УДК 004.4'242 DOI 10.26583/npe.2018.2.14

ВЕРИФИКАЦИЯ ПРОЦЕДУР ГЕНЕРАЦИИ И ЗАГРУЗКИ ПРИКЛАДНЫХ ПРОГРАММ СИСТЕМ БЕЗОПАСНОСТИ АТОМНЫХ ЭЛЕКТРОСТАНЦИЙ МЕТОДОМ ОБРАТНОГО ПРЕОБРАЗОВАНИЯ

М.А. Белоносов*, В.Л. Кишкин*,, С.А. Королев****

** ФГУП «ВНИИА им. Н.Л.Духова»*
127055, Москва, Сущевская ул.,22
*** НИЯУ МИФИ*
115409, г. Москва, Каширское шоссе, 31

Описан метод автоматической верификации прикладного программного обеспечения управляющих систем безопасности на базе аппаратуры ТПТС-СБ. Верификация выполняется путем сравнения двух математических моделей (ориентированных графов), одна из которых получена путем обработки исходных проектных данных – графических функциональных схем, а другая сформирована путем обратного преобразования программного кода, полученного из микроконтроллера. Вершинами в обоих графах являются функциональные блоки математических и логических операций, ребрами – связи между ними. Над построенными математическими моделями выполняется процедура сравнения – сопоставляются вершины и ребра графов, а также параметры вершин графов. Эквивалентность математических моделей является доказательством соответствия кода программы и исходной совокупности проектных функциональных схем.

Предложенный способ автоматической верификации позволяет доказать, что в процессе преобразования графических функциональных схем в код программы с последующей трансляцией и загрузкой кода в микроконтроллер в программу не внесено искажений. Постулируется, что любые искажения будут выявлены при выполнении процедуры верификации, которая выполняется штатно всякий раз после генерации и загрузки кода в микроконтроллер.

Решение обеспечивает приемлемую скорость при обработке больших объемов векторной графики, хранящейся в реляционной базе данных, и позволяет визуализировать результаты верификации. Предложенный способ реализован в инструментальных средствах GET-R1 для ТПТС-СБ и используется при разработке и верификации прикладного программного обеспечения систем безопасности Белорусской АЭС.

Ключевые слова: верификация, обратное преобразование, генерация кода, системы безопасности, микроконтроллер, математическая модель, инструментальные средства.

ВВЕДЕНИЕ

В ФГУП «ВНИИА им. Н.Л.Духова» завершена разработка программно-технических средств ТПТС-СБ для построения цифровых управляющих систем безопасности (УСБ) АСУТП АЭС. Этим системам отводится важнейшая задача обеспечения ядерной безопасности энергоблока при запроектных авариях, поэтому к ним предъявляются самые жесткие требования по разнообразию (диверсности), надежности и корректности программного обеспечения. Программно-технические средства ТПТС-СБ разработаны с учетом всех современных требований к таким системам.

Прикладные программы управляющих систем безопасности на базе ТПТС-СБ создаются в виде графических функциональных схем с помощью инструментальной среды GET-R1 [1] на проблемно ориентированном языке [2, 3]. Разработанные человеком графические алгоритмы управления проходят проверку на соответствие требованиям к проектированию прикладных программ для технических средств ТПТС-СБ; затем происходит автоматическая генерация кода программы на проблемно ориентированном языке и ее последующая трансляция в двоичное представление (байт-код). Транслированный байт-код загружается в микроконтроллер.

В статье описывается штатная процедура верификации прикладной программы после ее загрузки в микроконтроллер, включая считывание из микроконтроллера и обратное преобразование из байт-кода в графическое представление алгоритма.

Постулируется: в процессе преобразования графических алгоритмов в код программы, трансляции и загрузки кода в микроконтроллер не вносится скрытых искажений в программу; любые искажения будут выявлены с помощью предлагаемой процедуры верификации.

АРХИТЕКТУРА СИСТЕМЫ ТПТС-СБ И ПРИНЦИПЫ ПРОГРАММИРОВАНИЯ

ТПТС-СБ – программно-технические средства со встроенным программно-аппаратным разнообразием, предназначенные для построения цифровых систем безопасности АЭС. Разнообразие в системе достигнуто за счет её разделения на независимые диверситеты. Устройство и архитектура программно-технических комплексов на базе ТПТС-СБ подробно описаны в [4, 5].

Структура интегрированной УСБ на базе программно-технических средств ТПТС-СБ приведена на рис. 1.

Логическая обработка в обоих диверситетах выполняется программируемыми процессорными модулями автоматизации (ПМА). Эти микроконтроллерные модули в циклическом режиме считывают данные с модулей ввода, выполняют пользовательскую программу с помощью специального интерпретатора и выдают сигналы в модули вывода и модули приоритетного управления.

Для программирования ПМА используется язык прикладного уровня STEP-S, включающий в себя как логические и арифметические инструкции, так и сложные технологические, такие как интегрирование, ограничение сигнала, голосование и др. Программа на языке STEP-S представляет собой строго линейную, не содержащую циклов и переходов последовательность инструкций. Каждая инструкция содержит строго определенный набор аргументов и представляет собой оператор вида

CMD opd1 … opdN … value1 … valueN ,

где **CMD** – буквенно-цифровая последовательность, обозначающая инструкцию; **opd** (операнд, или маркер) – символьный адрес ячейки памяти ПМА; **value** – численная или символьная константа.

Для загрузки в ПМА программа на языке STEP-S транслируется в двоичное представление (байт-код) без изменения структуры программы.

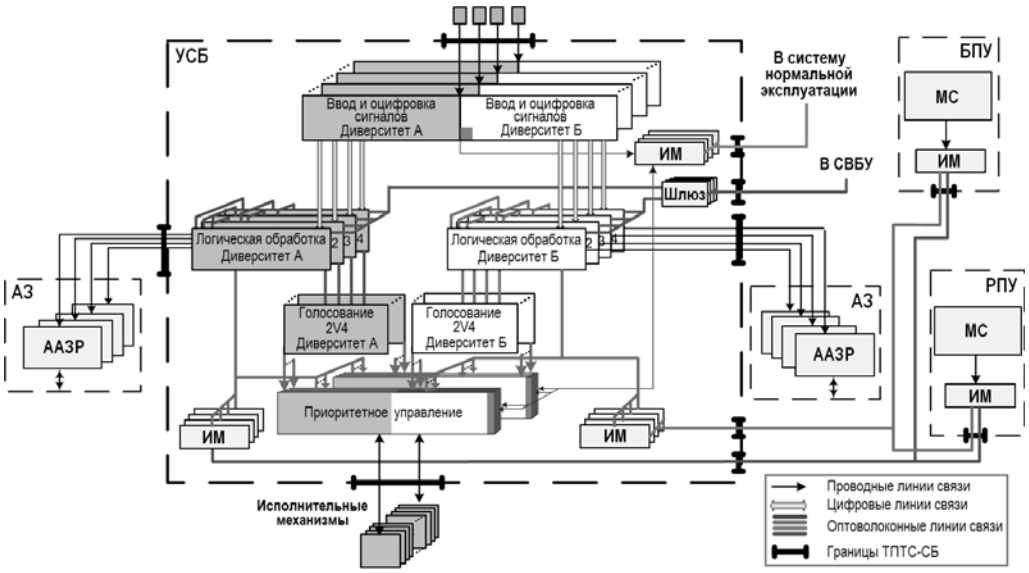

Рис. 1. Структура интегрированной УСБ на базе ТПТС-СБ: АЗ – аварийная защита; БПУ – блочный пульт управления; РПУ – резервный пульт управления; СВБУ – система верхнего блочного уровня; УСБ – управляющая система безопасности; ААЗР – автоматика аварийной защиты реактора; ИМ – интерфейсный модуль; МС – модуль связи

ИСХОДНЫЙ ПРОЦЕСС РАЗРАБОТКИ И ЗАГРУЗКИ ПРИКЛАДНОЙ ПРОГРАММЫ

Программы на языке STEP-S являются результатом обработки большого количества алгоритмов управления, разрабатываемых в графическом виде на языке функциональных схем [2, 3]. Эти диаграммы состоят из функциональных блоков, обозначающих арифметические, логические или сложные технологические операции. Входы и выходы функциональных блоков соединены между собой; таким образом, функциональная схема представляет собой графически изображенную последовательность вычислений. Каждому функциональному блоку заранее сопоставлена шаблонная последовательность инструкций STEP-S, содержащая не менее одной инструкции.

В качестве примера на рис. 2 представлена функциональная схема обработки сигналов от датчиков сейсмической обстановки путем голосования с учетом достоверности сигналов и выдачи сигнала аварийной защиты. В таблице 1 приведен фрагмент результирующей программы – сгенерированная последовательность инструкций на языке STEP-S, соответствующая функциональной схеме.

Каждый функциональный блок, изображенный на схеме, преобразован в последовательность из одной или более инструкций. Генерация кода производится в четыре этапа.

1. Производится расчет и назначение ячеек памяти ПМА (маркеров), необходимых для произведения вычислений и сохранения результатов.

2. Определяется порядок вычислений. Место, куда в итоговой программе попадают инструкции, соответствующие определенному функциональному блоку, определяется порядковым номером этого блока, вычисляемым автоматически с помощью топологической сортировки ациклического ориентированного графа – математической модели, соответствующей функциональным схемам ПМА [6]. Номера функциональных блоков на рис. 3 и номера инструкций STEP-S в табл. 1 соответствуют друг другу.

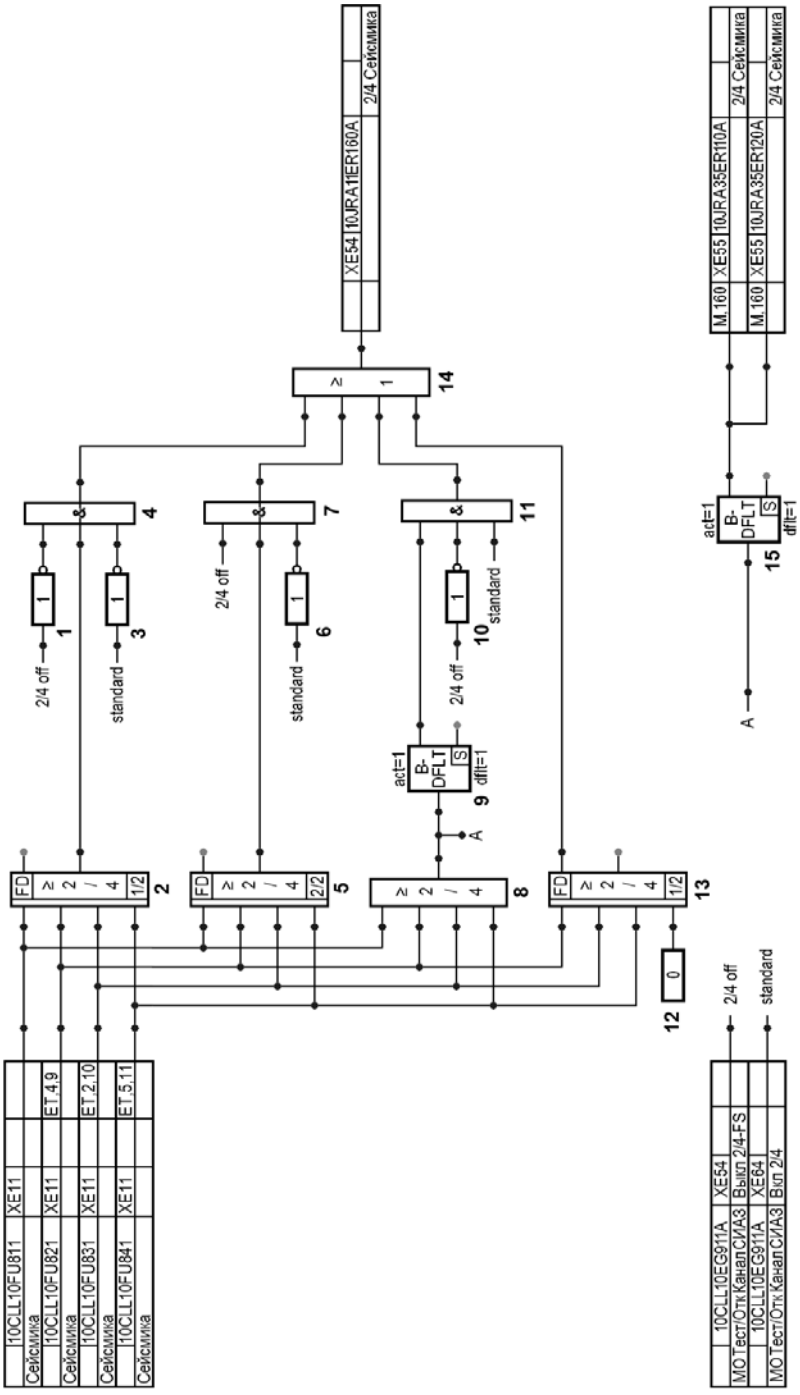

Рис. 2. Функциональная схема обработки сигналов от датчиков сейсмической обстановки

Таблица 1

Фрагмент прикладной программы на языке STEP-S

№	Инструкция STEP-S
1	NOT M,133 M,141
2	2/4-FS ET,4,9 ET,2,10 ET,5,11 M,140 M,145 M,146
3	NOT M,133 M,142
4	AND-3 M,150 M,141 M,135 M,152
5	2/4-FS2 M,138 ET,4,9 ET,2,10 ET,5,11 M,153 M,154
6	NOT M,135 M,143
7	AND-3 M,133 M,153 M,143 M,155
8	2/4 M,138 ET,4,9 ET,2,10 ET,5,11 M,147
9	B-DFLT M,147 M,148 M,149 1 1
10	NOT M,135 M,144
11	AND-3 M,142 M,156 M,144 M,158
12	B-LADK M,143 0
13	2/4-FS M,138 ET,4,9 ET,2,10 ET,5,11 M,156 M,157
14	OR-4 M,158 M,155 M,152 M,146 M,159
15	B-DFLT M,147 M,150 M,151 1 1

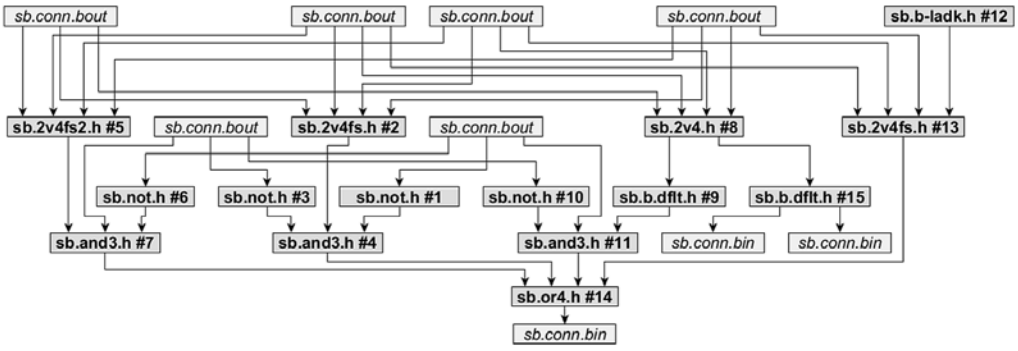

Рис. 3. Математическая модель-граф, соответствующая функциональной схеме на рис. 2. Вершины *sb.conn.bout* соответствуют входным сигналам функциональной схемы, *sb.conn.bin* – выходным. Остальные вершины соответствуют функциональным блокам и имеют порядковый номер

3. Выполняются автоматические проверки целостности и корректности проектных данных, а также соблюдения формальных правил проектирования для платформы ТПТС-СБ. В случае наличия ошибок генерация прекращается.

4. Выполняется обработка последовательности функциональных блоков в порядке, определенном на этапе 2. Производится подстановка шаблонных инструкций, соответствующих каждому функциональному блоку, в результирующую программу с проставлением рассчитанных ранее ячеек памяти и значений.

Все этапы генерации выполняются автоматически. Результирующая программа проходит этап трансляции, затем полученный байт-код загружается в ПМА ТПТС-СБ.

ПРОЦЕСС ОБРАТНОГО ПРЕОБРАЗОВАНИЯ ПРОГРАММЫ В ГРАФИЧЕСКОЕ ИЛИ ТАБЛИЧНОЕ ПРЕДСТАВЛЕНИЕ

При проектировании управляющих систем безопасности АЭС необходимо доказать корректность всех этапов генерации, трансляции и загрузки кода, а также отсутствие искажений в результирующей программе по сравнению с графическим алгоритмом. Сделать это напрямую крайне сложно по следующим причинам:

– строгое доказательство корректности всех программ, работающих на разных этапах генерации кода, является трудоемким процессом;

– нет гарантии, что в процессе генерации не произойдет сбоя, который повлияет на целостность данных. Такой сбой может быть вызван и внешними факторами, например, ошибками передачи данных, и внутренними факторами, такими как скрытые ошибки программы, ошибки чтения (записи) данных и т.п.

Единственно приемлемым способом доказательства корректности программы является разработка процедуры обратного преобразования сгенерированной и загруженной программы в графическое или табличное представление и сравнение восстановленных данных с исходным проектом. При этом графическое представление – это общая функциональная схема, включающая в себя все функциональные блоки, из которых сгенерирована программа ПМА, и связи между ними; табличное представление – это список всех функциональных блоков и их параметров в табличном виде. Однако и такое доказательство связано с рядом трудностей. В сгенерированной на языке STEP-S программе отсутствуют следующие данные:

– информация о принадлежности инструкций STEP-S конкретному алгоритму управления;

– графическая информация, необходимая для разбиения графики на схемы и определения координат функциональных блоков;

– обозначения алгоритмов, надписи и расшифровки, необходимые для понимания алгоритма человеком;

– информация о графическом исполнении функциональных блоков (одному и тому же функциональному блоку могут соответствовать несколько пиктограмм).

Поэтому задача полного восстановления исходного графического представления алгоритмов управления по коду программы является не решаемой без привлечения проектной информации.

Однако для решения задачи доказательства корректности процедур генерации и загрузки кода не требуется полного восстановления графики. Более того, такая процедура при больших объемах проекта будет занимать неоправданно много времени. Вместо полного восстановления проекта предлагается построить математические модели проекта в виде ориентированных графов по разным исходным данным: один граф строится по проекту, другой – по коду программы. Вершинами графов являются функциональные блоки, ребрами – связи между ними. Над построенными математическими моделями выполняется процедура сравнения – сравниваются вершины и ребра графов, а также ячейки памяти и значения констант. Эквивалентность математических моделей является доказательством соответствия кода программы и исходного набора проектных функциональных схем. Математическая модель для приведенной выше функциональной схемы показана на рис. 3.

Процедура обратного восстановления реализована независимо от программ загрузки, трансляции и генерации кода. Восстановление выполняется в шесть этапов.

1. Выполняется считывание байт-кода из ПМА и обратная трансляция в строковое представление на языке STEP-S.

2. Строится список последовательностей инструкций STEP-S для функциональных

блоков из библиотеки и сортировка этого списка по количеству инструкций в каждом блоке.

3. Выполняется поиск шаблонных последовательностей функциональных блоков в коде программы и разбиение программы на соответствующие фрагменты. Каждый фрагмент инструкций STEP-S заменяется соответствующим функциональным блоком.

4. Выполняется подстановка ячеек памяти (маркеров) и констант из кода во входы и выходы функциональных блоков.

5. Определяются связи между функциональными блоками по соответствию ячеек памяти (если один и тот же маркер присвоен входу одного блока и выходу другого, то эти вход и выход связываются). В конце этого этапа уже имеется полноценный ориентированный граф, построенный по коду программы.

6. Выполняется построение [7] такого же ориентированного графа по проекту (базе данных). При этом последовательность функциональных блоков и связей между ними уже хранится в базе данных. Остальная проектная информация игнорируется.

Результат сравнения двух ориентированных графов может быть визуализирован как в табличном виде, так и в виде общей функциональной схемы, включающей в себя все алгоритмы ПМА. При визуализации результатов сравнения в виде общей функциональной схемы считается, что различные графические представления одной и той же функции, например, вертикальное и горизонтальное изображение функции AND2 («И на два входа»), инвариантны между собой. Поэтому для сравнения используется тот графический вариант блоков, который наиболее подходит с визуальной точки зрения. Функциональные схемы, сгенерированные по коду и по проекту одних и тех же версий, будут идентичны.

ПРЕИМУЩЕСТВА В СРАВНЕНИИ С ИЗВЕСТНЫМИ РЕШЕНИЯМИ

Известна процедура верификации генератора кода инструментальных средств SPACE [8] для программно-технических средств TELEPERM XS (AREVA). Программа, выполняющая верификацию, называется Retrans и разработана сотрудниками Institut fur Sicherheitstechnologie (ISTec). Retrans является независимой процедурой обратного преобразования из кода на языке С в программное представление проекта с последующим сравнением. Retrans использует при восстановлении как сам код программы, так и дополнительную проектную информацию, содержащуюся в генерируемом исходном коде в виде комментариев.

Решение, примененное для TELEPERM XS, не верифицирует этапы трансляции и загрузки кода, на которых могут произойти сбои. Верификация сгенерированного кода на языке С доказывает только корректность генерации, но не может служить доказательством идентичности проекта и скомпилированной программы, загруженной в микроконтроллер.

Предлагаемое решение для аппаратуры ТПТС-СБ является более полным, так как верифицируется вся цепочка обработки, включая генерацию, трансляцию и загрузку. При этом математическая модель проекта восстанавливается абсолютно независимо без привлечения какой-либо проектной информации. Кроме этого, использование графовых моделей позволяет наглядно визуализировать результаты сравнения, применив алгоритм иерархической укладки графов [9, 10]. Вероятность того, что одновременно в генераторе кода и в процедуре обратного преобразования возникнут две независимых ошибки, одна из которых скроет другую, пренебрежительно мала. Поэтому такое решение можно считать строгим доказательством корректности выполнения автоматического преобразования алгоритмов управления в объектный код прикладной программы при условии выполнения процедуры обратного преобразования каждый раз после генерации и загрузки кода.

ЗАКЛЮЧЕНИЕ

Предложен метод полной верификации микроконтроллерных программ через обратное преобразование. Метод может быть применен для верификации прикладных программ любых микроконтроллеров интерпретирующего типа, программируемых с помощью графических языков стандарта МЭК 61131-3.

В инструментальные средства GET-R1 для ТПТС входит программный компонент, реализующий этот метод. В рамках этих инструментальных средств предложенный способ верификации применяется при разработке прикладного программного обеспечения управляющих систем безопасности Белорусской АЭС-2 на базе ТПТС-СБ.

Для обеспечения полной гарантии корректности кода в аппаратуре ТПТС-СБ предусмотрена возможность загрузки кода с отложенным запуском, что позволяет вначале загрузить программу, затем выполнить процедуру обратного преобразования и только после этого запустить новую программу.

Основные проектные решения по АСУТП современных АЭС российского дизайна представлены в [11 –13, 15, 16, 20], методы верификации программно-технических комплексов АСУТП АЭС на базе ТПТС – в [14, 21], современные международные требования к ПТК АСУТП АЭС в части обеспечения безопасности – в документах МАГАТЭ и МЭК [17 – 19].

Литература

1. *Белоносов М.А. и др.* Инструментальные средства сквозного проектирования систем контроля и управления на базе ТПТС // Доклады БГУИР. – 2015. – Т. 2, – № 88. – С. 47-51.

2. International Electrotechnical Comission. Standard IEC61131. – 2003. –Vol. 3. – 226 p.

3. *Зюбин В.Е.* Программирование ПЛК: языки МЭК 61131-3 и возможные альтернативы // Промышленные АСУ и контроллеры. – 2005. – Т. 11. – С. 31-35.

4. *Тимохин Д.С. и др.* Структура автоматизированной системы управления технологическими процессами Белорусской АЭС с точки зрения безопасности // Доклады БГУИР. – 2015. – Т. 2. – № 88. – С. 28-32.

5. *Нариц А.Д. и др.* Комплекс средств автоматизации ТПТС-СБ // Доклады БГУИР. – 2015. – Т. 2. – № 88. – С. 38-42.

6. *Tarjan R.* Depth-first search and linear graph algorithms / XIIth Annu. Symp. Switch. Autom. Theory (swat 1971). – 1971. –Vol. 1. – No. 2. –pp. 146-160.

7. *Филатова Н.Н.* Структурный синтез схем автоматизации в условиях неполных требований к технической реализации // Известия ВолГТУ. – 2012. – Т. 4. – № 13. – С. 17-22.

8. *Miedl H.* Retrans – a tool to verify the functional equivalence of automatically generated source code with its specification. / Probabilistic Safety Assessment and Management (PSAM-III). – Crete, Greece, 1996. – pp. 137-147.

9. *Бабурин Д. Е.* Иерархический подход для автоматического размещения ациклических графов. Электронный ресурс: http://www.iis.nsk.su/files/articles/sbor_kas_09_baburin.pdf (дата доступа 02.02.2018).

10. *Spqnetmann M., von Hanxleden R., Fuhrmann D.I.H.* On the automatic layout of data flow diagrams. Arbeit. 2009.

11. *Зверков В.В.* Программно-технические комплексы управляющих систем безопасности АЭС // Электрические станции. – 2017. – № 1. – С. 2-10.

12. *Боженков О.Л.* Системная инженерия АСУТП АЭС // Ядерные измерительно-информационные технологии. – 2009. – № 2. – С. 27-30.

13. *Зверков В.В.* Анализ подходов к построению АСУТП АЭС // Электрические станции. – 2015. – №8. – С. 2-6.

14. *Korolev S., Tolokonsky A., Rogov V.* The optimal approach for the processes of verification and validation of NPP software and hardware complexes // Journal of Physics: Conference Series. – 2017. – Vol. 781. – No. 1. – doi.org/10.1088/1742-6596/781/1/012048

15. *Дунаев В.Г., Королев С.А.* АСУТП энергоблоков АЭС с ВВЭР. В кн. Ядерная Энергетика.

Проблемы. Решения. Ч. 1. / Под ред. М.Н. Стриханова. – М.: ЦСПиМ, 2011. – С. 315-356.

16. *Зверков В.В.* Автоматизированная система управления технологическими процессами АЭС. – М.: МИФИ, 2014. – 558 с.

17. МАГАТЭ, № SSR-2/1. Безопасность атомных электростанций: проектирование. Конкретные требования безопасности. Вена. 2012.

18. МЭК 61513-2002. Атомные электростанции. Системы контроля и управления, важные для безопасности. Общие требования.

19. МАГАТЭ NS-G-1.1. Программное обеспечение управляющих систем, важных для безопасности, выполненных на основе компьютерной техники. Руководство по безопасности. Вена. 2000

20. Системы управления и защиты ядерных реакторов. Серия: Безопасность атомных станций / Под ред. М.А. Ястребенецкого. – Киев. Основа-Принт. 2011. – 770 с.

21. *Королев С.А., Толоконский А.О., Рогов В.В.* Современные методы верификации программно-технических комплексов АСУТП АЭС на базе ТПТС // Электрические станции. – 2016. – № 8. – С. 9-15.

Поступила в редакцию 19.10.2017 г.

Авторы

Белоносов Михаил Александрович, зам. начальника научно-исследовательского отдела
E-mail: belonosoff@gmail.com

Кишкин Владимир Львович, первый зам. главного конструктора, зав. кафедрой
E-mail: npk1@vniia.ru

Королев Сергей Андреевич, зам. зав. кафедрой, доцент, канд. техн. наук
E-mail: litos_mephi@mail.ru, SAKorolev@mephi.ru

UDC 004.4'242

VERIFICATION OF APPLIED PROGRAMS GENERATION AND LOADING FOR SAFETY SYSTEMS OF NUCLEAR POWER PLANTS BASED ON A REVERSE-ENGINEERING METHOD

Belonosov M.A.*, Kishkin V.L.*,**, Korolev S.A.**

* FSUE «VNIIA n.a. N.L. Dukhov»
22 Sushchevskaya str., Moscow, 127055 Russia,
** NRNU MEPHI
31 Kashirskoe shosse, Moscow, 115409 Russia

ABSTRACT

The article describes the automatic verification method used for application software of digital safety systems based on the TPTS-SB equipment. Verification is performed by comparing two mathematical models (oriented graphs): one obtained by processing the original design data, i.e., graphical functional diagrams, and the other formed by reversing the applied program code downloaded from the microcontroller. The vertices in both graphs are the functional blocks of mathematical and logical operations, while the edges are the connections between them. Over the constructed mathematical models, a comparison procedure is performed: the vertices and edges of the graphs are compared as well as the parameters of the graph vertices.

The equivalence of mathematical models is the proof of the correspondence between the program code and the initial set of design functional diagrams.

The proposed automatic verification method makes it possible to prove that no distortion is introduced into the program during the process of converting graphical functional diagrams to the program code with its subsequent translation and loading into the microcontroller. It is postulated that any distortions will be detected during the verification procedure, which is regularly performed every time after loading the code into the microcontroller.

The solution also provides an acceptable speed when processing large volumes of vector graphics stored in a relational database, and makes it possible to visualize the verification results. The proposed method is implemented in the GET-R1 instrumentation tools for TPTS-SB and is used in designing and verifying the application software of the security systems at the Belarusian NPP.

Key words: verification, reverse engineering, code generation, safety systems, microcontroller, mathematical model, instrumentation tools.

REFERENCES

1. Belonosov M.A., Galitsyn Y.S., Krajushkin U.V., Zhukov I.M., Gritsenko S.Y. The end-to-end engineering tools for instrumentation and control systems for nuclear power plants'. *Reports of BSUIR*. 2015, v. 2, no. 88, pp. 47-51 (in Russian).

2. International Electrotechnical Comission, Standard IEC61131. 2003, v. 3, 226 p.

3. Zyubin V.E. PLC Programming: IEC 61131-3 languages and possible alternatives. *Promyshlennye ASU i kontrollery*. 2005, v. 11, pp. 31-35 (in Russian).

4. Timohin D.S., Gricenko S.Yu.,. Artem'ev K.P. The structure of the automated process control system of the Belarusian NPP in terms of safety'. *Reports of BSUIR*. 2015, v. 2, no. 88, pp. 28-32 (in Russian).

5. Naritz A. D., Moiseev M. I., Novikov A. N., Karpov P. S., Borzenko A. A. The complex of automation system TPTS-SB. *Reports of BSUIR*. 2015, v. 2, no. 88, pp. 38-42 (in Russian).

6. Tarjan R. Depth-first search and linear graph algorithms. XII-th Annu. Symp. Switch. Autom. Theory (swat 1971). 1971, v. 1, no. 2, pp. 146-160 (in Russian).

7. Filatova N.N. Structural synthesis of automation schemes in conditions of incomplete requirements for technical implementation. *Izvestiya VolGTU*. 2012, v. 4, no. 13, pp. 17-22 (in Russian).

8. Miedl H. Retrans – a tool to verify the functional equivalence of automatically generated source code with its specification. Probabilistic Safety Assessment and Management (PSAM-III). Crete, Greece, 1996, pp. 137-147.

9. Baburin D.E. Hierarchical approach for automatic allocation of acyclic graphs. Available at: http://www.iis.nsk.su/files/articles/sbor_kas_09_baburin.pdf (accessed Feb 2, 2018) (in Russian).

10. Sponemann M., Hanxleden R., H. Fuhrmann D. I. On the automatic layout of data flow diagrams. Arbeit. 2009.

11. Zverkov V.V. Program-Technical Complexes of Control Systems for Safety of Nuclear Power Plants'. *Elektricheskie stantsii*. 2017, no. 1, pp. 2-10 (in Russian).

12. Bozhenkov O.L. System engineering of the automated process control system of NPPs. *Yadernye izmeritel'no-informacionnye tehnologii*. 2009, no. 2, pp. 27-30 (in Russian).

13. Zverkov V.V. Analysis of approaches to the construction of automated process control systems of NPPs. *Electricheskie Stantsii*. 2015, no. 8, pp. 2-6 (in Russian).

14. Korolev S., Tolokonsky A., Rogov V. The optimal approach for the processes of verification and validation of NPP software and hardware complexes. *Journal of Physics: Conference Series*. 2017, v. 781, no. 1, pp. 82-89.

15. Dounaev V.G., Korolev S.A. PCS of power units of nuclear power plants with VVER. In:

Nuclear Power. Problems. Solutions. Part 1. Ed. M.N. Strikhanov. Moscow. TsSPiM Publ., 2011, pp. 315-356 (in Russian).

16. Zverkov V.V. *Automated control system for technological processes of NPPs*. Moscow. MEPhI Publ., 2014, 558 p. (in Russian).

17. IAEA, SSR-2/1. Safety of nuclear power plants: design. Specific safety requirements. Vienna. IAEA. 2012.

18. IEC 61513-2002. Nuclear power plants. Monitoring and control systems important for safety. General requirements. 2002.

19. IAEA, NS-G-1.1. The software of control systems, important for safety, executed on the basis of computer equipment. Safety Guide. Vienna. IAEA. 2000.

20. Control systems and protection of nuclear reactors. Ser. Safety of Nuclear Power Plants. Ed. M.A. Yastrebenetsky. Kiev. Osnova-Print Publ., 2011, 770 p. (in Russian).

21. Korolev S.A., Tolokonsky A.O., Rogov V.V. Modern methods of verification of software and hardware complexes of automated process control systems of nuclear power plants based on TPTS. *Elektricheskie stantsii*. 2016, no. 8. pp. 9-15 (in Russian).

Authors

Belonosov Mikhail Aleksandrovich, Deputy Chief of Research Department
E-mail: belonosoff@gmail.com

Kishkin Vladimir L'vovich, First Deputy Chief Designer, Department Chair
E-mail: npk1@vniia.ru

Korolev Sergey Andreevich, Deputy Department Chair, Associate Professor,
Cand. Sci. (Engineering)
E-mail: litos_mephi@mail.ru, SAKorolev@mephi.ru