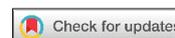


# МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

## MATHEMATICAL MODELING



УДК 004.942  
<https://doi.org/10.37661/1816-0301-2022-19-1-88-95>

Обзорная статья  
Original Paper

## Методология разработки программного обеспечения с использованием модели распределенных объектно-ориентированных стохастических гибридных систем

Р. Е. Шарыкин

Белорусский государственный университет,  
пр. Независимости, 4, Минск, 220030, Беларусь  
✉ E-mail: [sharykin@bsu.edu](mailto:sharykin@bsu.edu)

### Аннотация

В статье представляется методология разработки программного обеспечения на основе модели распределенных объектно-ориентированных стохастических гибридных систем. Предлагается ориентироваться на создание математической модели для рассматриваемой системы вместе с ее спецификацией на всех этапах разработки целевого программного обеспечения.

Выделяются такие этапы разработки, как построение предварительной математической модели посредством составления ее спецификации, выбор и спецификация метрик системы, статистический анализ модели, апробация в условиях, приближенных к реальным, анализ с применением аналитических методов и реализация. Предлагаются формализм для описания рассматриваемой системы и подход к ее анализу, согласно результатам которого спецификация модели и соответствующая ей математическая модель модифицируются посредством выполнения этапов разработки. Такой подход позволяет получить на выходе не только готовое программное обеспечение, но и математическую модель с изученными свойствами, реализацией которой является данное программное обеспечение.

**Ключевые слова:** математическое моделирование, переписывающая логика, статистический анализ, стохастические гибридные системы, разработка программного обеспечения

**Для цитирования.** Шарыкин, Р. Е. Методология разработки программного обеспечения с использованием модели распределенных объектно-ориентированных стохастических гибридных систем / Р. Е. Шарыкин // Информатика. – 2022. – Т. 19, № 1. – С. 88–95. <https://doi.org/10.37661/1816-0301-2022-19-1-88-95>

**Конфликт интересов.** Автор заявляет об отсутствии конфликта интересов.

---

Поступила в редакцию | Received 25.02.2022  
Подписана в печать | Accepted 01.03.2022  
Опубликована | Published 29.03.2022

# Methodology of software development with the use of the model of distributed object-based stochastic hybrid systems

Raman E. Sharykin

*Belarusian State University,  
av. Nezavisimosti, 4, Minsk, 220030, Belarus*  
✉E-mail: sharykin@bsu.edu

## Abstract

Software development methodology based on the model of distributed object-based stochastic hybrid systems is proposed. Such mathematical model is planned to be created for the system being designed along with the system specification at all stages of the software development.

The following stages of the development are considered as building of preliminary mathematical model by designing its specification, choosing and specification of system metrics, statistical analysis of the model, approbation of the mathematical model in conditions close to real, analysis by analytical methods and the implementation. The formalism is proposed for describing the system under consideration and an approach to its analysis. At the steps of the methodology, we adjust the model specification and its corresponding mathematical model in accordance with the results of the analysis. This approach allows to develop not only the software, but also a mathematical model with its properties, which implementation is the resulting software.

**Keywords:** mathematical modeling, rewriting logic, statistical analysis, stochastic hybrid systems, software development.

**For citation.** Sharykin R. E. *Methodology of software development with the use of the model of distributed object-based stochastic hybrid systems*. *Informatika [Informatics]*, 2022, vol. 19, no. 1, pp. 88–95 (In Russ.). <https://doi.org/10.37661/1816-0301-2022-19-1-88-95>

**Conflict of interest.** The author declare of no conflict of interest.

**Введение.** Ввиду растущей сложности распределенных систем, разрабатываемых в настоящее время, применение формальных методов для понимания динамики таких систем на этапах их разработки представляется важной задачей. Одно из неудобств применения формальных методов в данной области объясняется трудностью определения подходящей формальной модели, например сети Петри или дискретной марковской цепи. После того как выбор формальной модели сделан, заменить ее будет очень затратно в случае, если обнаружится, что другая формальная модель лучше подходит для описания рассматриваемой системы. В связи с этим важной задачей является разработка подхода, который бы позволял не ограничиваться возможностями отдельного формализма и проводить анализ системы так, чтобы еще до выбора некоторой формальной модели можно было на основе экспериментов учесть аспекты, которые влияют на ее дизайн как можно более полно.

Сложности, возникающие при применении формальных методов в процессе разработки программного обеспечения, описаны в следующих работах.

В статье [1] рассматривается использование мониторов реального времени исполнения для верификации заранее заданных свойств безопасности. Проводится формальное исследование такого подхода, разрабатывается семантика языка спецификации свойств, операционная семантика процесса встраивания мониторов в систему и доказывается корректность этого метода. Ограничения проводимого исследования включают рассмотрение системы в виде ее реализации на конкретном языке программирования, в данном случае Java. При переходе к реализации могут утрачиваться некоторые аспекты системы, например замена дифференциальных уравнений разностными схемами, что влечет за собой появление погрешностей.

В работе [2] предлагается модель параллельных стохастических гибридных систем. Стохастические гибридные системы [3] позволяют моделировать системы, имеющие как дискретную,

так и непрерывную составляющие в условиях вероятностной неопределенности. В данной модели система моделируется множеством агентов, динамика каждого агента следует динамике некоторой стохастической гибридной системы [3] и агенты взаимодействуют друг с другом посредством общих переменных. Показывается, что композиция агентов является корректной операцией. Ограничения данного исследования включают сложность моделирования распределенных систем, сообщающихся путем асинхронных сообщений, а также отсутствие объектно-ориентированного подхода, позволяющего единообразно задавать конструкцию многих объектов в рамках единой структуры.

Для того чтобы учесть аспекты, возникающие при разработке систем (распределенных, асинхронных, стохастических, объектно-ориентированных), необходим формализм, имеющий самые широкие выразительные возможности. Формализм переписывающей логики [4] хорошо удовлетворяет требованию не ограничиваться отдельным «классическим» формализмом. Многие общепринятые виды формализма имеют известное представление в виде «переписывающих теорий» [4, 5]. Вероятностная версия переписывающей логики [6, 7] является еще более подходящим выбором, так как позволяет моделировать стохастические системы. Переписывающие теории дают возможность задавать очень широкий спектр систем, но отсутствие встроенных в математическую модель структур для описания указанных выше аспектов делает их недостаточно удобными при задании систем, состоящих из объектов, которые являются распределенными, объектно-ориентированными, стохастическими и с асинхронной коммуникацией.

Структура модели распределенных объектно-ориентированных стохастических гибридных систем (РООГС) [8] обусловлена факторами модульности, иерархичности, объектно-ориентированности и распределенной природы таких систем, т. е. позволяет задавать их простым и понятным образом. Преимущество модели РООГС состоит в возможности задавать систему как набор объединенных в классы объектов, имеющих внутреннее состояние, которое подчиняется динамике, определяемой системой стохастических дифференциальных уравнений, и общающихся посредством асинхронных сообщений. Спецификация модели РООГС на языке SHYMaude [9] легко транслируется в спецификацию переписывающей логики, тем самым наследуя все ее преимущества, а также может выполняться в системе Maude [10], открывая возможности для непосредственной имитации модели. Такой подход к спецификации системы значительно приближен к реальным задачам и позволяет определять системы формально, но максимально близко к тому, как они устроены на практике.

**Методология разработки программного обеспечения.** Использование переписывающей логики позволяет начать применение формальных методов на ранних этапах проектирования, в то же время не ограничивая разработчика определенным формализмом. Основанный на переписывающей логике формализм может быть сжатым, интуитивно понятным и хорошо подходит для спецификации распределенных параллельных систем с асинхронной коммуникацией.

Спецификации, выполненные на языке SHYMaude, после трансляции в язык Maude могут дорабатываться с помощью системы Maude [10], что позволяет симулировать и улучшать модель системы в самом начале ее разработки. Как только разработчик остается удовлетворенным полученной спецификацией, он может выбрать наиболее подходящую математическую модель с известным представлением в переписывающей логике для формального доказательства критически важных свойств системы либо нахождения более существенных недостатков, которые невозможно выявить с помощью техники имитаций. Язык Maude также представляет средства, облегчающие проведение формальных доказательств в формализме переписывающей логики [11].

Следующим важным вопросом является анализ системы на ранних этапах ее разработки. Желательный метод должен быть легким в том смысле, что он может помочь разработчику сконцентрироваться на модели и ее параметрах. В работе [9] предложен метод, с помощью которого можно получить быструю оценку основных поведенческих свойств системы и экспериментально их изучить, позволяя потратить усилия на проведение более сложных формальных процедур валидации на поздних этапах, когда пространство возможностей для дизайна модели уже сокращено.

В рамках методологии, основанной на работе [9], автор предлагает работать со спецификацией системы, выполненной с помощью языка SHYMaude. Для анализа системы рекомендуется использовать:

- инструмент MultiVeStA [12] для выполнения имитаций спецификаций на языке Maude;
- язык MultiQuaTEh [12] для формального задания свойств исследуемой системы. MultiQuaTEh является расширением языка QuaTEh [13], позволяющим более компактно задавать метрики систем;
- клиент-серверную архитектуру для выполнения распределенных имитаций.

В процессе анализа MultiVeStA использует статистический метод Монте-Карло и предоставляет возможность генерации реализаций до достижения предустановленного уровня точности результата. Также система обеспечивает выполнение распределенных вычислений, основанное на клиент-серверном взаимодействии [12, 13].

Опишем поэтапно методологию разработки:

1. Система специфицируется с помощью языка SHYMaude. Спецификация одновременно определяет модель РООСГС. На данном этапе работа по спецификации сродни работе по реализации системы на языке программирования, что значительно облегчает задачу.

2. Выбираются и специфицируются на языке MultiQuaTEh метрики, оценки которых представляются важными разработчику.

3. Используется система MultiVeStA, проводящая статистическую оценку метрик для спецификации системы с помощью метода доверительных интервалов на основе метода Монте-Карло.

4. Полученные результаты анализируются. В случае обнаружения недочетов в спецификации и (или) способов ее «улучшения» спецификация и соответствующая ей модель корректируются. Возвращаемся к этапу 3.

5. Проводится апробация системы. Система реализуется на языке высокого уровня (например, Java), предпочтительно имеющем известное представление в переписывающей логике. Возможно проведение статистического анализа, аналогичного проводимому на этапе 3. В случае нахождения аспектов системы, требующих коррекции и (или) дополнительного анализа (например, возможности реализации некоторого действия несколькими способами), проводятся коррекция и (или) дополнительный анализ, по результатам которого делается выбор предпочтительного способа реализации. Одновременно корректируется исходная модель и, если это видится целесообразным, выполняется переход к этапу 3. Если результаты удовлетворительны и делается вывод, что предварительный анализ модели можно считать законченным, переходим к этапу 6.

6. Имеются две возможности усиления результатов, полученных с помощью предыдущих пунктов:

– *аналитическое исследование свойств.* По имеющейся спецификации подбирается подходящая математическая модель и аналитически исследуется с доказательством интересующих свойств модели;

– *автоматическое доказательство свойств.* Из системы удаляются вероятности и недетерминизм, и используется система автоматического доказательства теорем системы Maude [14] для получения дополнительных доказанных утверждений относительно свойств системы. С помощью данного подхода могут доказываться утверждения, не имеющие отношение к вероятностной природе системы, например утверждение, что ни при каких сценариях поведения одного из участников протокола определенные данные не могут быть получены из доступного ему трафика в незашифрованном виде.

Данный этап в основном может потребоваться для приложений, требующих крайне высокой надежности результатов.

7. Реализация системы на практике. Если природа системы позволяет выделять отдельные прогоны в процессе ее работы, то на данном этапе также можно проводить фоновый статистический анализ для оценки основных метрик системы в реальных условиях.

В предлагаемой методологии формальные методы используются в трех аспектах: для задания модели (модель РООСГС), задания метрик (язык QuaTEh) и получения значимых оценок метрик (статистический анализ).

**Апробация методологии.** В целях апробации изложенной методологии были выполнены разработки системы стохастической коллаборационной защиты от вирусов, системы одного окна и системы закупок предприятия с использованием обратных аукционов.

**Система стохастической коллаборационной защиты от вирусов.** В работе [15] методология была использована для разработки предварительной модели стохастической групповой системы защиты от вирусов и ее анализа. Продемонстрировано, как получение оценки основных численных метрик системы, обычно используемых для оценки систем защиты от вирусов, дает возможность оценить систему до фиксации формальной модели и проведения ее углубленного анализа. Из результатов анализа видно, что система защиты позволяет «сохранить» значительный процент узлов незараженными при атаке вируса на каждом шаге распространения, выбирающего следующий атакуемый узел в соответствии с равномерным распределением на множестве всех узлов сети. Также изучено влияние алгоритма выбора групп оповещения на эффективность системы защиты. Анализ показывает, что важным условием эффективности системы является равномерное покрытие всего множества узлов группами оповещения при рассматриваемом типе вируса.

Вместе с тем было обнаружено, что существенное упрощение алгоритма выбора групп оповещения не повлияло на эффективность системы при проведении эксперимента. Это указывает на потенциальную возможность упрощения системы с сохранением ее свойств и может быть более тщательно проверено на более поздних этапах разработки.

Полученные результаты дают возможность утверждать, что применение изложенного подхода статистического анализа систем на этапе проектирования позволяет изучать важные аспекты систем до их анализа аналитическими методами, исправлять какие-либо недостатки и определять некоторые потенциальные «упрощения» отдельных компонентов.

В работе [16] была проведена апробация реализации системы стохастической коллаборационной защиты от вирусов в условиях, приближенных к реальным. Рассматривались две вариации алгоритма с использованием протоколов TCP/IP и UDP в части рассылки оповещений об обнаружении вирусов и назначении групп оповещений. Установлено, что применение протокола UDP приводит к увеличению статистической эффективности системы защиты. Исследования возможного потенциала увеличения эффективности в рамках исходной спецификации на языке переписываемой логики проводились путем снижения задержек в доставке сообщений. Так как протокол UDP не предусматривает создания полноценного соединения, доставка сообщения вызывает значительно меньшие задержки, чем доставка при использовании протокола TCP/IP. Предполагалось, что вирусы для своего распространения используют в основном протокол TCP/IP.

При исследовании влияния размера группы оповещения и общего количества узлов с сохранением в процентном соотношении размера групп было выяснено, что увеличение размера общего количества узлов приводит к заметному росту эффективности системы [16]. Вместе с тем для некоторых сценариев заражения это сопряжено с увеличением нагрузки на сеть ввиду одновременного увеличения количества оповещений в сети. Предложен механизм иерархической организации системы защиты для предотвращения быстрого роста данной нагрузки.

Таким образом, предлагаемая система может использоваться для защиты сетей, состоящих из того количества узлов, которое встречается в реальных условиях.

**Система одного окна.** В работе [17] проиллюстрировано применение методологии на примере разработки системы одного окна, которая была специфицирована с помощью языка SHYMaude [9]. Сформулированы четыре метрики, измеряющие такие параметры, как длина очереди документов, время их обработки, загруженность сотрудников, и позволяющие оценить эффективность такой системы. Для проведения статистического анализа данных метрик использовался инструмент MultiVeStA. Предложенные метрики были специфицированы с помощью языка MultiQuaTEh. Также определено оптимальное количество сотрудников, задействованных в системе на обработке документов. При оптимальном количестве сотрудников сохраняется баланс в виде приемлемых параметров: максимально наблюдаемой длины очереди документов на рассмотрение, времени обработки документов и загруженности сотрудников.

Для того чтобы максимально приблизить систему к практическому применению, введено управление количеством сотрудников, осуществляемое менеджером. При этом был обнаружен

недостаток такой доработки: на раннем этапе функционирования системы предложенный алгоритм неудачно управляет количеством сотрудников, что приводит к значительному росту значения одной из целевых метрик, а именно максимальной наблюдаемой длины очереди, выводя ее из желаемого интервала. Для исправления этого недостатка было предложено ввести правило, при котором менеджер начинает управлять количеством сотрудников только после того, как длина очереди достигает предварительно установленной нижней границы. После устранения недостатка было обнаружено, что максимальная наблюдаемая длина очереди находится в желаемом интервале.

Данное исследование показывает, что использование предлагаемой методологии дает возможность своевременно исправлять грубые ошибки при проектировании систем одного окна.

**Система закупок предприятия.** В работе [18] методология использована для предварительной оценки модели системы закупок предприятия на примере модели закупок на основе повторяемого обратного аукциона. Модель системы закупок предприятия была специфицирована для двух типов аукционов с помощью языка SHYMaude. Были сформулированы метрики, представляющие интерес с точки зрения оценки эффективности предложенной системы: цена, по которой будет производиться поставка товара после единичного проведения аукциона, и цена поставки, усредненная по множеству проведенных аукционов. Также были определены параметры, которые могут влиять на значения второй рассматриваемой метрики. Для выполнения статистического анализа метрик использован инструмент MultiVeStA, предложенные метрики специфицированы с помощью языка MultiQuaTEh.

В результате анализа модели на тестовых данных было обнаружено, что применение аукциона первой цены дает более низкую цену поставки товара, но эта разница незначительна.

Увеличение времени проведения аукциона приводит к снижению цены поставки. Вместе с тем основное снижение цены происходит в течение времени, которое может доставлять сложности с проведением аукциона, так как время его проведения может быть ограничено. Можно определить некоторое пороговое значение количества участников, после которого дополнительные участники не приводят к заметному уменьшению цены закупки. При увеличении количества раундов аукциона цена снижается в сторону предельно допустимой минимальной цены продажи, установленной в системе, без возможности четко обозначить некоторую переломную точку. Использование повторных аукционов позволяет достичь более низких значений цены поставки товара, чем простое увеличение длительности отдельно взятого аукциона без повторов.

Таким образом, применение техники повторных аукционов дает альтернативу увеличению длительности аукциона для достижения меньшей средней цены поставки, рассчитываемой за долгосрочный период.

**Заключение.** В статье представлена методология разработки программного обеспечения на основе модели РООСГС. Для формального моделирования разрабатываемых систем предлагается использование данной модели, а для их спецификации – языка SHYMaude. Анализ систем рекомендуется выполнять с помощью статистического подхода. Для более тщательного изучения свойств системы дополнительно к статистическому анализу могут быть применены аналитический подход [5] и (или) подход, основанный на использовании техники автоматического доказательства теорем [11].

Согласно представленной методологии поэтапный процесс разработки системы включает: спецификацию системы; спецификацию требований; трансляцию спецификации системы в ее исполняемую версию, одновременно являющуюся формальной моделью благодаря свойствам переписывающей логики; исследование статистическими методами требований разработчиков, сформулированных в виде метрик; анализ результатов и доработку исходной спецификации; повторение процесса до получения удовлетворяющего требованиям разработчиков результата.

Процесс доработки может быть повторен на этапе апробации системы в условиях, максимально приближенных к реальным, и продолжен на этапе реализации уже практической версии системы. С переходом на каждый последующий этап цена изменений первоначальной модели возрастает, что подтверждает целесообразность разделения процесса разработки на выделенные в статье этапы.

**Список использованных источников**

1. Formalizing Java-MaC / U. Sammapun [et al.] // *Electronic Notes in Theoretical Computer Science*. – 2003. – Vol. 89, iss. 2. – P. 171–190.
2. Bernadsky, M. Structured modeling of concurrent stochastic hybrid systems / M. Bernadsky, R. Sharykin, R. Alur // *Lecture Notes in Computer Science*. – 2004. – Vol. 3253. – P. 309–324.
3. Bujorianu, M. L. Toward a general theory of stochastic hybrid systems / M. L. Bujorianu, J. Lygeros // *Lecture Notes in Control and Information Science*. – 2006. – Vol. 337. – P. 3–30.
4. Meseguer, J. Conditional rewriting logic as a unified model of concurrency / J. Meseguer // *Theoretical Computer Science*. – 1992. – Vol. 96, iss. 1. – P. 73–155.
5. Martí-Oliet, N. Rewriting logic: roadmap and bibliography / N. Martí-Oliet, J. Meseguer // *Theoretical Computer Science*. – 2002. – Vol. 285, iss. 2. – P. 121–154.
6. Agha, G. A. PMAude: Rewrite-based specification language for probabilistic object systems / G. A. Agha, J. Meseguer, K. Sen // *Electronic Notes in Theoretical Computer Science*. – 2006. – Vol. 153, iss. 2, no. 2. – P. 213–239.
7. A rewriting based model for probabilistic distributed object systems / N. Kumar [et al.] // *Lecture Notes in Computer Science*. – 2003. – Vol. 2884. – P. 32–46.
8. Шарыкин, Р. Е. Модель распределенных объектно-ориентированных стохастических гибридных систем / Р. Е. Шарыкин, А. Н. Курбацкий // *Журнал Белорусского государственного университета. Математика. Информатика*. – 2019. – № 2. – С. 52–61.
9. Шарыкин, Р. Е. Верификация распределенных объектно-ориентированных стохастических гибридных систем / Р. Е. Шарыкин, А. Н. Курбацкий // *Вестник Гродненского государственного университета имени Янки Купалы. Сер. 2. Математика. Физика. Информатика, вычислительная техника и управление*. – 2019. – Т. 9, № 2. – С. 123–133.
10. Maude: Specification and programming in rewriting logic / M. Clavel [et al.] // *Theoretical Computer Science*. – 2002. – Vol. 285, iss. 2. – P. 187–243.
11. Building equational proving tools by reflection in rewriting logic / M. Clavel [et al.] // *CAFE: An Industrial-Strength Algebraic Formal Method*. – Amsterdam, 2000. – P. 1–31.
12. Sebastio, S. MultiVeStA: Statistical model checking for discrete event simulators / S. Sebastio, A. Vandin // *Proc. of the 7th Intern. Conf. on Performance Evaluation Methodologies and Tools, Torino, Italy, 10–12 Dec. 2013*. – Torino, 2013. – P. 310–315.
13. Sen, K. On statistical model checking of stochastic systems / K. Sen, M. Viswanathan, G. Agha // *Lecture Notes in Computer Science*. – 2005. – Vol. 3576. – P. 266–280.
14. The maude formal tool environment / M. Clavel [et al.] // *Lecture Notes in Computer Science*. – 2007. – Vol. 4624. – P. 173–178.
15. Шарыкин, Р. Е. Применение формальных методов при проектировании коллаборационной системы противовирусной защиты / Р. Е. Шарыкин, А. Н. Курбацкий // *Журнал Белорусского государственного университета. Математика. Информатика*. – 2020. – № 1. – P. 59–69.
16. Шарыкин, Р. Е. Апробация модели стохастической коллаборационной защиты от вирусов / Р. Е. Шарыкин // *Системный анализ и прикладная информатика*. – 2021. – № 4. – С. 62–70.
17. Шарыкин, Р. Е. Применение формальных методов при проектировании системы одного окна / Р. Е. Шарыкин // *Журнал Белорусского государственного университета. Математика. Информатика*. – 2021. – № 1. – С. 79–90.
18. Шарыкин, Р. Е. Методика применения формальных методов при проектировании системы закупок предприятия / Р. Е. Шарыкин // *Вестник Гродненского государственного университета имени Янки Купалы. Сер. 2. Математика. Физика. Информатика, вычислительная техника и управление*. – 2022. – Т. 12, № 1. – С. 134–143.

**References**

1. Sammapun U., Sharykin R., DeLap M., Kim M., Zdancewic S. Formalizing Java-MaC. *Electronic Notes in Theoretical Computer Science*, 2003, vol. 89, iss. 2, pp. 171–190.
2. Bernadsky M., Sharykin R., Alur R. Structured modeling of concurrent stochastic hybrid systems. *Lecture Notes in Computer Science*, 2004, vol. 3253, pp. 309–324.
3. Bujorianu M. L., Lygeros J. Toward a general theory of stochastic hybrid systems. *Lecture Notes in Control and Information Science*, 2006, vol. 337, pp. 3–30.

4. Meseguer J. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 1992, vol. 96, iss. 1, pp. 73–155.
5. Marti-Oliet N., Meseguer J. Rewriting logic: roadmap and bibliography. *Theoretical Computer Science*, 2002, vol. 285, iss. 2, pp. 121–154.
6. Agha G. A., Meseguer J., Sen K. PMAude: Rewrite-based specification language for probabilistic object systems. *Electronic Notes in Theoretical Computer Science*, 2006, vol. 153, iss. 2, no. 2, pp. 213–239.
7. Kumar N., Sen K., Meseguer J., Agha G. A rewriting based model for probabilistic distributed object systems. *Lecture Notes in Computer Science*, 2003, vol. 2884, pp. 32–46.
8. Sharykin R. E., Kourbatski A. N. *A model of distributed object-based stochastic hybrid systems*. Zhurnal Belorusskogo gosudarstvennogo universiteta. Matematika. Informatika [Journal of the Belarusian State University. Mathematics. Informatics], 2019, no. 2, pp. 52–61 (In Russ.).
9. Sharykin R. E., Kourbatski A. N. *Verification of distributed object-oriented stochastic hybrid systems*. Vestnik Grodnenskogo gosudarstvennogo universiteta imeni Yanki Kupaly. Seriya 2. Matematika. Fizika. Informatika, vychislitel'naya tekhnika i upravlenie [Bulletin of Grodno State University named after Yanka Kupala. Series 2. Mathematics. Physics. Informatics, Computer Technology and Management], 2019, vol. 9, no. 2, pp. 123–133 (In Russ.).
10. Clavel M., Duran F., Eker S., Lincoln P. D. Maude: Specification and programming in rewriting logic. *Theoretical Computer Science*, 2002, vol. 285, iss. 2, pp. 187–243.
11. Clavel M., Duran F., Eker S., Meseguer J. Building equational proving tools by reflection in rewriting logic. *CAFE: An Industrial-Strength Algebraic Formal Method*. Amsterdam, 2000, pp. 1–31.
12. Sebastio S., Vandin A. MultiVeStA: Statistical model checking for discrete event simulators. *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools, Torino, Italy, 10–12 December 2013*. Torino, 2013, pp. 310–315.
13. Sen K., Viswanathan M., Agha G. On statistical model checking of stochastic systems. *Lecture Notes in Computer Science*, 2005, vol. 3576, pp. 266–280.
14. Clavel M., Duran F., Hendrix J., Lucas S., Meseguer J., Olveczky P. The maude formal tool environment. *Lecture Notes in Computer Science*, 2007, vol. 4624, pp. 173–178.
15. Sharykin R. E., Kourbatski A. N. *Application of formal methods in the design of a collaborative virus defense system*. Zhurnal Belorusskogo gosudarstvennogo universiteta. Matematika. Informatika [Journal of the Belarusian State University. Mathematics. Informatics], 2020, no. 1, pp. 59–69 (In Russ.).
16. Sharykin R. E. *Approbation of the stochastic group virus protection model*. Sistemnyi analiz i prikladnaia informatika [System Analysis and Applied Informatics], 2021, no. 4, pp. 62–70 (In Russ.).
17. Sharykin R. E. *A method of applying format methods in the design of a single window system*. Zhurnal Belorusskogo gosudarstvennogo universiteta. Matematika. Informatika [Journal of the Belarusian State University. Mathematics. Informatics], 2021, no. 1, pp. 79–90 (In Russ.).
18. Sharykin R. E. *A methodology to apply formal methods in the design of an enterprise procurement system*. Vestnik Grodnenskogo gosudarstvennogo universiteta imeni Yanki Kupaly. Seriya 2. Matematika. Fizika. Informatika, vychislitel'naya tekhnika i upravlenie [Bulletin of Grodno State University named after Yanka Kupala. Series 2. Mathematics. Physics. Informatics, Computer Technology and Management], 2022, vol. 12, no. 1, pp. 134–143 (In Russ.).

### Информация об авторе

Шарыкин Роман Евгеньевич, соискатель кафедры технологий программирования, факультет прикладной математики и информатики, Белорусский государственный университет.  
E-mail: sharykin@bsu.edu

### Information about the author

Raman E. Sharykin, Aspirant of the Department of Software Engineering, Faculty of Applied Mathematics and Computer Science, Belarusian State University.  
E-mail: sharykin@bsu.edu