

Т. В. Лях¹, В. Е. Зюбин¹, Н. О. Гаранина²

¹ *Институт автоматики и электрометрии СО РАН
пр. Академика Коптюга, 1, Новосибирск, 630090, Россия*

² *Институт систем информатики им. А. П. Ершова СО РАН
пр. Академика Коптюга, 6, Новосибирск, 630090, Россия*

antsys_nsu@mail.ru, zyubin@iae.nsk.su, garanina@iis.nsk.su

АВТОМАТИЗИРОВАННАЯ ВЕРИФИКАЦИЯ АЛГОРИТМОВ УПРАВЛЕНИЯ СЛОЖНЫМИ ТЕХНОЛОГИЧЕСКИМИ ОБЪЕКТАМИ НА ПРОГРАММНЫХ ИМИТАТОРАХ *

Статья посвящена проблеме проверки алгоритмов управления, созданных в рамках процесс-ориентированного подхода, на соответствие входным спецификациям. Представлена общая схема верификации алгоритма управления, приведены ее реализация в автоматизированном варианте и результаты практической апробации в проекте по автоматизации Большого солнечного вакуумного телескопа.

Ключевые слова: автоматизация, промышленные алгоритмы управления, верификация, процесс-ориентированные языки программирования.

Введение

Алгоритмы управления сложными технологическими объектами обладают рядом свойств, специфичных для области промышленной автоматизации [1–4]:

- открытость – наличие «окружающей среды», внешнего мира, с которым взаимодействует алгоритм управления;
- событийность – алгоритм управления формирует управляющие воздействия как реакцию на события (значимые изменения во входных данных), в том числе на управляющие команды от оператора;
- неопределенная продолжительность функционирования алгоритма управления;
- синхронизм – необходимость синхронизации реакции алгоритма управления с событиями на объекте управления;
- логический параллелизм – алгоритм управления структурно отражает параллелизм физических процессов на объекте управления, их независимость.

Реализация алгоритмов управления средствами объектно-ориентированных языков общего назначения чревата чрезмерным усложнением программной архитектуры при росте сложности алгоритма [5]. Поэтому в области промышленной автоматизации используются специализированные языковые средства для разработки алгоритмов управления: языки МЭК 61131-3, G (NI LabVIEW), Reflex [6–8].

* Работа выполнена при финансовой поддержке Федерального агентства научных организаций (государственная регистрация № АААА-А17-117060610006-6) и при финансовой поддержке РФФИ (проект № 17-07-01600).

Использование языков МЭК 61131-3 трудоемко из-за низкой выразительности этих языков, а в некоторых случаях и неприемлемо, например при необходимости интеграции кода в сторонние системы [6].

Исследователи альтернативных лингвистических средств для описания алгоритмов управления [2; 4; 7–10] предлагают и практически обосновывают эффективность предметно-ориентированных языков на основе модели конечного автомата, в частности процесс-ориентированного языка Reflex [10].

При использовании конечно-автоматных языков в промышленной автоматизации [7; 8; 11; 12] основную проблему представляет решение задачи тестирования и верификации созданных алгоритмов, поскольку методы, разработанные для тестирования и верификации программного обеспечения в области объектно-ориентированного программирования, слабо применимы [5]. Управляющий алгоритм невозможно тестировать автономно. Тестирование алгоритма на реальном объекте управления может привести к поломке оборудования или аварийной ситуации. Поэтому наиболее распространенный подход – ручная проверка на этапе пуско-наладки: проверяющий контролирует реакцию алгоритма на различные ситуации, постепенно усложняя тесты. Подход очень трудоемкий. Он приводит к серьезным психологическим нагрузкам на разработчиков, не гарантирует полноту верификации, затрудняет контроль качества верификации и в итоге усложняет разработку управляющих алгоритмов [11].

Поэтому разработка методов верификации алгоритмов управления интересна не только с теоретической, но и с практической стороны [5; 13–18]. Современная тенденция – использовать для тестирования и верификации алгоритмов управления программные имитаторы объекта управления [11; 19–21].

В статье предлагается подход к верификации алгоритмов управления сложными технологическими объектами на основе концепции виртуальных объектов управления (ВОУ), включающий в себя создание кода алгоритма управления и программную реализацию объекта управления в виде ВОУ.

Основные положения, изложенные в статье, были представлены на V Международной научной конференции «Математическое и компьютерное моделирование» (Омск, 2017) [22].

Общая схема верификации алгоритмов управления технологическими объектами

Общая схема итерационной разработки алгоритмов управления (рис. 1), предложенная ранее в [11], включает следующие шаги:

- верифицируемый алгоритм управления (его часть) реализуется программно и оформляется в обособленный алгоблок;
- модель технологического объекта (его часть) также реализуется программно и оформляется в обособленный алгоблок, называемый виртуальным объектом управления;
- проводится верификация через создание тестовых ситуаций (сценариев) и контроль реакции алгоритма управления;
- коррекция кода алгоритма управления и (или) виртуального объекта управления по результатам верификации.

Схема позволяет использовать итерационный подход к разработке промышленных алгоритмов управления сложными технологическими объектами. В упрощенном виде схема была опробована при создании набора виртуальных лабораторных стендов для обучения студентов ИТ-специальностей, специализирующихся в области промышленной автоматизации [23]. Упрощение заключалось в неизменности ВОУ. При этом ВОУ выполнены с использованием графики для повышения наглядности и визуального контроля корректности алгоритма управления.

Несмотря на высокую эффективность при использовании в учебном процессе, подход не нашел практического применения в реальных проектах в силу высокой трудоемкости создания графических моделей ВОУ.

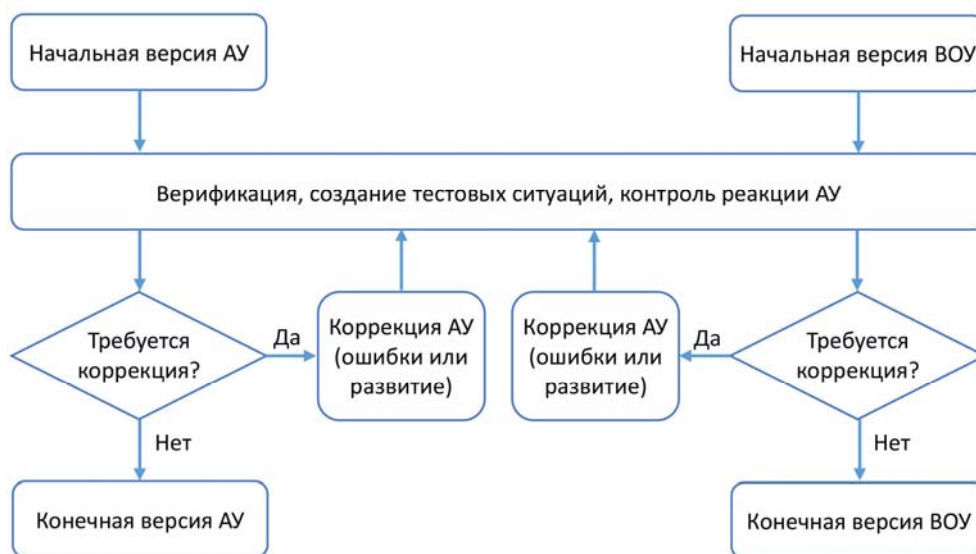


Рис. 1. Общая схема итерационной разработки алгоритмов управления:
 АУ – алгоритм управления, ВОУ – виртуальный объект управления

Метод разработки алгоритмов управления на основе концепции виртуальных объектов управления

Для использования в реальных проектах автоматизации предложенная схема была программно реализована в автоматизированном комплексе верификации алгоритмов управления.

В автоматизированном варианте (рис. 2, таблица) управление сценариями работы и контроль реакции алгоритма производится оператором через графический интерфейс (1). Графический интерфейс предоставляет оператору возможность отправлять штатные команды алгоритму управления (6) через очередь сообщений (2) и контролировать сообщения от алгоритма управления через очередь сообщений (4), управлять поведением ВОУ (9) через очередь сообщений (3) и контролировать сообщения от ВОУ через очередь сообщений (5). Также на графическом интерфейсе оператора (ГИО) отображаются состояния входных (7) и выходных (8) дискретных сигналов алгоритма управления. Для имитации входных аналоговых сигналов от датчиков / АЦП организован дополнительный канал связи между ВОУ и АУ (10), а для имитации выходных аналоговых сигналов (ЦАП) – канал связи (11).

Программный комплекс автоматизированной верификации алгоритмов управления был реализован на базе пакета LabVIEW (рис. 3). Комплекс включает ГИО (2), который конфигурируется через модуль загрузки конфигурационных файлов (МЗКФ) (1) и взаимодействует с целевыми модулями алгоблоков АУ (3) и ВОУ (4). Модули загрузки конфигурационных файлов и ГИО реализованы на языке G LabVIEW. Исполняемые модули АУ и ВОУ генерируются из описания на языке Reflex и в виде DLL интегрируются в LabVIEW, что в отличие от подхода, использованного при разработке виртуальных лабораторных стендов, обеспечивает возможность итерационно развивать не только АУ, но и ВОУ.

Управление верификацией ведется оператором через ГИО (рис. 4). ГИО представлен расположенной внизу окна неизменяемой панелью управления верификацией (3) и шестью вкладками: генерации входных сообщений для АУ и ВОУ, переменных АУ (рис. 5), переменных ВОУ, отладочной информации АУ (рис. 6), отладочной информации ВОУ и вкладки «Помощь» с руководством по работе с комплексом. Панель управления верификацией содержит элементы управления режимом отладки (пошаговый / непрерывный), запуска и остановки алгоблоков.

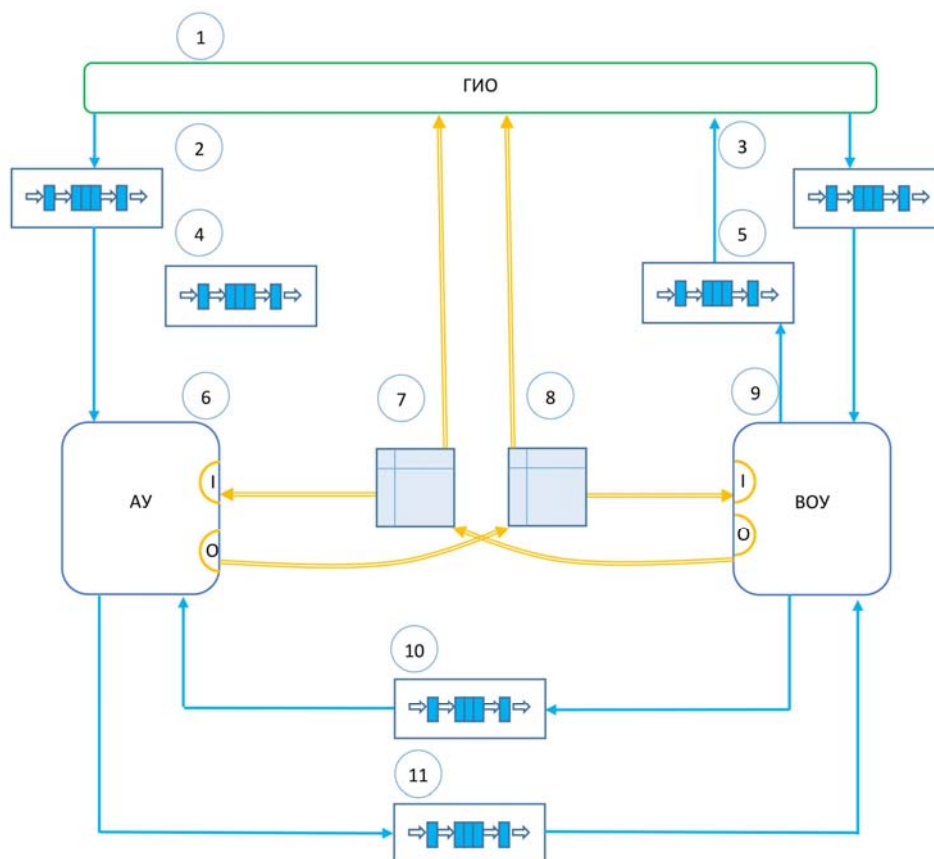








Рис. 2. Архитектура автоматизированного комплекса верификации: 1 – ГИО; 2 – очередь сообщений от ГИО; 3 – очередь сообщений для ВОУ; 4 – очередь сообщений от АУ; 5 – очередь сообщений от ВОУ; 6 – АУ; 7 – буфер входных дискретных сигналов для АУ; 8 – буфер выходных дискретных сигналов для АУ; 9 – ВОУ; 10 – очередь сообщений для имитации аналоговых данных от датчиков / АЦП для АУ; 11 – очередь сообщений для имитации выходных аналоговых сигналов (ЦАП) от АУ

Используемые условные обозначения

	Алгоблок
	Цифровые порты алгоритмического блока (I – входные, O – выходные)
	Буфер значений дискретных цифровых сигналов
	Очередь сообщений
	Цифровые данные
	Сообщения

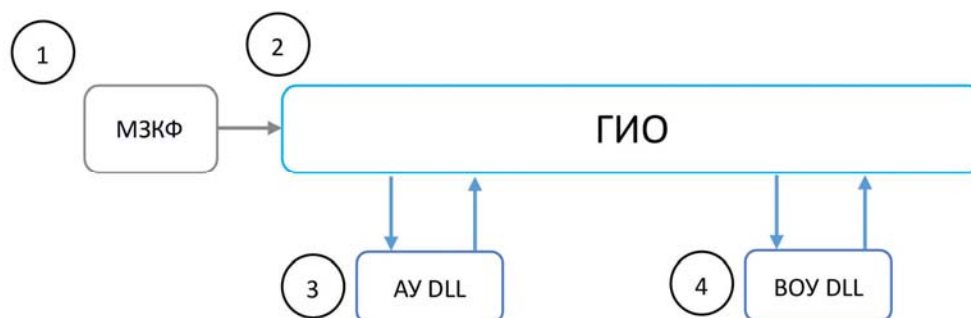


Рис. 3. Архитектура программного комплекса автоматизированной верификации алгоритмов управления

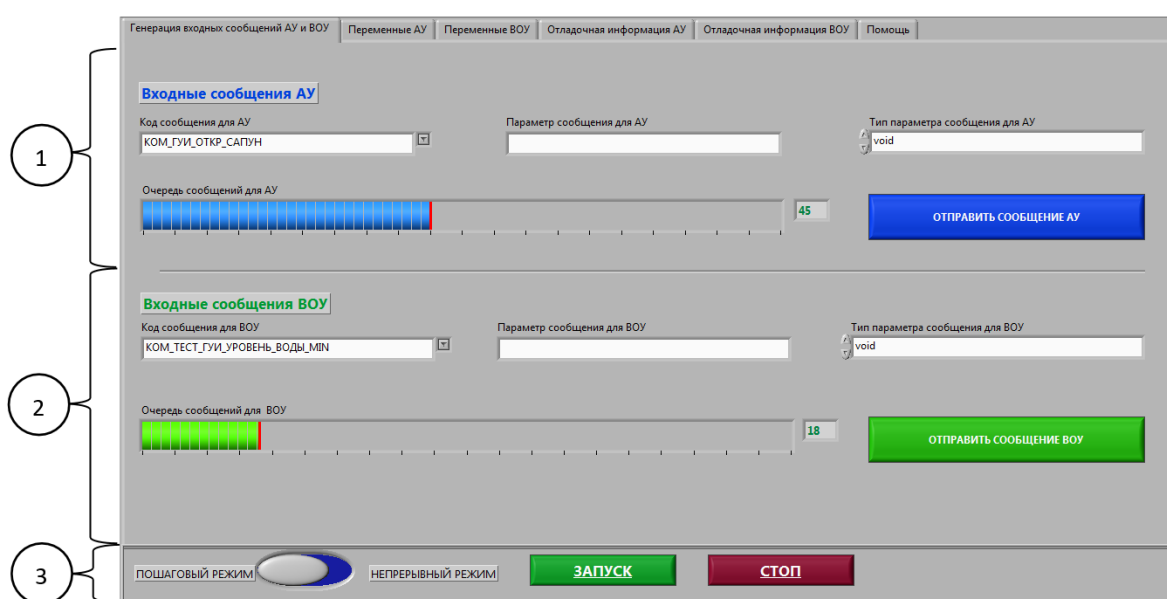


Рис. 4. Графический интерфейс оператора, вкладка «Генерация входных сообщений для АУ и БОУ»: 1 – панель генерации входных сообщений для АУ; 2 – панель генерации входных сообщений для БОУ; 3 – область управления верификацией

Вкладка генерации входных сообщений АУ и БОУ (см. рис. 4) разделена на две области: панель управления АУ (1) и панель управления БОУ (2) для ввода входных сообщений для АУ и БОУ соответственно.

Вкладка «Переменные АУ» (рис. 5) разделена на три области:

- панель значений входных портов АУ (1) с указанием имени входного порта, источника, который генерирует значения порта (БОУ / РУЧН), и значения битов порта, отображаемых цветом;
- панель значений входных и выходных глобальных переменных АУ (2) с указанием типа, имени переменной и ее текущего значения;
- панель режима управления входным портом АУ (3), позволяющая оператору устанавливать значения битов выделенного порта вручную.

Вкладка «Переменные БОУ» выглядит аналогично.

На вкладке «Отладочная информация АУ» (см. рис. 6) отображаются выходные сообщения АУ (1) и текущие состояния процессов АУ (2).

Вкладка «Отладочная информация БОУ» выглядит аналогично.

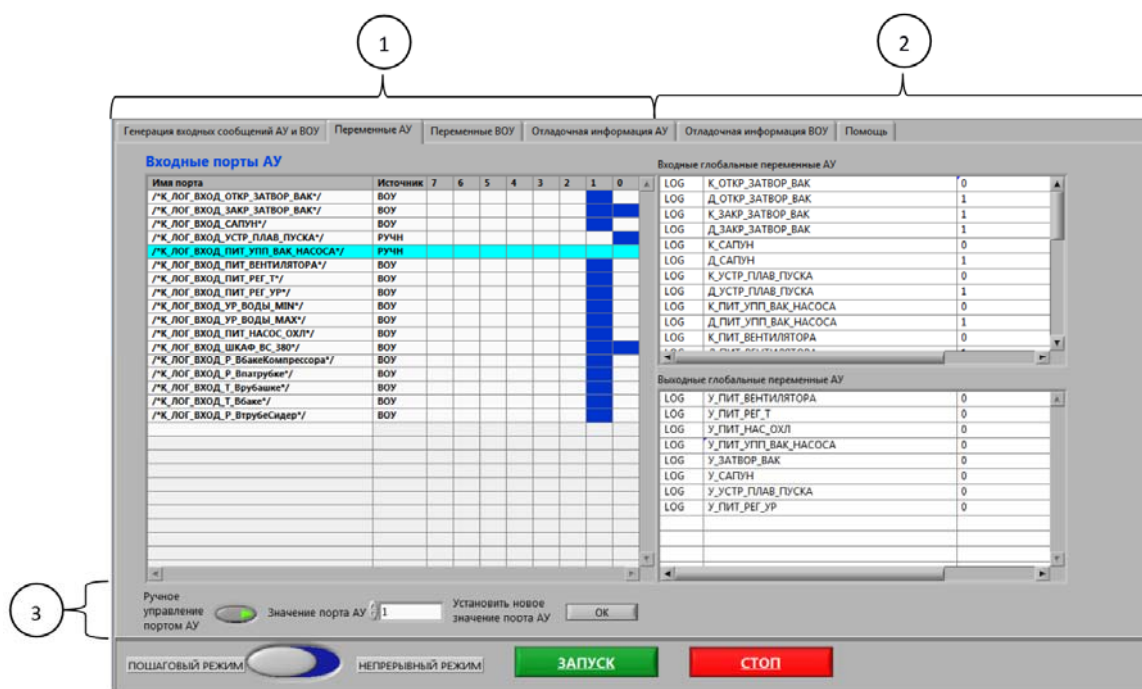


Рис. 5. Графический интерфейс оператора, вкладка «Переменные АУ»:

1 – панель значений входных портов АУ; 2 – панель значений входных и выходных глобальных переменных АУ; 3 – панель режима управления входным портом АУ

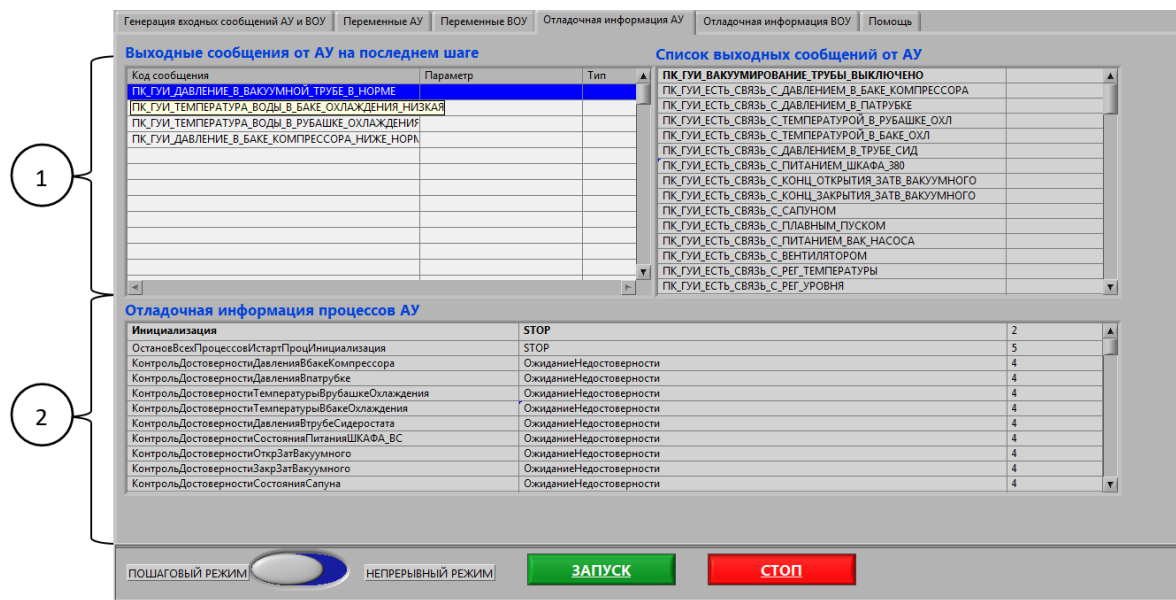


Рис. 6. Графический интерфейс оператора, вкладка «Отладочная информация АУ»:

1 – панель выходных сообщений АУ; 2 – панель текущих состояний процессов АУ

При запуске комплекса МЗКФ на основании конфигурационных файлов, создаваемых транслятором языка Reflex, настраивает вкладки ГИО (имена переменных, портов, процессов, входных / выходных сообщений) и передает управление ГИО. ГИО выделяет память под очереди сообщений, буферы входных / выходных переменных, буфер состояния процессов,

и затем передает указатели на выделенные области памяти алгоблокам. По началу верификации (кнопка «Запуск» на панели управления верификацией) ГИО циклически активизирует алгоблоки в определенной последовательности: сначала ВОУ, затем АУ. Создание тестовых ситуаций производится оператором через вкладки «Генерация входных сообщений АУ и ВОУ», «Переменные АУ», «Переменные ВОУ». Контроль корректности алгоритма управления ведется визуально. Разработка алгоритма управления соответствует общей схеме итерационной разработки (см. рис. 1).

Практическая апробация подхода

Решение было практически апробировано на задаче автоматизации Большого солнечного вакуумного телескопа (пос. Листвянка, Иркутская обл.) [5; 24]. В проекте был создан и верифицирован алгоритм управления системой вакуумирования. Верифицировалась работа алгоритма при создании вакуума в трубе телескопа, развакуумировании перед техническими работами, реакция алгоритма на изменение температуры окружающей среды, уровня воды в системе охлаждения, на разгерметизацию трубы телескопа, отказ отсечных клапанов, вакуумных заслонок, насосов, вытяжных вентиляторов, датчиков и исполнительных органов системы климат-контроля и т. д.

Верификация была проведена на территории разработчика и обеспечила значительное сокращение общей трудоемкости работ, в частности, продолжительности пусконаладки на целевом объекте была сокращена более чем в два раза.

Выявленные недостатки подхода:

- визуальный контроль за реакцией алгоритма и сложность анализа отображаемой информации не исключает вероятности пропуска ошибки при верификации;
- в силу того что на каждой итерации должна быть заново проверена реакция алгоритма в соответствии со списком тестовых ситуаций, возникает большое количество рутинных действий.

Заключение

В работе была предложена и рассмотрена автоматизированная схема итерационной разработки алгоритмов управления технологическим объектом. Итерационная схема разработки предполагает верификацию алгоритма управления, специфицированного на процесс-ориентированном языке Reflex, на виртуальном объекте управления до начала приемо-сдаточных испытаний на целевом объекте. Эффективность автоматизированной схемы верификации была подтверждена в проекте автоматизации Большого солнечного вакуумного телескопа при создании ПО подсистемы вакуумирования.

Список литературы

1. *Закревский А. Д.* Параллельные алгоритмы логического управления. М.: Эдиториал УРСС, 2003. 200 с.
2. *Wagner F., Schmuki R., Wagner T., Wolstenholme P.* Modeling software with finite state machines: a practical approach // Auerbach Publications. 2006. 390 p.
3. *Harel D.* Statecharts: A visual formalism for complex systems // Science of computer programming. 1987. Vol. 8.3. P. 231–274.
4. *Зюбин В. Е.* Программирование ПЛК: языки МЭК 61131-3 и возможные альтернативы // Промышленные АСУ и контроллеры. 2005. № 11. С. 31–35.
5. *Лях Т. В., Зюбин В. Е., Сизов М. М.* Опыт применения языка Reflex при автоматизации Большого солнечного вакуумного телескопа // Промышленные АСУ и контроллеры. 2016. № 7. С. 37–43.
6. *Зюбин В. Е.* К пятилетию стандарта IEC 1131-3. Итоги и прогнозы // Приборы и системы. Управление, контроль, диагностика. 1999. № 1. С. 64–71.
7. *Samek M., Montgomery P.* State Oriented Programming // Embedded Systems Programming. 2000. August. P. 22–43.

8. *Шальто А. А., Туккель Н. И.* SWITCH технология – автоматный подход к созданию программного обеспечения «реактивных» систем // Программирование. 2001. № 5. С. 45–62.
9. *Розов А. С., Зюбин В. Е., Нефедов Д. В.* Программирование встраиваемых микроконтроллерных систем на основе гиперпроцессов // Вестн. НГУ. Серия: Информационные технологии. 2017. Т. 15, № 4. С. 64–73.
10. *Зюбин В. Е.* «Си с процессами» – язык программирования логических контроллеров // Мехатроника, автоматизация, управление. 2006. № 12. С. 31–35.
11. *Зюбин В. Е.* Итерационная разработка управляющих алгоритмов на основе имитационного моделирования объекта управления // Автоматизация в промышленности. 2010. № 11. С. 43–48.
12. *Розов А. С., Лях Т. В., Краснов Д. В., Санжиев Е. С.* Практическая апробация языка IndustrialC на примере автоматизации установки термовакуумного напыления // Вестн. НГУ. Серия: Информационные технологии. 2017. Т. 15, № 3. С. 90–99.
13. *Гаранина Н. О., Зюбин В. Е., Лях Т. В.* Онтологический подход к организации шаблонов требований в рамках системы поддержки формальной верификации программных систем. // Системная информатика. 2017. № 9. С. 111–132.
14. *Shilov N. V., Garanina N. O.* Combined Logics of Knowledge, Time, and Actions for Reasoning about Multi-agent Systems. Knowledge Processing and Data Analysis // Lecture Notes in Computer Science. 2011. Iss. 6581. P. 48–58.
15. *Шелехов В. И.* Верификация и синтез программ сложения на базе правил корректности операторов // Моделирование и анализ информационных систем. 2010. Т. 17, № 4. С. 101–110.
16. *Clarke E. M., Gao S.* Model checking hybrid systems // International Symposium on Leveraging Applications of Formal Methods, Verification and Validation. Lecture Notes in Computer Science. 2014. Iss. 8803. P. 385–386.
17. *Clarke E. M., Grumberg O., Peled D.* Model checking. MIT Press, 1999. 314 p.
18. *Clarke E. M., Emerson A., Sistla P.* Automatic verification of finite-state concurrent systems using temporal logic specifications // ACM Transactions on Programming Languages and Systems. Vol. 8. Iss. 2. 1986. P. 244–263.
19. *Isermann R., Schaffnit J., Sinsel S.* Hardware-in-the-loop simulation for the design and testing of engine-control systems // Control Engineering Practice. 1999. Vol. 7. P. 643–653.
20. *Xiao B., Starke M., King D., Irmingier Ph., Herron A., Ollis B., Xue Y.* Implementation of system level control and communications in a Hardware-in-the-Loop microgrid testbed // IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT). 2016. P. 1–5.
21. *Зюбин В. Е.* Использование виртуальных объектов для обучения программированию информационно-управляющих систем // Информационные технологии. 2009. № 6. С. 79–82.
22. *Лях Т. В., Зюбин В. Е.* Автоматическая верификация алгоритмов управления сложными технологическими объектами на программных имитаторах // Математическое и компьютерное моделирование: Сб. материалов V Междунар. науч. конф., посвящ. памяти Р. Л. Долганова. Омск, 2017. С. 128–130.
23. *Зюбин В. Е., Калугин А. А.* Виртуальные лабораторные стенды: обучение программированию задач промышленной автоматизации // Промышленные АСУ и контроллеры. 2009. № 2. С. 39–44.
24. *Ковадло П. Г., Лубков А. А., Бевзов А. Н., Будников К. И., Власов С. В., Зотов А. А., Колобов Д. Ю., Курочкин А. В., Котов В. Н., Лылов С. А., Лях Т. В., Максимов А. С., Перебейнос С. В., Петухов А. Д., Пещеров В. С., Попов Ю. А., Русских И. В., Томин В. Е.* Система автоматизации Большого солнечного вакуумного телескопа // Автометрия. 2016. Т. 52, вып. 2. С. 97–106.

T. V. Liakh¹, V. E. Zyubin¹, N. O. Garanina²

¹ Institute of Automation and Electrometry SB RAS
1 Academician Koptyug Ave., Novosibirsk, 630090, Russian Federation

² A. P. Ershov Institute of Informatics Systems SB RAS
6 Academician Koptyug Ave., Novosibirsk, 630090, Russian Federation

antsys_nsu@mail.ru, zyubin@iae.nsk.su, garanina@iis.nsk.su

AUTOMATIC VERIFICATION OF CONTROL ALGORITHMS FOR COMPLEX TECHNOLOGICAL OBJECTS ON SOFTWARE SIMULATORS

The article is devoted to the analysis of process-oriented control algorithms, on the correspondence to input specifications. Article presents the general scheme of verification of the control algorithm, its implementation in an automated form and the results of practical approbation of the approach in the project for the automation of the Large Solar Vacuum Telescope.

Keywords: automation, industrial control algorithms, verification, process-oriented programming languages.

References

1. Zakrevsky A. D. Parallel algorithms of logical control. Moscow, Editorial URSS, 2003. 200 p. (in Russ.)
2. Wagner F., Schmuki R., Wagner T., Wolstenholme P. Modeling software with finite state machines: a practical approach. *Auerbach Publications*, 2006. 390 p.
3. Harel D. Statecharts: A visual formalism for complex systems. *Science of computer programming*, 1987, 8.3, p. 231–274.
4. Zyubin V. E. Programming of PLC: languages IEC 61131-3 and possible alternatives. *Industrial control systems and controllers*, 2005, no. 11, p. 31–35. (in Russ.)
5. Lyakh T. V., Zyubin V. Y., Sizov M. M. Experience of Reflex Language Application at Automation of the Large Solar Vacuum Telescope. *Industrial Automated Control Systems and Controllers*, 2016, no. 7, p. 37–43. (in Russ.)
6. Zyubin V. E. To the fifth anniversary of the IEC 1131-3 standard. Results and forecasts. *Devices and systems. Management, control, diagnostics*, 1999, no. 1, p. 64–71. (in Russ.)
7. Samek M., Montgomery P. State Oriented Programming. *Embedded Systems Programming*, August 2000, p. 22–43.
8. Shalyto A. A., Tukkell N. I. SWITCH technology – automatic approach to software development of “reactive” systems. *Programming*, 2001, no. 5, p. 45–62. (in Russ.)
9. Rozov A. S., Zyubin V. E., Nefedov D. V. Programming of Embedded Microcontroller Systems Based on Hyperprocesses. *Vestnik NSU. Series: Information Technologies*, 2017, vol. 15, no. 4, p. 64–73. (in Russ.)
10. Zyubin V. E. “C with processes” – programming language of logic controllers. *Mechatronics, automation, control*, 2006, no. 12, p. 31–35. (in Russ.)
11. Zyubin V. E. Iterative development of control algorithms based on simulation of control object. *Automation in industry*, 2010, no. 11, p. 43–48. (in Russ.)
12. Rozov A. S., Lyakh T. V., Krasnov D. V., Sanzhiev E. S. Practical approbation of the IndustrialC language on the example of automation of the thermovacuum sputtering installation. *Vestnik NSU. Series: Information Technologies*, 2017, vol. 15, no. 3, p. 90–99. (in Russ.)
13. Garanina N. O., Zyubin V. E., Lyakh T. V. Ontological approach to the organization of requirement templates within the system of support of formal verification of software systems. *System Informatics*, 2017, no. 9, p. 111–132. (in Russ.)
14. Shilov N. V., Garanina N. O. Combined Logics of Knowledge, Time, and Actions for Reasoning about Multi-agent Systems. Knowledge Processing and Data Analysis. *Lecture Notes in Computer Science*, 2011, iss. 6581, p. 48–58.

15. Shelekhov V. I. Verification and synthesis of addition programs based on the correctness rules of operators. *Modeling and analysis of information systems*, 2010, vol. 17, no. 4, p. 101–110. (in Russ.)
16. Clarke E. M., Gao S. Model checking hybrid systems. *International Symposium on Leveraging Applications of Formal Methods, Verification and Validation. Lecture Notes in Computer Science*, 2014, iss. 8803, p. 385–386.
17. Clarke E. M., Grumberg O., Peled D. Model checking. MIT Press, 1999, 314 p.
18. Clarke E. M., Emerson A., Sistla P. Automatic verification of the finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 1986, vol. 8, iss. 2, p. 244–263.
19. Isermann R., Schaffnit J., Sinsel S. Hardware-in-the-loop simulation for the design and testing of engine-control systems. *Control Engineering Practice*, 1999, vol. 7, p. 643–653.
20. Xiao B., Starke M., King D., Irminger Ph., Herron A., Ollis B., Xue Y. Implementation of the system-level control and communications in a Hardware-in-the-Loop microgrid testbed. *IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2016, p. 1–5.
21. Zyubin V. E. Use of virtual objects for teaching programming of information-control systems. *Information technologies*, 2009, no. 6, p. 79–82. (in Russ.)
22. Liakh T. V., Zyubin V. E. Automatic verification of control algorithms for complex technological objects on software simulators. *Proc. of the V International Scientific Conference Mathematical and computer modeling dedicated to the memory of R. L. Dolganov*. Omsk, 2017, p. 128–130. (in Russ.)
23. Zyubin V. E., Kalugin A. A. Virtual laboratory stands: teaching programming of problems of industrial automation. *Industrial Automated Control Systems and Controllers*, 2009, no. 2, p. 39–44. (in Russ.)
24. Kovadlo P. G., Lubkov A. A., Bezvov A. N., Budnikov K. I., Vlasov S. V., Zotov A. A., Kolobov D. U., Kurochkin A. V., Kotov V. N., Lylov S. A., Lyakh T. V., Maksimov A. S., Perebeynos S. V., Petukhov A. D., Peshcherov V. S., Popov U. A., Russkih I. V., Tomin V. E. Automation system of the Large Solar Vacuum Telescope. *Optoelectronics, Instrumentation and Data Processing*, 2016, vol. 52, no. 2, p. 97–106. (in Russ.)

Received 23.09.2018

For citation:

Liakh T. V., Zyubin V. E. Garanina N. O. Automatic Verification of Control Algorithms for Complex Technological Objects on Software Simulators. *Vestnik NSU. Series: Information Technologies*, 2018, vol. 16, no. 4, p. 85–94. (in Russ.) DOI 10.25205/1818-7900-2018-16-4-85-94