УДК 004.415:681.3

ИМИТАЦИОННАЯ МОДЕЛЬ РЕЗЕРВИРОВАННЫХ ИНФОРМАЦИОННЫХ И УПРАВЛЯЮЩИХ СИСТЕМ С УНИФИЦИРОВАННЫМИ ПРОЦЕДУРАМИ ВОССТАНОВЛЕНИЯ

к.т.н. В.В. Скляр, А.Х. Аль Тарази
(представил д.т.н., проф. В.С. Харченко)

Изложены принципы разработки и функциональные возможности имитационной модели, предназначенной для исследования алгоритмов восстановления решающих устройств резервированных информационных и управляющих систем (ИУС).

Постановка проблемы. Одним из основных методов повышения надежности современных цифровых систем является резервирование: структурное, функциональное, информационное, временное, версионное и их различные комбинации [1]. Метод резервирования широко используется в критических ИУС для АЭС, аэрокосмических и транспортных комлексов и т.п., где его применение регламентировано рядом международных и национальных стандартов. При этом решение задачи обеспечения требуемого быстродействия, точности, достоверности и других важных характеристик ИУС существенным образом зависит от применяемых решающих устройств (РУ) или подсистем, объединяющих выходы каналов системы. В резервированных ИУС сложность РУ может колебаться от простого коммутатора до отдельного конструктива со сложной пространственной и логической структурой. Таким образом, влияние РУ на надежность и безопасность системы в целом требует обоснованного выбора различного рода параметров РУ, в первую очередь, алгоритма его работы.

Анализ литературы. Основные подходы к имитационному моделированию различных аспектов программного обеспечения (ПО) изложены в монографиях [2, 3]. Эти подходы были развиты в ряде современных зарубежных работ. В [4, 5] разработана методика тестирования ПО со специально внесенными дефектами (Fault Injection), работа [6] посвящена особенностям моделирования коммерческого ПО (COTS – Commercial Off The Shelf), в [7] усовершенствован подход к разработке генератора случайных чисел. В [8] предложена методика моделирования и определения показателей безотказности программных средств методом Монте-Карло.

В [9] исследованы алгоритмы восстановления вычислительного процесса. Проведенный анализ литературы показал, что известные методики моделирования не учитывают логику работы РУ резервированных ИУС.

Целью статьи является разработка имитационной модели (ИМ), предназначенной для исследования резервированных ИУС и позволяющей учесть алгоритмы восстановления, реализуемые решающими подсистемами.

Принципы разработки ИМ. Для реализации ИМ было разработано специальное инструментальное средство (ИС) "Solver Devices Simulation" (SDS). При разработке технического задания (ТЗ) на ИС использовались положения стандарта IEEE Std 830-1998 “IEEE Recommended Practice for Software Requirements Specifications”. Разработка функциональных требований к ИС SDS и проект ИС реализовывались с использованием UML. Из нефункциональных требований в ТЗ были выделены требования к программной документации и к пользовательскому интерфейсу, а также дополнения к требованиям на сетевую версию ИС SDS. Программирование выполнялось на языке C++.

Основной цикл работы ИМ представлен на рис. 1.

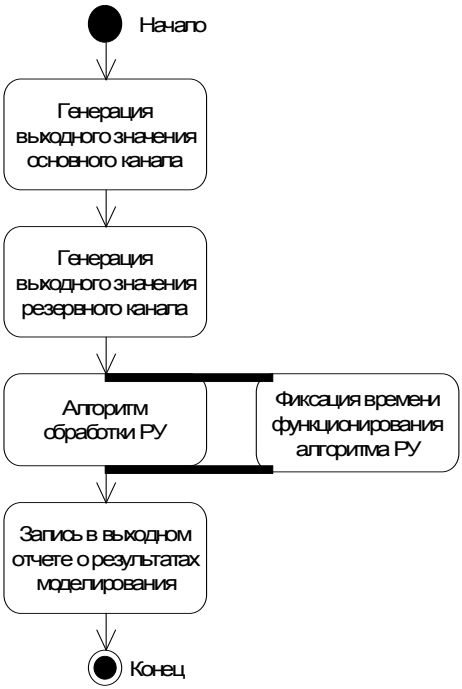

Рис. 1. Основной цикл работы ИС "Solver Devices Simulation"

В статье описана версия ИМ для дублированной системы, однако, те же принципы могут быть использованы для моделирования систем любой конфигурации. Выходной результат представляется в виде последовательности двоичных символов различной длины. В простейшем случае он может быть представлен в виде единичного бита ("0" или "1"). При нормальной работе системы (совпадение результатов каналов) один канал является основным, другой – резервным ("горячий резерв"). Выходной результат системы берется из основного канала. В начале работы основным является первый канал, при несовпадении результатов и отработке алгоритма восстановления основной и резервный каналы могут многократно меняться местами.

Первоначально генерируется выходной результат основного канала.

В простейшем случае его генерация представляет собой "заглушку", поэтому численное значение результата не имеет физического смысла. Однако, в ИМ предусмотрена возможность подачи на вход программного имитатора выходных результатов реальных программ. Выходной результат резервного канала генерируется на основе результата основного канала. При этом возможны следующие варианты генерирования результатов резервного канала (задаются пользователем):

– *совпадение* – выходной результат резервного канала тождественно равен выходному результату основного канала;

– *несовпадение* – выходной результат резервного канала тождественно неравен выходному результату основного канала; несовпадение результатов формируется в последнем бите обрабатываемой последовательности для реализации консервативного подхода к оценке времени работы РУ;

– *случайное несовпадение* – искажение в выходном результате резервного канала генерируется методом случайных чисел, распределенных по равномерному закону; частота искажений должна задаваться пользователем;

– *результаты генерируются реальными программами* – в этом случае генератор результатов выполняет "транзитную", а не управляющую функцию (см. выше); кроме того, в этом случае должен дополнительно генерироваться эталонный результат.

Диаграмма состояний РУ представлена на рис. 2.

Тип алгоритма восстановления должено устанавливаться пользователем. Алгоритм восстановления задействуется при несовпадении результатов в каналах. Время работы РУ фиксируется при помощи таймера.

Алгоритмы работы РУ. РУ должно реализовать один из двух основных методов восстановления (реконфигурации): без применения или с применением диагностики.

Реконфигурация (перемена местами ведущего и резервного каналов) может реализовываться одним из следующих способов:

– без смены канала, однако, включается счетчик несовпадений, при количестве несовпадений выше заданного числа – переход на резервный канал (или диагностика);

– со «мгновенной» сменой канала – переход на резервный канал при первом же несовпадении результатов;

– со случайным выбором канала:

– равновероятный выбор канала; номер ведущего канала определяется с равной вероятностью генератором случайных чисел;

– неравновероятный выбор канала; выбирается тот канал, в котором было меньше сбоев;

– реконфигурация по результатам диагностики; если произошел сбой, и

по результатам диагностики оказались работоспособными оба канала, то выбор может осуществляться одним из указанных выше способов. Диагностические алгоритмы представляются в виде "черного ящика", а в программном коде в виде "нулевого оператора", т.е. при несовпадении результатов никаких действий не выполняется, а результаты диагностики заведомо считаются положительными (формируется положительный флаг). В полной мере моделирование диагностики может быть выполнено для сетевой версии ИМ.

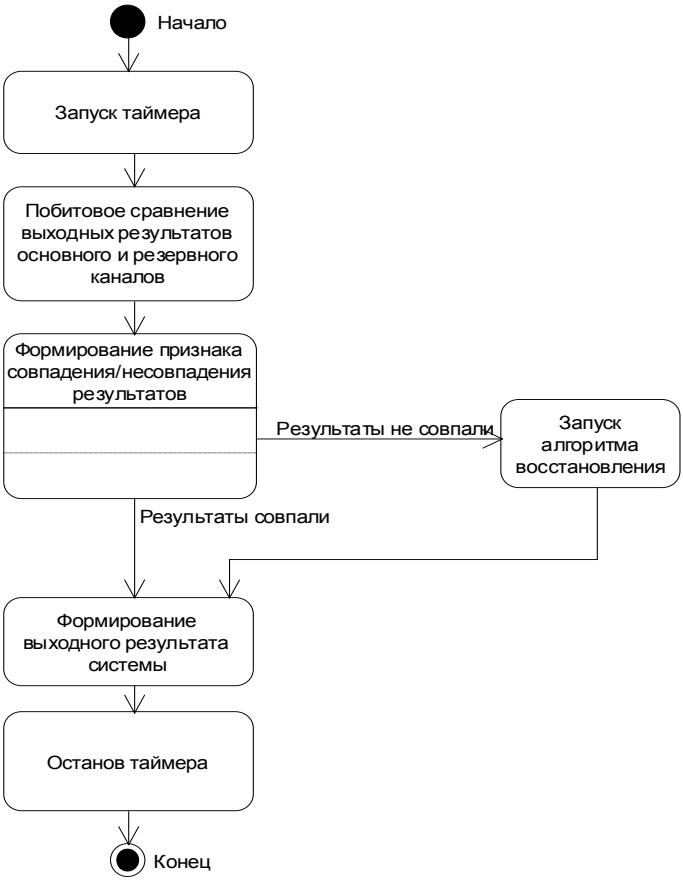

Рис. 2. Диаграмма состояний решающего устройства

Диагностика может реализовываться одним из следующих способов:

– с прерыванием (синхронная для обоих каналов);

– без прерывания (асинхронная); прерывание выполняется, если по результатам диагностики обнаружилась ошибка реконфигурации (переключение на неработоспособный канал);

– диагностика обоих каналов;

– диагностика основного канала;

– диагностика резервного канала.

Кроме того, при моделировании алгоритмов работы РУ необходимо учитывать, что в ходе восстановления реализуется либо повторный пересчет результата, либо пропуск сбойного такта. Еще одним классификационным признаком алгоритма восстановления является асинхронизм или синхронизм каналов.

Таким образом, могут быть реализованы следующие варианты повторного пересчета (пропуска) такта:

– синхронный (прерывание осуществляется для обоих каналов):

– синхронный пересчет – до совпадения результатов; однако, при этом включается счетчик несовпадений, в случае превышения заданного числа несовпадений осуществляется реконфигурация с диагностикой или без диагностики;

– синхронный пропуск такта – игнорирование данных, для которых произошел сбой и пересчет для новых данных; однако, при этом включается счетчик несовпадений, в случае превышения заданного числа несовпадений осуществляется реконфигурация с диагностикой или без диагностики;

– асинхронный с естественным выравниванием процессов по результатам реконфигурации или диагностики; при этом учитываются результаты только основного канала, который продолжает работу независимо от резервного, а резервный канал выступает в роли "догоняющего":

– асинхронный пересчет;

– асинхронный пропуск такта.

Итак, проведенное исследование возможных вариантов реализации алгоритмов восстановления позволило выделить для них четыре существенных классификационных признака:

– способ реконфигурации;

– способ диагностирования;

– способ восстановления работы (пропуск такта или повторный пересчет);

– синхронизм/асинхронизм.

В ТЗ на ИС SDS виды алгоритмов восстановления были заданы в виде матрицы, элементами которой являются булевы переменные, отражающие применение в данном алгоритме того или иного способа реализации каждого из четырех указанных классификационных признаков. В результате для дублированной системы было получено 66 вариантов реализации алгоритмов восстановления.

Выводы. В результате проведенного исследования осуществлена

систематизация алгоритмов работы решающих устройств резервированных цифровых систем. Разработана имитационная модель на основе ИС "Solver Devices Simulation" для исследования работы резервированных систем, в которых применяются РУ на основе полученных алгоритмов.

Дальнейшие исследования целесообразно направить на постановку серии экспериментов по изучению различных алгоритмов работы РУ, в том числе, для оценки времени их функционирования и эффективности по критерию "время работы/затраты на реализацию".

ЛИТЕРАТУРА

1. *Харченко В.С., Жихарев В.Я., Илюшко В.М., Нечипорук Н.В. Многоверсионные системы, технологии, проекты. – Х.: НАКУ «ХАИ», 2003. – 486 с.*
2. *Тоценко В.Г., Александров А.В., Парамонов Н.Б. Корректность, устойчивость, точность программного обеспечения. – К.: Наук. думка, 1990. – 200 с.*
3. *Laprie J.-C. Dependability Handbook. LAAS Report n 98 – 346. – Toulouse, France: LAAS, 1998. – 365 p.*
4. *Deconinck G., De Florio V., Botti O. Software-Implemented Fault-Tolerance and Separate Recovery Strategies Enhance Maintainability // IEEE Transactions on Reliability. – 2002. – Vol. 51. – № 2. – P. 158 – 165.*
5. *Constantinescu C. Experimental Evaluation of Error-Detection Mechanisms // IEEE Transactions on Relaibility. – 2003. – Vol. 52. – № 1. – P. 53 – 57.*
6. *Arlat J., Fabre J-C., Rodriguez M., Salles F. Dependability of COTS Microkernel-Based Systems // IEEE Transactions on Computers. – 2002. – Vol. 51. – № 2. – P. 138 – 163.*
7. *Bucci M. et al. A High-Speed Oscillator-Based Truly Random Number Source for Cryptographic Applications on a Smart Card IC // IEEE Transactions on Computers. – Vol. 52. – № 4. – P. 403 – 409.*
8. *Харченко В.С., Скляр В.В. Моделирование и оценка безотказности необслуживаемых компьютерных систем управления с многоверсионными программными средствами // Электронное моделирование. – 2001. – Т. 23. – № 4. – С. 69 – 81.*
9. *Фурманов А.А., Харченко В.С., Аль-Тарази А.Х. Алгоритмы восстановления вычислительного процесса в многоверсионных программных системах // Зб. наук. праць Кіровоградського державного технічного університету «Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація». – Кіровоград: КДТУ, 2002. – Вип. 11. – С. 12 – 15.*

Поступила 2.02.2004

***СКЛЯР Владимир Владимирович**, канд. техн. наук, старший научный сотрудник Государственного научно-технического центра ядерной и радиационной безопасности. В 1992 году окончил Харьковское ВВКИУ РВ. Область научных интересов – методы стандартизации, оценки и обеспечения надежности и безопасности аппаратных и программных средств компьютерных систем.*

***АЛЬ ТАРАЗИ Ахмед Хусни**, аспирант кафедры компьютерных систем и сетей Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ». В 2001 году окончил НТУ «КПИ». Область научных интересов – методы моделирования и оценки*

отказоустойчивости компьютерных систем и сетей.