Вісник Харківського національного університету
Серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»
№ 833, 2008, с.231-246

УДК 621.039:004.05

(n,m)-версионные системы: таксономия, модели и технологии

А. А. Сиора[1)], В. В. Скляр[2)], В. С. Харченко[3)]
[1)]*Научно-производственное предприятие «Радий»,*
[2)]*Государственный научно-технический центр ядерной и радиационной безопасности,*
[3)]*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ»,Украина*

Basic concepts of diversity as key approach to decreasing probability of common cause failure of computer-based systems are specified. The taxonomic scheme including interconnected concepts of a version, multiversity, version redundancy, multi-version (n,m-version) system, multi-version technology and project is described. Known classifications of version redundancy kinds are analyzed and it's suggested to use three-space matrix with coordinates "lifecycle stages and processes – level of presentation – multi-version project decisions" called "cube of multiversity". Models of n,m-version systems are generalized. It's specified an indicator of cost-effectiveness of using version redundancy as specific increasing of dependability.

1. Введение. Общая постановка задачи

1.1. Актуальность. При решении проблем обеспечения надежности и безопасности критических и бизнес-критических компьютерных систем обработки информации и управления (далее компьютерных систем) используется принцип диверсности, при котором применяется несколько разных вариантов реализации резервных каналов. Необходимость использования принципа диверсности диктуется тем, что только при его применении появляется реальная возможность противостоять наиболее опасному с точки зрения их последствий для резервированных структур (или избыточных процессов) виду отказов, так называемому отказу по общей причине (ООП) [1]. При таком отказе существует дефект (ошибка), вызывающий потерю работоспособности всех каналов системы (или отрицательный результат выполнения процессов) независимо от их числа резервных каналов (избыточных процессов).

Наиболее вероятным источником ООП для компьютерных систем являются дефекты программных средств, внесенные при разработке, невыявленные при тестировании и верификации и проявляющиеся при определенном наборе входных данных или характеристиках физической или информационной среды. При идентичном исполнении каналов резервированной системы такие дефекты тиражируются и проявляются одновременно, вызывая непарируемый (фатальный) отказ. В настоящее время они составляют весомую долю причин отказов «отказоустойчивых» (резервированных) систем для приложений, где применение принципа диверсности не является нормативным требованием, составляя по некоторым оценкам до 80 процентов причин отказов [2].

В более широком контексте можно говорить как о вероятных причинах ООП не только о дефектах разработки (design faults, включающих как программные дефекты, так и некоторые дефекты производства аппаратных средств), а и о дефектах взаимодействия (interaction faults) системы со средой [3]. Возможность

возникновения ООП вследствие физического дефекта (physical faults) невысока, так как мала вероятность события, при котором возникают идентичные дефекты одноименных элементов аппаратных средств в разных резервных каналах.

Использование принципа диверсности позволяет снизить риски ООП, поскольку в случае проектных дефектов уменьшается вероятность одновременного и однотипного (при дефектах взаимодействия - одновременного) отказа разных каналов (версий).

1.2. Проблема использования нескольких видов диверсности. Несмотря на интенсивные исследования, проводившиеся в Украине [4-6] и за ее пределами такими специалистами как A.Avizienis, J-C.Laprie, B.Randel [3,7,8], B. Littlewood, L. Strizini, P.Popov, A.Romanovsky [9-12], N.Leveson [13] и другими, многолетнее практическое использование принципа диверсности, остается нерешенным ряд традиционных и новых проблем, связанных:

- во-первых, с выбором видов и объема версионной избыточности и вариантов их совместного использования;
- во-вторых, с оценкой уровня реальной диверсности и эффективности ее использования;
- в-третьих, с разработкой архитектур и технологий синтеза систем, устойчивых к причинам, порождающим ООП и др.

В атомной энергетике, где применение принципа диверсности в наиболее ответственных системах обеспечения безопасности реактора является обязательным [1], актуальна проблема обоснования выбора и доказательства достаточности выбранного вида или видов версионной избыточности для снижения рисков ООП до приемлемо низкого уровня.

В известных работах проблематика использования нескольких видов диверсности другими авторами не исследовалась. В то же время на практике, в атомной энергетике, аэрокосмических системах, железнодорожном транспорте, опасных химических производствах, в сервис-ориентированных бизнес-критических системах, основанных на web-технологиях [1,14,15], нашли применение системы, в которых де-факто используется как продуктная, так и процессная диверсность, диверсность и программных, и аппаратных средств. Это обусловлено интеграционными тенденциями развития компьютерных технологий и программной инженерии, которые по многим аспектам «движутся» навстречу друг другу. Пример такой тенденции – активное внедрение программируемых интегральных схем, в которых грань между программными и аппаратными средствами более размыта, чем в микропроцессорных технологиях.

2. Истоки исследования авторов

Данная статья опирается на результаты исследований авторов и их коллег, относящиеся:

- к разработке терминологических и методологических аспектов применения многоверсионности [4,16,17];
- к разработке моделей многоверсионных систем [4,5,18,19], методов детерминированной и вероятностных оценки с использованием метрик диверсности и стандартных показателей, учитывающих эти метрики [5,20,21];

- к разработке и исследованию методов обеспечения требуемого уровня диверсности при различных ограничениях и многоверсионных технологий разработки и реализации гарантоспособных систем для критических и бизнес-критических приложений [5,6,15,22-25];

- к анализу результатов применения многоверсионных систем и технологий в программно-технических комплексах АЭС [22,26].

В [18] было впервые введено понятие *мультидиверсной* системы как системы с несколькими (двумя) видами версионной избыточности. Однако, это понятие и модель носят частный характер, поскольку речь идет о системах с двумя видами версионной избыточности и фиксированных схемах ее использования.

3. Цель работы

Данная работа преследует несколько взаимосвязанных целей:

- уточнить терминологические аспекты использования принципа диверсности и предложить более полную таксономическую схему (раздел 4);

- проанализировать классификационные схемы видов версионной избыточности, учитывая возможности современных компьютерных технологий и опыт разработки многоверсионных систем и проектов (раздел 5);

- обобщить модели многоверсионных систем при использовании нескольких видов версионной избыточности – (*n,m)-версионных систем* (раздел 6) и технологий их создания (раздел 7).

4. Таксономия многоверсионности

4.1. Терминологические замечания. Термин *диверсность* является калькированным переводом английского термина *diversity* [7]. В русскоязычной и украиноязычной научно-технической литературе ему могут соответствовать термины *разнообразие*, *версионность* или, *многовариантность* или *многоальтернативность* [4,5] и Наибольшее распространение получили термины *разнообразие*, *диверсность* и *многоверсионность* [1,19].

При этом термин *диверсность* имеет двойную нагрузку: с одной стороны, в широком смысле, он указывает на сам факт использования процессно-продуктного разнообразия, с другой, – в узком буквальном толковании – фиксирует число используемых вариантов (версий). С учетом этого обстоятельства строже использовать термин *многоверсионность,* который в частном случае – при двух версиях совпадает (или может быть заменен) с термином *диверсность.*

Использование в качестве базового термина *многоверсионность* позволяет устранить еще одну терминологическую неоднозначность, относящуюся к системам, в которых используется указанный принцип. Такие системы могут быть многоверсионными (при числе версий, большем двух) и двухверсионными (при двух версиях)*,* В последних одна из подсистем часто называется *основной* (primary system), а другая – *диверсной* (secondary или diverse system).

Следует заметить, что идентичным по отношению к термину *двухверсионная система*, формально говоря, является термин *диверсионная система*, однако он носит двусмысленный характер и не используется.

4.2. Таксономическая схема. Таким образом, имеем таксономическую схему, включающую следующие элементы (рис.1):

- *версия* – вариант адекватной по решаемой задаче реализации продукта (архитектуры, набора аппаратных или программных средств и др.) или процесса;

- *версионная избыточность* (ВИ) – вид избыточности, отличающийся использованием разных версий, который, в свою очередь, может иметь несколько подвидов (типов);

- *многоверсионность* или *диверсность* (МВ) – *принцип*, предусматривающий использование нескольких версий; этот принцип подразумевает выполнение одной и той же задачи (реализацию продукта или процесса) двумя и более способами и обработку получаемых данных для контроля, выбора или формирования конечного или промежуточного результатов и принятия решения об их дальнейшем использовании;

- *многоверсионная система* (МВС) – система, в которой используется несколько версий-продуктов;

- *мультидиверсная система* (МДВС) – МВС, в которой используется два вида версионной избыточности;

- *многоверсионная технология* (МВТ) – совокупность взаимосвязанных правил и проектных действий, в которой в соответствии со стратегией МВ используется несколько версий-процессов, приводящих к получению двух и более промежуточных или конечных продуктов; при этом для разработки многоверсионной системы должна быть использована МВТ, для разработки одноверсионной системы – могут быть использованы как многоверсионная, так и одноверсионная технология;

- *многоверсионный проект* (МВП) – проект, в котором применяется многоверсионная технология (используется версионная избыточность процессов), приводящая к созданию одно- или многоверсионной системы (реализации версионной избыточности продуктов);

- *стратегия многоверсионности* – совокупность общих критериев и правил, определяющих принципы формирования и выбора видов версионной избыточности и многоверсионных технологий при разработке МВП.

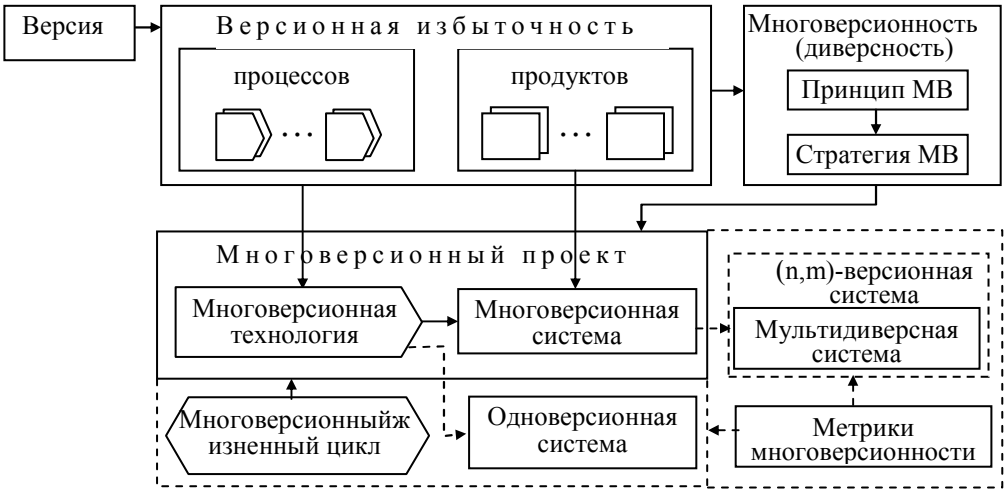


Рис.1. Таксономическая схема многоверсионности

Кроме того, важными элементами таксономии многоверсионности являются понятия *многоверсионного жизненного цикла* – жизненного цикла МВП и *метрик многоверсионности* – показателей степени независимости версий МВП.

Часто термины версионная избыточность (вид), многоверсионность и диверсность (принцип) могут использоваться как синонимы с учетом того, что принцип МВ реализуется посредством применения соответствующего вида ВИ.

Далее будет дано более строгое формальное толкование рассмотренных понятий, в частности, МВС, в которых используется n версий с m видами версионной избыточности – (*n,m)-версионных систем*.

5. Классификация версионной избыточности

Версионная избыточность определяет существо многоверсионной технологии и системы, поэтому важно уточнить классификацию видов ВИ. Различные варианты классификационных схем приведены в [6,14,27-30].

5.1. Системный уровень. Наиболее полная классификация ВИ на системном уровне предложена в [14,27] и включает шесть типов:

1) *субъектная ВИ (human diversity)* достигается за счет использования различных исполнителей (команд исполнителей) работ, а именно:

- разработки проекта несколькими различными организациями;
- разработки проекта несколькими командами в пределах одной организации;
- разработки проекта или его частей несколькими инженерами в одной команде под общим управлением одного менеджера;
- тестирования одной или нескольких версий проекта несколькими тестировщиками;
- верификации проекта несколькими несколькими аудиторами или сертификационными органами;

2) *ВИ проектирования (design diversity)* базируется на применении при разработке аппаратных, программных средств и систем в целом различных:

- концептуальных подходов;
- технологий и моделей жизненного цикла;
- архитектур;

3) *программная ВИ (software diversity)* предполагает использование различных версий программного обеспечения (ПО), разнообразие которых может быть обусловлено применением различных видов версионной избыточности применительно к этапам разработки ПО, а именно:

- различных алгоритмов и логики их реализации;
- различных архитектур ПО;
- различных языков программирования;
- различных операционных систем и баз данных;
- различной синхронизации и порядка выполнения;

4) *функциональная ВИ (functional diversity)* достигается за счет реализации версиями различной функциональности для достижения идентичного результата, в частности:

- различных целей и функций;
- различных алгоритмов или логики управления;
- различных исполнительных механизмов;
- различных шкал времени;

5) *сигнальная ВИ (signal diversity)* обеспечивается путем использования различных входных параметров (различных источников данных), определяющих алгоритм функционирования системы, реализуемых путем измерения:

- различных параметров процессов с использованием различных физических эффектов;
- различных параметров процессов с использованием идентичных физических эффектов;
- идентичных параметров процессов с использованием различных множеств идентичных датчиков;

6) *ВИ оборудования (equipment diversity)* достигается за счет использования:

- оборудования от различных производителей, использующих разные или идентичные проектные решения (технологии);
- различной элементной базы (в первую очередь, процессоров);
- различных печатных плат и технологий трассировки;
- различной шинной организации.

5.2. Программная многоверсионность. В [6,28] предложено детализировать программную МВ по признакам разнообразия:

- *моделей жизненного цикла, принципов и процессов разработки;*
- *ресурсов и средств;*
- *проектных решений*.

В свою очередь, например, модель ЖЦ двухверсионной системы может включать две различные модели ЖЦ для каждой из версий ПО:

а) итерационную модель с максимальным набором процессов обеспечения качества (возможно также диверсных) для основной версии;

б) водопадную модель с обратной связью, включающую минимально необходимый набор процессов обеспечения качества для резервной версии (за счет чего уменьшается её стоимость по сравнению с основной версией).

Многоверсинность ресурсов и средств включает разнообразие:

а) исполнителей;

б) используемых языков и нотаций;

в) инструментов разработки.

Многоверсионность проектных решений детализируется с учетом разнообразия:

а) архитектур и платформ;

б) протоколов;

в) форматов и способов представления данных.

5.3. Многоверсионность систем на ПЛИС. В соответствии с [6,28] при реализации многоверсионности на системном уровне различают также:

- *по степени охвата МВ*: системы с полной и частичной ВИ;
- *по глубине МВ*: системы с общей и раздельной версионной избыточностью.

Кроме того, в этой работе дана детальная классификация ВИ во многоверсионных системах, использующих технологии ПЛИС. В качестве класификационных признаков выбраны:

- *ВИ элементной базы*, включающая разнообразие:

а) фирм-производителей;

б) технологий реализации элементной базы;

в) семейств элементной базы;

г) видов элементов, относящихся к одному семейству;

- *ВИ инструментальных средств* (САПР), включающая разнообразие:

а) разработчиков инструментальных средств;

б) видов (наименований) инструментальных средств;

в) конфигурации (состава и версии) инструментальных средств языков проектирования;

- *ВИ языков программирования* (САПР) на базе:

а) одновременного использования графического языка представления схем и языка программирования (языка описания аппаратуры)

б) использования разных языков программирования (или языков описания аппаратуры)

- *ВИ языков проектирования спецификаций* на базе одновременного использования разных языков спецификаций.

5.4. Концептуальная многоверсионность. В работах [29,30] предложены варианты классификации МВ на концептуальном уровне в соответствии с:

- *системами счисления*, применяемыми для представления и обработки информации в каналах МВС, когда параллельно используются версии, базирующиеся на позиционной и непозиционной (на основе систем остаточных классов (СОК)) системах; в этом случае идет речь о МВС с версионно-информационной и версионно-информационно-структурной избыточностью;

- *подходами к разработке к системы,* когда используются версии, строящиеся на основе концепций «белого» и «черного ящиков».

Применительно к многоверсионным системам на ПЛИС, концепцию «черного ящика» можно реализовать на основе аппарата генетических алгоритмов (ГА) [31]. Кроме того, в [30] предложено для таких систем использовать признак, по которому различают:

- *внутреннюю многоверсинность* (в рамках САПР-ориентированного и ГА-ориентированного подходов);

- *внешнюю многоверсинность*, когда две версии реализуются в соответствии с САПР- и ГА- подходами.

При внутренней МВ для систем с версионно-информационной избыточностью каналы строятся на основе использования, например, разных наборов оснований СОК, разных схем реализации процедур обработки информации в СОК и т.д.[24,28].

5.5. Куб многоверсионности. Анализ рассмотренных классификаций позволяет о утверждать, что:

- они представлены классификационными схемами смешанного фасетно-иерархического или матричного (сетевого) типов;

- наиболее детальной и системной является классификация, представленная в [14,27], хотя в ней не выдержан в полной мере принцип ортогональности элементов отдельных таксонов; например, пересекающимися и зависимыми являются подмножества проектной и программной, функциональной и сигнальной версионной избытрчности;

- сложность и многообразие продуктной (системных, аппаратных и программных компонент) и процессной (технологий разработки, тестирования и сопровждения) видов ВИ обусловливают высокую размерность задачи выбора вариантов реализации компьютерных систем.

Исходя из этого, можно сделать вывод, что классификацию версионной избыточности целесообразно представить матрицей схемой $VR = || vr_{ijk} ||$ в трехмерном пространстве. Назовем ее *кубом многоверсионности* (рис.2) с

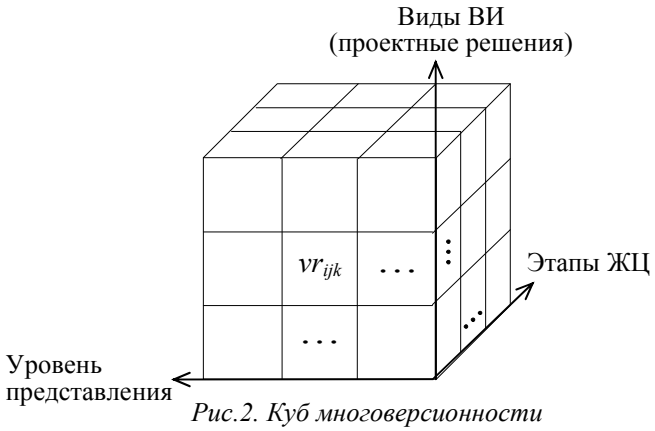


Рис.2. Куб многоверсионности

координатами: «этапы и процессы жизненного цикла системы (от спецификации до валидации и использования по назначению) – уровень проектных решений (от концептуального до компонентного) – конкретные виды версионной избыточности (проектные решения)», которые соответсвуют нижним индексам элементов vr_{ijk} матрицы (куба).

Ниже приведены примеры двумерных классификационных матриц проектных решений, представляющих собой сечения куба многоверисонности. Первая из них (табл.1) - фрагмент для уровня ПО по признаку *разнообразия ресурсов и средств* [28], вторая (табл.2) – фрагмент для *МВС на базе ПЛИС* [6].

Табл.2 содержит варианты решений, базирующихся на совместном использовании нескольких видов избыточности – пп. 1.4.2-1.4.4, 2.3.3 – 2.3.8, 3.3.3 – 3.3.8, 4.2.4-4.2.15, (например, пп. 4.2.4-4.2.15 соответствуют 12 = 4x3 парам, определяемым комбинациями 4 видов разнообразия элементной базы и 3 видов разнообразия инструментальных средств). Следует подчеркнуть, что отдельной задачей является анализ и учет совместимости различных видов ВИ, используемой на одном и разных этапах жизненного цикла (см. п.7.1)

6. Модели (*n,m*)-версионных систем

6.1. Теоретико-множественная модель. Многоверсионная система W описывается пятеркой [5]:

$$W_n = \{X, F, U, V, Z\}, \qquad (6.1)$$

где X, U – входные и выходные алфавиты (сигналы);

$F = \{f_d, d = 1,..., k\}$ – множество выполняемых функций;

$V = \{v_i,\ i = 1,\ldots,\ n\}$ – множество версий с выходными алфавитами (сигналами) $U_1,\ldots,\ U_n$;

Z – функция обработки результатов выполнения версий (отображения U_{id} в U_d при выполнении функции f_d, т.е. $U_d = Z(U_{1d},\ldots,U_{nd})$). Эта функция может описывать операцию мажоритарного голосования (кодового, арифметического, медианного и др.) или более сложную операцию, учитывающую особенности выбора версий с учетом различных вариантов адаптации систем к отказам программных и аппаратных компонент [5,15,32].

Табл.1. Фрагмент сечения куба многоверсионности для ПО

Этапы ЖЦ ПО	Вид ВИ (проектного решения)		
	1. Разнообразие исполнителей	2. Разнообразие используемых языков и нотаций	3. Разнообразие инструментов разработки
1. Разработка требований	1.1. Различные команды аналитиков, экспертов, проводящих верификацию	1.2. Различные нотации (IDEF, UML и т.д.)	1.3. Различные инструменталь-ные средства автоматизации процесса формализации требо-ваний, построения моделей
2. Проектирование	2.1. Различные команды проектировщиков, экспертов, проводящих верификацию и т.д.	2.2. Различные нота-ции, используемые при проектировании (UML, Booch и т.д.)	2.3. Различные средства построения и верификации моделей этапа проектирования
3. Кодирование и отладка	3.1. Различные команды программистов, тестировщиков, экспертов, проводящих верификацию	3.2.1. Различные языки программирования 3.2.2. Различные способы представления результатов тестирования	3.3.1. Различные среды и платформы разработки 3.3.2. Различные библиотеки функций и компонентов 3.3.3. Различные компиляторы 3.3.4. Различные средства авто-матизации тестирования
4. Интеграция	4.1. Различные команды интеграторов, экспертов, проводящих верифи-кацию и валидацию		4.3. Различные средства поддержки процесса интеграции
5. Эксплуатация	5.1. Различные команды поддержки и обслуживания		5.3. Различные средства поддер-жки проверки и обслуживания

В [18] предложена модель мультидиверсной системы, которая содержит две версии v_1, v_2, полученные с использованием нескольких видов версионной избыточности.

При этом считается, что устранение одного из этих видов избыточности не делает версии тождественными (тривиальными [5]), т.е., если существует хотя бы один дефект проектирования, который является относительным для каждого варианта версионной избыточности применительно к версиям v_1, v_2.

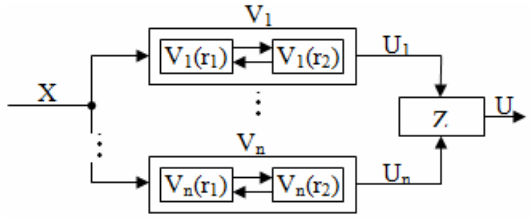


Рис.3. Структура (n,2)-версионной системы

Такая система может обозначаться (называться) как *(n,2)-версионная система.* Ее структура представлена на рис.3, а примером являются (2,2)-системы, в которых применяется две версии, отличающиеся аппаратной платформой (микросхемой ПЛИС) и средствами реализации проекта.

Табл.2. Фрагмент сечения куба многоверсионности для МВС на базе ПЛИС

Этапы жизненного цикла систем на базе ПЛИС	Вид ВИ (проектного решения)			
	1. Разнообразие элементной базы (ЭБ)	2. Разнообразие инструментальных средств (ИС)	3. Разнообразие языков проектирования (ЯП) (описания аппаратуры (ЯОА))	4. Разнообразие языков разработки спецификаций (ЯС)
1. Разработка схем представления алгоритмов формирования сигналов		1.2.1. Разные разработчики ИС 1.2.2. Разные виды ИС 1.2.3. Разные конфигурации ИС		1.4.1. Разнообразие ЯС 1.4.2-1.4.4. Комбинации пар, определяемых разнообразием ИС и ЯС
2. Разработка программных моделей алгоритмов формирования сигналов в среде проектирования		2.2.1. Разные разработчики ИС 2.2.2. Разные виды ИС 2.2.3. Разные конфигурации ИС	2.3.1. Одновременное использование графического языка представления схем и ЯП (ЯОА) 2.3.2. Разные ЯП (ЯОА) 2.3.3 – 2.3.8. Комбинации пар, определяемых разнообразием ИС и ЯП (ЯОА)	
3. Разработка программной модели системы в среде проектирования		3.2.1. Разные разработчики ИС 3.2.2. Разные виды ИС 3.2.3. Разные конфигурации ИС	3.3.1. Одновременное использование графического языка представления схем и ЯП (ЯОА) 3.3.2. Разные ЯП (ЯОА) 3.3.3 – 3.3.8. Комбинации пар, определя-	

			емых разнообразием ИС и ЯП (ЯОА)	
4. Имплементация программной модели системы в программируемый компонент	4.1. Разные фирмы-производители ЭБ 4.2. Разные технологии реализации ЭБ 4.3. Разные семейства ЭБ 4.4. Разные виды элементов, относящихся к одному семейству ЭБ	4.2.1. Разные разработчики ИС 4.2.2. Разные виды ИС 4.2.3. Разные конфигурации ИС 4.2.4 -4.2.15. Комбинации пар, определяемых разнообразием ЭБ и ИС		

Если $R = \{r_q, q = 1,..., m\}$ – множество видов версионной избыточности, θ – их отображение на элементы множества $v_j(\Delta R_j) \in V$, $\Delta R_j \subset R$, то *(n,m)-версионная система* и описывается следующим выражением, обобщающим (6.1):

$$W_{n,m} = \{X, F, U, V, R, \theta, Z\}. \quad (6.2)$$

6.2. Модели многоверсионных автоматов. В простейшем случае, когда функции исходной версии описываются моделью комбинационного автомата *(n,m)*-версионная система представляется следующим выражением:

$$U_d = Z(F_{1d}(X),...,F_{nd}(X)), \quad (6.3)$$

где F_{id} – вариант отображения входного сигнала X в выходной U_{1d} с учетом использования подмножества видов версионной избыточности $\Delta R_{id} \subset R$.

Разные виды ВИ для многоверсионных комбинационных автоматов (МВКА) могут быть получены с учетом разнообразия:

- исходных форм представления функций (табличный, числовой);
- используемых методов минимизации логических функций (карты Карно, Квайна – Мак-Класки, поразрядного сравнения и др.);
- представления минимальных (тупиковых) форм (дизъюнктивные, конъюнктивные, скобочные);
- базисов реализации (И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ и др.).

Для канонических моделей последовательностных автоматов диверсифицируются варианты функций переходов-выходов, реализации памяти. Вид функций переходов δ и выходов λ многоверсионного последовательностного автомата (МВПА) с памятью зависит от степени и глубины МВ (см. п.5.3).

Если F_d реализуется посредством последовательностного автомата A_d, то для его описания необходимо задать начальное состояние Y_{d0}, а также функции переходов δ_d и функцией выходов λ_d, т.е.

$$F_d = \{Y_{d0}, \delta_d, \lambda_d\}. \quad (6.4)$$

Если имеем автомат Мили с полной общей МВ, то эти функции описываются с учетом (6.3, 6.4) следующим образом:

$$Y_d(t+1) = Z_\delta\{\delta_{1d}[X(t), Y(t)], \ldots, \delta_{nd}[X(t), Y(t)]\}; \quad (6.5)$$

$$U_d(t) = Z_\lambda\{\lambda_{1d}[X(t), Y(t)], \ldots, \lambda_{1d}[X(t), Y(t)]\}, \quad (6.6)$$

где Z_δ, Z_λ - функции формирования общей функции переходов и выходов по функциям переходов и выходов, реализуемым отдельными версиями.

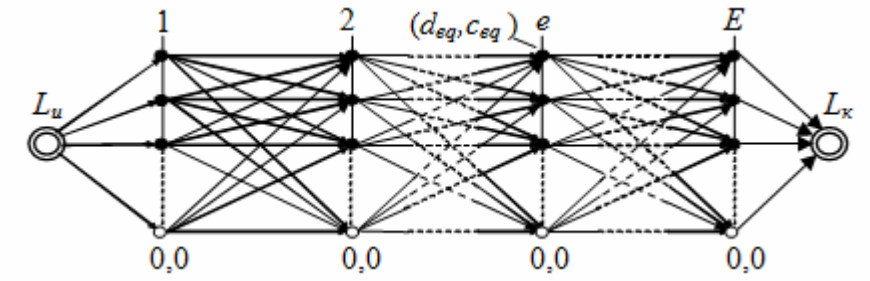
При использовании автоматной модели Глушкова, МВПА может описываться вариантами многоверсионной реализации управляющего и операционного автоматов [5].

7. Многоверсионные технологии и их выбор

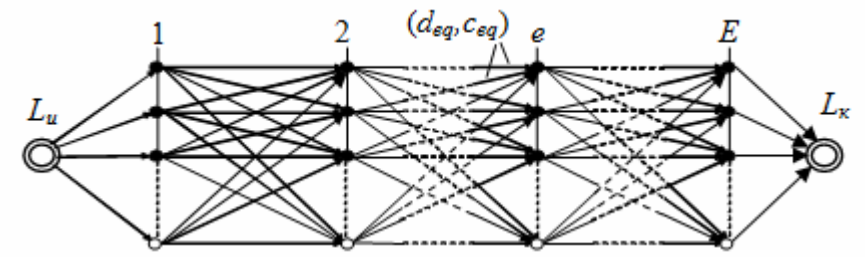
7.1. Графы многоверсионных технологий. Множество вариантов реализации принципа МВ привел к необходимости формализации понятия многоверсионной технологии и задачи ее выбора. МВТ фиксирует кортеж элементов – вариантов выбора видов версионной избыточности по Е этапам жизненного цикла и числа соответствующих версий.

Использование МВТ требует уточнения модели жизненного цикла. По аналогии со стандартными моделями разработки ПО и компьютерных систем, надежность и безопасность которых обеспечивается за счет многоверсионности, в ее основу следует положить *модель многоверсионного жизненного цикла* [19,28], базирующуюся на операциях генерации и выбора версий на различных этапах и при реализации различных процессов.

Вопросы постановки и решения задач описания, разработки и выбора многоверсионных технологий изложены в [6,19,25]. Задача выбора сформулирована и решается как оптимизационная задача поиска путей в биполярном E-уровневом графе с начальной L_u и конечной $L_к$ вершинами (рис.4) по критерию «многоверсионность (надежность, безопасность) – стоимость» [26]. Основные проблемы при этом – построение графа исходя из возможных видов ВИ и их совместимости, а также вычисление метрик диверсности d_{eq} ($e = 1,\ldots, E$; $q = 1,\ldots, Q_e$, Q_e – число вариантов выбора ВИ на этапе e) и стоимости c_{eq} по этапам ЖЦ.



а)



б)

Рис.4. Графы многоверсионных технологий
(вершины соответствуют выбранной паре (а) или одному из вариантов (б) ВИ)

Возможны два типа таких графов, когда вершины интерпретируются как вариант выбранного решения на определенном этапе жизненного цикла, характеризуемый видом и числом версий, а метриками взвешиваются вершины (рис.4,а), или когда каждая из вершин соответствует параметрам определенной версии, а метриками взвешиваются пары дуг (рис.4,б). В первом случае выбор МВТ сводится к выбору на графе одного из путей (нижний путь с незалитыми вершинами – одноверсионная технология с нулевыми метриками), во втором - пары путей с требуемыми характеристиками.

7.2. Показатели эффективности. Показатели надежности и безопасности должны рассчитываться с учетом метрик диверсности, получаемых на базе моделей дефектов. Модель дефектов *(n,m)*-версионной системы отличается от обычной МВС, в которой используется только один вид ВИ, тем, что множества дефектов от версий, полученных для разных видов версионной избыточности, могут соотноситься по-разному с учетом эффектов сжатия и компенсации [18,33].

Эффективность использования версионной избыточности H может характеризоваться приращением надежности (безопасности) системы при выборе s-той многоверсионной технологии, характеризуемой показателями $D(MBT_s)$ и C (MBT_s), на единицу затрат, обусловленных ее разработкой и применением по отношению к одноверсионной системе (технологии), характеризуемой показателями D (OBT) и $C(OBT)$:

$$H = [D\,(MBT_s) - D\,(OBT)] \,/\, [C\,(MBT_s) - C(OBT)]. \qquad (7.1)$$

Показатели D и C рассчитываются с учетом решений, принимаемых на каждом из этапов жизненного цикла системы.

8. Выводы. Направления дальнейших исследований

Многоверсионность – принцип, базирующийся на использовании различных продуктно-процессных средств для реализации идентичных функций с целью:

- создания компьютерных систем и инфраструктур, устойчивых к различным дефектам, и снижения риска отказа по общей причине;
- повышения полноты и достоверности верификации программного обеспечения и систем и снижения рисков невыявленных дефектов;

- улучшения информационной безопасности (целостности и конфиденциальности) и снижения рисков использования уязвимостей для несанкционированных вторжений.

В данной статье не ставилась задача детального анализа методов и проектных решений, основывающихся на использовании многоверсионности. В этой работе уточнена система терминов и понятий, связанных с многоверсионностью. Таксономическая схема, включающая понятие *(n,m)*-версионной системы, содержит следующую терминологическую цепочку:

- продукт (процесс) – *версия*,
- вид избыточности – *версионная избыточность*,
- принцип – *многоверсионность (диверсность)*,
- совокупность проектных действий – *многоверсионная технология*,
- результат разработки – *многоверсионная система*,
- совокупность процессов и продуктов – *многоверсионный проект*,
- модель жизненного цикла – *многоверсионный жизненный цикл*,
- показатель для оценки – *метрика многоверсионности (диверсности)*.

В качестве классификационной схемы видов ВИ предложен *куб многоверсионности*, который систематизирует их по этапам и процессам жизненного цикла, уровню введения избыточности (концептуальном, системном, компонентном) и конкретным проектным решениям. Эта трехмерная матрица может быть представлена набором двумерных матриц с фиксированными этапами ЖЦ или уровнем введения ВИ.

Модели *(n,m)*-версионных систем позволяют обобщить класс МВС с разными видами ВИ и оценить эффект от использования многоверсионности. Опыт разработки и практической реализации *(2,2)*-версионных информационно-управляющих систем на базе ПЛИС в АЭС показывает, что их внедрение позволило повысить уровень надежности и безопасности, степень соответствия требованиям нормативной базы.

В дальнейшем целесообразно исследовать различные варианты архитектур *(n,m)*-версионных систем, уточнив наполнение проектными решениями куб МВ для разных приложений. Ключевой проблемой остается оценка уровня гарантоспособности уникальных *(n,m)*-версионных систем в условиях недостаточной информации для вычисления метрик диверсности.

ЛИТЕРАТУРА

1. Ястребенецкий М.А., Васильченко В.Н., Виноградская С.В. и др. Безопасность атомных станций: Информационные и управляющие системы. / Под ред. М.А.Ястребенецкого. – К.: Техніка. – 2004. – 472 с.
2. Харченко В.С., Скляр В.В., Тарасюк О.М. Анализ рисков аварий для ракетно-космической техники: эволюция причин и тенденций // Радіоелектронні і комп’ютерні системи. – 2003. – № 3. – С.135-149.
3. Avizienis A., Laprie J.-C., Randell B., Landwehr C. Basic Concepts and Taxonomy of Dependable and Secure Computing // IEEE Trans. On Dependable and Secure Computing. – 2004. – Vol. 1. – №1. – P.11-33.
4. Харченко В.С. Модели и свойства отказоустойчивых многоальтернативных систем // Автоматика и телемеханика. – 1992. – № 12. – С.140-147.

5. Харченко В.С. Теоретические основы дефектоустойчивых цифровых систем с версионной избыточностью. – Х.: МОУ. – 1996. – 503 с.
6. Бахмач Е.С., Герасименко А.Д., Головир В.А. и др. Отказобезопасные информационно-управляющие системы на программируемой логике / Под ред Харченко В.С., Скляра В.В. – Нац. аэрокосм. ун-т «ХАИ», НПП «Радий». – 2008. – 380с.
7. Avizienis A., Lapri J.-C. Dependable Computing: From Concepts to Design Diversity // Proceedings IEEE. - 1986. – Vol. 74, n. 5. – P. 8–21.
8. Laprie J.-C. Dependability Handbook / LAAS Report n 98-346. – Toulouse: Laboratory for Dependability Engineering. – 1998. – 365 p.
9. Littlewood B., Popov P. Modelling the effects of combining diverse software fault removal techniques // IEEE Transactions on Software Engineering. – 2000. – SE-26(12). – P. 1157-1167.
10. Littlewood B., Strigini L. Littlewood B. Redundancy and diversity in security // Proc. 9th European Symposium on Research in Computer Security (ESORICS'2004), France. – 2004. – P. 117–126.
11. Popov P., Strigini L. Popov P. Conceptual models for the reliability of diverse systems – new results // Proc. 28th International Symposium on Fault-Tolerant Computing (FTCS-28). – Munich, Germany. – 1998. – P .80-89.
12. Popov P., Strigini L., Romanovsky A. Diversity for Off-The-Shelf Components // Proc. The International Conference on Dependable Systems and Networks – Goteborg, Sweden. – 2001. – P. 61-67.
13. Leveson N. Safeware: System Safety and Computers / Leveson N. – Addison-Wesley. – 1995. – 431 p.
14. Wood R. T. Diversity Approaches for Common Cause Failure Mitigation // IAEA Technical Meeting on Integrating Analog and Digital I&C Systems in Hybrid Main Control Rooms at Nuclear Power Plants. – Toronto, Canada, October 28–November 2, 2007.
15. Gorbenko A., Kharchenko V., Tarasyuk O, Furmanov A. F(I)MEA-Technique of Web-services Analysis and Dependability Ensuring. LNCS 4157. Rigorous Development of Complex Fault-Tolorant Systems/ Butler M., Jones C., Romanovsky A., Trubitsyna E. (eds.). Springer. – 2006. – P.153–168.
16. Харченко В.С. Гарантоздатність комп'ютерних систем: проблеми та результати //Авіаційно-космічна техніка і технологія. – 2005. – №7(23). – С.352-357.
17. Харченко В.С. Гарантоспособность и гарантоспособные системы: элементы методологии // Радіоелектронні і комп'ютерні системи.– 2006.– Вип. 5(17).– С.7–19.
18. Харченко В.С., Скляр В.В., Сиора А.А., Белый Ю.А. Модели безотказности и готовности встроенных мультидиверсных систем // Авиационно-космическая техника и технология. – 2008. – № 1(48). – С. 64-69.
19. Харченко В.С., Жихарев В.Я., Илюшко В.М. и др. Многоверсионные системы, технологии, проекты. / Под ред. В.С. Харченко. – Харьков: Нац. аэрокосм. ун-т «ХАИ». – 2002. - 486 с.
20. Харченко В.С., Пискачева И.В., Скляр В.В. Метрики диверсности: Классификация, анализ и применение для оценки надежности и безопасности компьютерных систем управления // Открытые

информационные и компьютерные интегрированные технологии.– Харьков: Нац. аэрокосмический ун-т «ХАИ».– 2001.– Вып. 9.– С. 194-214.
21. Скляр В.В. Анализ метрик многоверсионности программного обеспечения // Электронное моделирование. – 2004. – Т. 26. – № 4. – С. 95-104.
22. Kharchenko V., Yastrebenetsky M., Sklyar V. Diversity Assessment of Nuclear Power Plants Instrumentation and Control Systems // Proceeding by 7th International Conference on Probabilistic Safety Assessment and Management and European Safety and Reliability Conference.– Berlin (Germany). – 14–18 June 2004. – Volume 3. – P. 1351-1356.
23. 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.
24. Сиора А.А. Многоверсионый подход к реализации арифметических операций в системах обработки информации и управления критическими объектами на основе использования модулярной арифметики // Системи управління, навігації та зв'язку. – 2007. – Вип.4. – С. 146-153.
25. Kharchenko V.S., Sklyar V.V. (edits). FPGA-based NPP Instrumentation and Control Systems: Development and Safety Assessment**. –** Kharkiv, Kirovograd: Research and Production Corporation "Radiy", National Aerospace University named after N.E. Zhukovsky "KhAI", State Scientific Technical Center on Nuclear and Radiation Safety. – 2008. – 188 p.
26. Sklyar V., Kharchenko V. A Method of Multiversion Technologies Choice on Development of Fault-Tolerant Software Systems // Proceeding of Workshop on Methods, Models and Tools for Fault Tolerance. – Oxford, UK. – 2007. – P. 148-157.
27. Preckshot G. Method for Performing Diversity and Defense-in-Depth Analysis of Reactor Protection Systems. NUREG/CR-6303.– Livermore, USA: Lawrence Livermore National Laboratory, 1994.– 35 p.
28. Волковой А.В., Скляр В.В., Харченко В.С. Метод формирования моделей многоверсионного жизненного цикла для программных проектов // Інформаційно-керуючі системи на транспорті. – 2004. – №2. – С. 40-44.
29. Барсов В.В., Краснобаев В.А., Сиора А.А., Авдеев И.В. Методы многоверсионной обработки информации в модулярной арифметике. – Харьков: Министерство образования и науки Украины, УИПА. – 2008. – 460с.
30. Yakymets N. Resource-Oriented Diversification of Fault-Tolerant PLD-Systems / Yakymets N., Kharchenko V. // Радіоелектронні комп'ютерні системи. – 2006. – № 3. – С. 60-63.
31. Yakymets N., Kharchenko V. Design of Complex Fault-Tolerant PLD-Based Systems Using Genetic Algorithms // Proceedings of IEEE East-West Design and Test Symposium. – Yerevan, Armenia. – 2007. – P. 429 – 433.
32. Скляр В.В., Харченко В.С. Отказоустойчивые компьютерные системы управления с версионно-пороговой адаптацией: Способы адаптации, оценка надежности, выбор архитектур // Автоматика и телемеханика. – 2002. – № 6. – С 131-145.