НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
март–апрель 2023 Том 23 № 2 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
March–April 2023 Vol. 23 No 2 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

ПИСЬМА В РЕДАКЦИЮ
LETTERS

doi: 10.17586/2226-1494-2023-23-2-436-438
УДК 004.4′242

Валидация автоматных спецификаций

Анатолий Абрамович Шалыто✉

Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
shalyto@mail.ifmo.ru✉, https://orcid.org/0000-0002-2723-2077

Аннотация
Изложен новый взгляд на обеспечение качества автоматных программ. При этом вместо термина «верификация автоматных программ» предложено использовать термины «верификация автоматных моделей» и «валидация автоматных спецификаций». Первый термин применим при наличии формальной спецификации, а второй — при ее отсутствии, что более характерно для практики. Это позволяет более осмысленно подходить к пониманию того, как обеспечивать качество автоматных программ.

Ключевые слова
автоматы, программы, спецификация, верификация, валидация

Ссылка для цитирования: Шалыто А.А. Валидация автоматных спецификаций // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 2. С. 436–438. doi: 10.17586/2226-1494-2023-23-2-436-438

Validation of state machine specifications

Anatoly A. Shalyto✉

ITMO University, Saint Petersburg, 197101, Russian Federation
shalyto@mail.ifmo.ru✉, https://orcid.org/0000-0002-2723-2077

Abstract
The paper presents a new view on the quality assurance of state machine programs. Instead of the term “verification of state machine programs”, it is proposed to use the terms “verification of state machine models” and “validation of state machine specifications”. The first of which is applicable in the presence of a formal specification, and the second — in its absence, which is more typical for practice. This allows a more meaningful approach to understanding how to ensure the quality of state machine programs.

Keywords
state machine, programs, specification, verification, validation

For citation: Shalyto A.A. Validation of state machine specifications. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, vol. 23, no. 2, pp. 436–438 (in Russian). doi: 10.17586/2226-1494-2023-23-2-436-438

Под верификацией[1] обычно понимают проверку неформально построенной прикладной программы на предмет выполнения формальной спецификации. Этот процесс надо проводить заново для каждой вновь созданной прикладной программы. Считается, что верификация позволяет установить, что «мы создали продукт таким, каким и намеревались его сделать», а валидация подтверждает, что «мы создали правильный продукт». Обратим внимание на то, что в данном случае формальная спецификация строится неформально (ее не по чему верифицировать), а изменения, вносимые в нее, относятся к валидации. При этом тестирование

[1] Кулямин В.В. Методы верификации программного обеспечения // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы», 2008. 111 с. [Электронный ресурс]. Режим доступа: https://www.ispras.ru/publications/methods_of_software_verification.pdf (дата обращения: 29.11.2022).

может рассматриваться, как разновидность валидации, которая проводится с целью определения соответствия поведения системы ее ожидаемому поведению на конечном наборе тестов. Если формальная спецификация становится исполняемой, то вместо верификации каждой из генерируемых прикладных программ верификация может проводиться только один раз — однократно верифицируется только программа-генератор прикладных программ. Если квалификации создателей генератора для его верификации не хватает, то он может быть валидирован — проверен, например статистически, на правильность построения с его помощью прикладных программ. Поэтому если спецификация является исполняемой, а программа-генератор верифицирована или валидирована, то проблема верификации прикладных программ исчезает!

Возможны два варианта (случая) создания программ: при наличии формальной спецификации в начале проектирования, и при ее отсутствии на этой стадии создания системы. В первом случае при рассматриваемом подходе можно проводить верификацию не самой программы, а неформально построенной формальной модели, по которой она генерируется. При использовании автоматного программирования такая модель — система графов переходов конечных автоматов. Эта модель для программы является исполняемой спецификацией, которая должна быть проверифицирована по исходной спецификации, так как предполагается, что она существует. В этом случае можно говорить о «верификации автоматной модели». Однако формальная спецификация в начале проектирования существует крайне редко: либо для очень простых объектов, либо для невероятно ответственных объектов. Таким простым объектом[1], например, являются часы с будильником, поведение системы управления которым описывается всего одним графом переходов автомата всего с тремя состояниями. Показано, что даже такую модель трудно верифицировать: она успешно выполнилась для исходно построенной формальной спецификации в виде определенного числа темпоральных правил, но, когда в спецификацию были внесены изменения — увеличено число темпоральных правил, в графе переходов удалось обнаружить ошибку. Обратим внимание на то, что в данном случае формальная спецификация строится неформально — ее не по чему верифицировать, а изменения, вносимые в нее, относятся к валидации. Для очень ответственных объектов при наличии уникальных специалистов, как это было в свое время при автоматизации Лондонского метро, верификация проводится с дальнейшей валидацией в течение всего жизненного цикла.

Во втором случае при проектировании систем управления сложными технологическими объектами формальная спецификация в начале проектирования отсутствует. От Заказчика в лучшем случае можно получить только разрозненные сведения и знания о том, как система должна работать, причем обычно в основном режиме. Все тонкости работы по разным причинам на ранних этапах создания системы Заказчик описать не может. Какие-то знания (возможно, и основные) добавляет Разработчик системы управления, если он опытный специалист. По информации от Заказчика и Разработчика можно неформально писать программу (остаться без формальной спецификации в наглядной форме) или строить формальную спецификацию (зафиксировать указанную выше информацию на каком-либо формальном языке спецификаций). Первый путь — традиционный [1], но, по моему мнению, тупиковый, поэтому лучше использовать второй путь с генерацией программы по неформально построенной формальной спецификации. При этом возникает вопрос, какой математический аппарат применять при ее построении. Мною в качестве формальной спецификации было предложено использовать систему графов переходов конечных автоматов. По этой спецификации программа должна строиться (генерироваться) формально и изоморфно. При этом она не только будет соответствовать спецификации, но и будет «внешне похожа» на нее. Правильность такой спецификации требует проверки — проведения валидации. «Правильность» неформальное понятие, оно должно «устраивать»: оборудование (при существующей на данный момент спецификации не должно происходить его аварий и поломок) и всех специалистов, участвующих в создании системы, которые делают все возможное, чтобы и в дальнейшем оборудованию было «комфортно». Любое испытание можно рассматривать как валидацию, проводимую с целью подтверждения правильности спецификации и уточнения ее при необходимости. Уточнение обеспечивается путем внесения соответствующих изменений в существующую к этому моменту систему графов переходов. Однако, так как в ходе корректировки могут быть внесены ошибки, то после этого необходимо снова провести испытания и т. д. Эксплуатация системы (тем более опытная) также может рассматриваться как уточнение спецификации. Только в момент списания системы, если она существовала в единственном экземпляре, можно считать, что, наконец-то, получена спецификация, которая является окончательной, и то при условии, что на ее основе не будет создаваться модификация системы. Таким образом, в каждый момент времени существует формальная спецификация, которая на любом этапе валидации может потребовать корректировки.

Из изложенного следует, что уточнение формальной спецификации системы управления проводится путем валидации в течение всего ее жизненного цикла, а при создании модификации и после его завершения. К средствам валидации, в частности, относится тестирование, а также проверка выполнения темпоральных свойств, используемых в традиционной верификации [1]. Обратим внимание, что в предлагаемом подходе система графов переходов является не только исполняемой спецификацией, но и языком программирования. При этом после валидации спецификации, верификация или даже валидация программы, построенной по ней, не требуется! Предложенный подход использован в работе

[1] Ульянцев В.И., Шалыто А.А. О верификации простых программ со сложным поведением. 2013 [Электронный ресурс]. Режим доступа: http://is.ifmo.ru/works/2013/ulyantsev-shalyto-verification.pdf (дата обращения: 28.11.2022).

[2], который в динамике проиллюстрирован в видеоприложении, созданном совместно с Ю.Ю. Янкиным[1].

Таким образом, можно сделать вывод, что, в свое время назвав книгу «Верификация автоматных программ» [3], авторы поступили недостаточно корректно, так как уже при ее написании предполагали, что автоматные программы строятся не эвристически, а формально по спецификации. Поэтому она должна была называться либо «Верификация автоматных моделей», либо «Валидация автоматных спецификаций», причем второе название является значительно более практичным. Исходя из изложенного, выражение «Верификация автоматных программ» [4] можно использовать как жаргон для понятия «Валидация автоматных спецификаций», так же как широко известное выражение «Минимизация булевых функций», которое повсеместно применяется вместо понятия «Минимизация булевых формул» [5]. Последнее связано с тем, что булеву функцию (таблицу истинности), если она существенно зависит от всех своих переменных, нельзя проминимизировать — в отличие от булевой формулы, для которой в большинстве случаев это возможно.

[1] Янкин Ю.Ю., Шалыто А.А. Метод создания программного обеспечения модулей, выполненных на основе программируемых логических интегральных схем. Видеоприложение, 2012 [Электронный ресурс]. Режим доступа: https://www.youtube.com/watch?v=YNWdmnwHZi8 (дата обращения: 29.11.2022).

Литература

1. Карпов Ю.Г. Model Checking. Верификация параллельных и распределенных программных систем. СПб.: БХВ-Петербург, 2010. 560 с.
2. Янкин Ю.Ю., Шалыто А.А. Автоматное программирование ПЛИС в задачах управления электроприводом // Информационно-управляющие системы. 2011. № 1. С. 50–56.
3. Вельдер С.Э., Лукин М.А., Шалыто А.А., Яминов Б.Р. Верификация автоматных программ. СПб.: Наука, 2011. 244 с.
4. Кузьмин Е.В., Соколов В.А. Моделирование, спецификация и верификация «автоматных» программ // Программирование. 2008. Т. 34. № 1. С. 38–60.
5. Шалыто А.А. Логическое управление. Методы аппаратной и программной реализации алгоритмов. СПб.: Наука, 2000. 780 с.

References

6. Karpov Iu.G. *Model Checking. Verification of Parallel and Distributed Software Systems*. St. Petersburg, BHV Petersburg Publ., 2010, 560 p. (in Russian)
7. Yankin Y.Y., Shalyto A.A. A method of finite-state machine realization in electric motor drives control. *Information and Control Systems*, 2011, no. 1, pp. 50–56. (in Russian)
8. Velder S.E., Lukin M.A., Shalyto A.A., Iaminov B.R. *Verification of Automata-Based Programs*. St. Petersburg, Nauka Publ., 2011, 244 p. (in Russian)
9. Kuzmin E.V., Sokolov V.A. Modeling, specification, and verification of automaton programs. *Programming and Computer Software*, 2008, vol. 34, no. 1, pp. 27–43. https://doi.org/10.1134/s0361768808010040
10. Shalyto A.A. *Logical Control: Methods for Hardware and Software Implementation of Algorithms*. St. Petersburg, Nauka Publ., 2000, 789 p. (in Russian)

Автор

Шалыто Анатолий Абрамович — доктор технических наук, профессор, главный научный сотрудник, профессор, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, SC 56131789500, https://orcid.org/0000-0002-2723-2077, shalyto@mail.ifmo.ru

Author

Anatoly A. Shalyto — D.Sc., Full Professor, Chief Researcher, ITMO University, Saint Petersburg, 197101, Russian Federation, SC 56131789500, https://orcid.org/0000-0002-2723-2077, shalyto@mail.ifmo.ru

Статья поступила в редакцию 29.11.2022
Одобрена после рецензирования 07.01.2023
Принята к печати 26.03.2023

Received 29.11.2022
Approved after reviewing 07.01.2023
Accepted 26.03.2023