 2009. – P. 25 – 32.
14. Analysis Techniques for System Reliability – Procedure for Failure Modes and Effects Analysis (FMEA), IEC 60812. – 2006. – 41 p.
15. Харченко В.С. Комплексный анализ гарантоспособности информационно-управляющих систем и инфраструктур / В.С. Харченко, И.Э. Комари // FME(C)A-модели и информационная технология: зб. наук. пр. – – К.: НАУ, 2008. – Вип. 1 (23). – С. 92 – 97.
16. Тарасюк О.М. Формальные методы разработки критического программного обеспечения. Лекционный материал / Учеб. пособие / О.М. Тарасюк, А.В. Горбенко; под ред. В.С. Харченко. – Харьков: Нац. аэрокосм. ун-т «ХАИ», 2008. – 214 с.
17. Mosleh A., Bari R. Probabilistic Safety Assessment and Management / A. Mosleh, R. Bari. – Springer-Verlag, 1999. – 2933 p.
18. Hollnagel E. Resilience engineering: concepts and precepts / E. Hollnagel, D. Woods, N. Leveson. – Ashgate, 2006. – 397 p.
19. Abrial J.-R. A System Development Process with Event-B and the Rodin Platform / J.-R. Abrial // LNCS 4789. Formal Methods and Software Engineering. – 2007. – P. 1 – 3.

Стаття надійшла до редакції 28.10.2009