

**ЭВОЛЮЦИЯ КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ
ЧЕРЕЗ АВТОМАТИЗАЦИЮ ПРОЦЕССОВ РАЗРАБОТКИ ПРОТОКОЛОВ**

© 2026 В.Л. Оленев

Санкт-Петербургский государственный университет аэрокосмического приборостроения,
г. Санкт-Петербург, Россия

Статья поступила в редакцию 24.12.2025

Современный процесс проектирования коммуникационных протоколов отличается высокой сложностью и остается сильно неформализованным и эмпирическим процессом. Это приводит к длительным циклам разработки, высокой стоимости и рискам наличия скрытых ошибок в готовой связанной аппаратуре. Существующие подходы не обеспечивают сквозного формализованного процесса от формирования требований до получения верифицированной спецификации, что затрудняет автоматизацию и устранение ошибок. Целью проведенного в статье исследования является формализация и разработка стратегии полной автоматизации сквозной методологии проектирования коммуникационных протоколов для обеспечения надежности, сокращения времени разработки и исключения человеческих ошибок. Предложенная на предыдущих этапах исследования методология, охватывающая все этапы от сбора требований до финализации спецификации, в данной статье представлена при помощи агрегативного подхода (А-схем). Каждый этап методологии представлен в виде агрегата с четко определенными множествами входных/выходных сигналов и состояниями, оператором сопряжения, что создает унифицированную модель всего процесса разработки новых протоколов. Для автоматизации каждого из 13 агрегатов проанализирован и предложен набор формальных методов, которые могут решить конкретную задачу данного этапа, при этом быть в достаточной степени формализованными для достижения автоматизации всего процесса. Подобный подход также дает возможность сделать модульную и четкую спецификацию интерфейсов между этапами. Предложенный подход позволяет трансформировать проектирование протоколов в строгий, управляемый и автоматизированный инженерный процесс. Это приведет к сокращению времени выхода на рынок новой связанной аппаратуры, значительному снижению стоимости разработки и существенному повышению надежности протоколов и стандартов за счет раннего обнаружения ошибок. Результаты работы закладывают фундамент для создания конкретных инструментов САПР, быстрой разработки семейств протоколов и ускоренной эволюции коммуникационных технологий.

Ключевые слова: проектирование протоколов, автоматизация, коммуникационные технологии, агрегативные схемы, А-схемы.

DOI: 10.37313/1990-5378-2026-28-2-185-195

EDN: CWBOPE

Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации, соглашение № FSRF-2023-0003, «Фундаментальные основы построения помехозащищенных систем космической и спутниковой связи, относительной навигации, технического зрения и аэрокосмического мониторинга».

ВВЕДЕНИЕ

Современное состояние проектирования коммуникационных протоколов характеризуется растущей сложностью и высокими требованиями к надёжности, особенно в таких областях, как космические системы, мобильная связь и интернет вещей. Используемые протоколы передачи данных должны обеспечивать не только высокую скорость обмена информацией, но и устойчивость к ошибкам, низкую задержку, энергоэффективность и совместимость с существующими стандартами. Однако сам процесс разработки этих протоколов остаётся в значительной степени неформализованным, фрагментарным и зависимым от опыта конкретных рабочих групп.

Как отмечается в анализе существующих подходов, проведенном в [1], многие организации используют эмпирические методы разработки, основанные на периодических встречах, обсуждениях и инициативной отработке принятых решений на моделях. Это приводит к длительным циклам разработки, высокой стоимости и рискам наличия скрытых ошибок в спецификациях, которые обнаруживаются лишь на этапе аппаратной реализации и эксплуатации. Существующие формальные

методы охватывают лишь отдельные этапы разработки, но не обеспечивают сквозного формализованного процесса от сбора требований до получения проверенной спецификации. Более того, отсутствие единой методологии затрудняет автоматизацию, параллельную организацию работ и системное управление ошибками. Это особенно критично в условиях сжатых сроков и необходимости разработки протоколов для специализированных применений, где ошибка может привести к катастрофическим последствиям.

Таким образом, назрела острая необходимость в новой методологии, которая бы формализовала и автоматизировала весь цикл разработки коммуникационных протоколов - от формирования технических требований до создания верифицированных спецификаций и прототипов. Именно такая методология проектирования коммуникационных протоколов представлена в [2]. Её ключевые преимущества — полнота охвата процесса, возможность параллельной разработки поуровневых и сетевых моделей, встроенный механизм обнаружения и исправления ошибок на ранних стадиях.

ПРОЕКТИРОВАНИЕ КОММУНИКАЦИОННЫХ ПРОТОКОЛОВ

Разработанная методология проектирования коммуникационных протоколов представляет собой комплексный, итеративный процесс, охватывающий полный цикл создания протокола, соответствующего актуальным технологическим запросам и требованиям индустрии. Схема данного процесса представлена на рисунке 1.

Начальной фазой методологии (этап 1) является систематизация и анализ технических требований к проектируемому протоколу. Требования структурируются по двум ключевым категориям. Первая категория включает требования к внутренней архитектуре и логике протокола, таким как механизмы обеспечения качества обслуживания (QoS), структура кадров и пакетов, алгоритмы управления потоком и ошибками. Вторая категория определяет эксплуатационные требования, предъявляемые к конечным устройствам, реализующим данный протокол, в контексте их работы в составе коммуникационных сетей. Сюда входят параметры производительности: коммутационные характеристики, задержки передачи, джиттер, пропускная способность, устойчивость к сбоям и совместимость с существующей инфраструктурой. Итогом этого этапа является четко сформулированный набор критериев, которым должен соответствовать будущий протокол.

Следующим шагом (этап 2) является трансляция собранных требований в формализованный документ - концептуальную версию спецификации протокола. Этот документ детально описывает логику работы, форматы сообщений, состояния системы, последовательности работы протокола и межуровневые интерфейсы взаимодействия.

Учитывая высокую сложность современных протоколов, созданная спецификация не может быть отдана в реализацию без тщательной валидации. Для этой цели методология предусматривает этап программного имитационного моделирования (этапы 3 и 4). На этапе 3 на базе документа спецификации создаются две взаимодополняющие, но структурно различные модели: поуровневая модель (этап 3.1) и сетевая модель (этап 3.2). Поуровневая модель фокусируется на детальной проверке корректности работы каждого отдельного логического уровня протокола (например, канального, сетевого, транспортного), изолируя его механизмы от влияния других уровней и среды. Сетевая модель, в свою очередь, создается только после успешной верификации как минимум одного варианта поуровневой модели. Ее цель - исследовать поведение протокола в условиях, приближенных к реальным: при взаимодействии множества узлов в топологии сети, при наличии больших объемов трафика, потерь и задержек. Таким образом, поуровневая модель отвечает за корректность алгоритмов, а сетевая - за их эффективность и устойчивость в сети.

После создания и настройки моделей (конфигурация параметров, топологии, нагрузок) следует этап 4 - проведение серии имитационных экспериментов. Процесс моделирования для поуровневой (этап 4.1) и сетевой (этап 4.2) моделей осуществляется параллельно для оптимизации времени разработки. Критически важным элементом методологии является механизм управления ошибками. Если в ходе моделирования на любой из параллельных веток (4.1 или 4.2) выявляется несоответствие поведения протокола ожиданиям, составляется детальный отчет об ошибке. Данный отчет направляется на анализ экспертной группе, а работы в соответствующей ветке моделирования приостанавливаются. Происходит возврат к этапу 2 для пересмотра и уточнения спецификации протокола на основе выявленного недочета. После корректировки документ вновь поступает на моделирование. Такой итеративный цикл «спецификация - моделирование - верификация - доработка» продолжается до тех пор, пока обе команды моделирования не подтвердят отсутствие критических ошибок в поведении протокола.

Стадия проверки (этап 5) заключается в сопоставлении итоговых результатов моделирования с исходными техническими требованиями, собранными на этапе 1. Оценивается, удовлетворяет ли поведение смоделированного протокола всем заявленным критериям производительности, надежности и функциональности. В случае выявления несоответствия спецификация вновь отправляется на

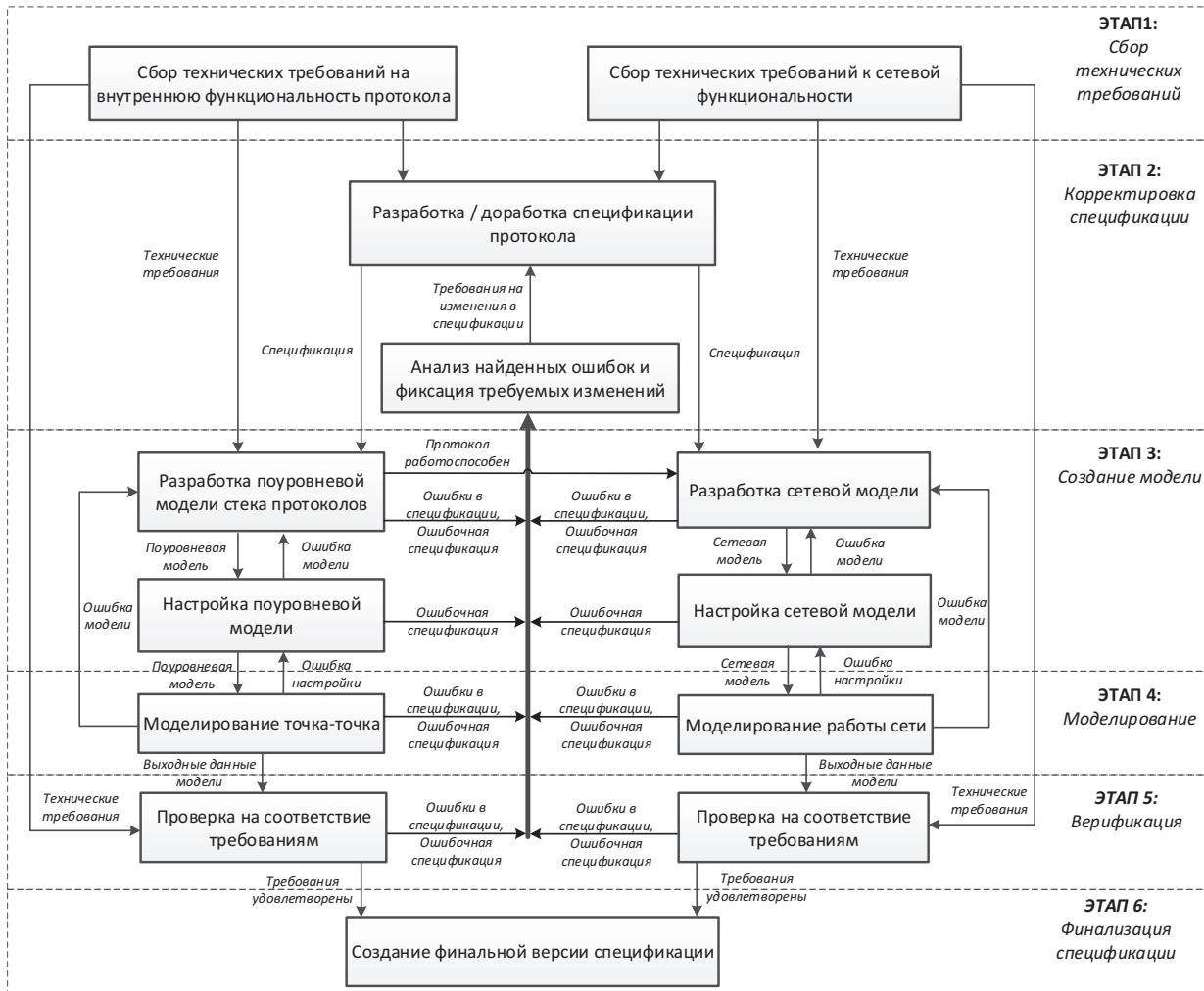


Рис. 1. Структура методологии разработки коммуникационных протоколов

доработку (возврат на этап 2). Если же все требования выполнены, спецификация считается проверифицированной и проходит процедуру финализации (этап 6). Полученный документ является базовым для последующих фаз проекта, выходящих за рамки данной методологии, таких как разработка аппаратного обеспечения, написание эталонного программного референс-кода или стандартизация.

Два параллельных процесса в представленной структуре отображают одновременную разработку поуровневой и сетевой моделей. Важным аспектом функционирования являются документы, передаваемые с этапа на этап, а также динамика их движения между командами, отвечающими за реализацию каждого из этапов.

Предложенная методология представляет собой строгий процесс разработки протоколов. Однако её истинный потенциал и практическая эффективность могут быть полностью раскрыты только через комплексную автоматизацию. Автоматизация необходима по нескольким причинам:

Современные протоколы — это многоуровневые системы с сотнями состояний и тысячами взаимодействий. Вручную отслеживать циркуляцию документов (технических требований, спецификаций, моделей, отчётов об ошибках) между параллельными ветками разработки (поуровневой и сетевой моделью) без ошибок практически невозможно. Автоматизированная система гарантирует, что ни одно изменение спецификации не будет потеряно, а все необходимые проверки и синхронизации между командами выполняются корректно и вовремя.

Процесс работы методологии итеративен, поскольку при обнаружении ошибки в модели или спецификации параллельная ветка моделирования останавливается, далее корректируется спецификация и моделирование запускается повторно. Ручная синхронизация команд и согласование работ занимает дни или недели, а автоматизация позволяет сократить ее до часов или минут. Инструмент мог бы автоматически генерировать каркасы моделей на основе обновлённой спецификации, запускать симуляции и анализировать результаты по заданным критериям, формировать стандартизированные отчёты об ошибках и направлять их нужным экспертам, возобновлять работу параллельных процессов после.

На практике, при жестких сроках проектов, команды могут начинать разработку сетевой модели до готовности поуровневой, что приведет к проблемам. Автоматизированная система может выступить в роли диспетчера, который не позволит начать следующий этап, не выполнив предусловия.

Автоматизация стирает грань между формальным описанием процесса и его реальным выполнением. Разработанная и верифицированная спецификация протокола, представленная в машиночитаемом формате внутри системы, может стать основой для автоматической генерации кода программных моделей (на SystemC, SDL, Python), кода аппаратных описаний (VHDL, Verilog) для прототипирования, тестовых сценариев и тестов для верификации реализаций. Это резко снижает вероятность ошибок при «ручном» переводе спецификации в код и многократно увеличивает скорость перехода от проектирования к реализации.

Автоматизированная среда проектирования становится хранилищем формализованных требований, шаблонов моделей, проверенных блоков протоколов и сценариев тестирования. Это позволяет не начинать каждый новый проект с чистого листа, а комбинировать и адаптировать уже существующие, верифицированные компоненты, что является ключом к быстрой разработке семейств протоколов.

Разработанная методология предоставляет теоретический фундамент для корректного и полного процесса проектирования. Без автоматизации методология может остаться сложным, но редко применяемым подходом, при этом с автоматизацией процесса она становится неким конвейером для создания безошибочных протоколов, который крайне востребован в эпоху, когда время выхода на рынок и надёжность коммуникаций являются критическими конкурентными преимуществами. Реализация такой системы — следующий логический и необходимый шаг для трансформации проектирования коммуникационных протоколов в строгую автоматизированную инженерную последовательность задач.

АГРЕГАТИВНАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ МЕТОДОЛОГИИ ПРОЕКТИРОВАНИЯ КОММУНИКАЦИОННЫХ ПРОТОКОЛОВ

Для полной автоматизации разработанной методологии необходимо создать ее представление в виде некоторой формализовано взаимосвязанных компонентов. Более того, необходимо провести анализ, может ли быть автоматизирован каждый из компонентов методологии в отдельности. Однако, представленная методология представляет собой совокупность различных действий, каждое из которых может быть представлено как отдельная сложная формализованная система. Широко известность в качестве универсального метода для формального описания функционирования таких систем получил подход, разработанный Н. П. Бусленко [3]. Агрегативной системой, или Асхемой, называют категорию сложных систем, которые из-за своей высокой структурной сложности не могут быть представлены в виде единой математической модели, а только как комбинация из отдельных элементов, описанных различными формальными техниками.

В случаях, когда аналитические методы оказываются недостаточно результативными или вообще неприменимыми, используется метод имитационного моделирования. При этом описание моделируемого объекта в терминах агрегативной схемы (А-схемы) служит базой для последующего создания самой имитационной модели, а также для разработки её математического аппарата: архитектуры системы, внутренней логики работы, интерфейсов. Использование А-схемы в качестве формы представления исследуемой системы обеспечивает значительную унификацию системы. Унификация касается не только алгоритмов, лежащих в основе имитационного процесса, и процедур автоматизации моделирования, но и распространяется на последующие этапы работы. Такой подход открывает возможность применения типовых, хорошо отработанных методов для обработки данных, получаемых в ходе эксперимента, и для последующего анализа результатов моделирования сложной системы.

Агрегативные системы применяются для унификации описания и возможности анализа поведения сложных систем. Они позволяют формально описать структуру и логику работы системы, находить узкие места, тестировать различные конфигурации и управляющие воздействия до этапа реализации. Формализация агрегативных схем - это формальное описание динамической структуры взаимодействия её компонентов во времени, представленное в виде, максимально приближённом к алгоритму компьютерной реализации, то есть один из наиболее мощных и адекватных формализмов для автоматизации технических процессов.

Чтобы описать реальную систему при помощи А-схемы, нужно отдельно описать каждый из агрегатов (A_n) и связи между ними. Таким образом, для представления методологии проектирования коммуникационных протоколов как агрегативной системы [4], каждый процесс, обозначенный на рисунке 1, должен быть представлен как отдельный агрегат. Каждый из этих элементов характеризуется следующими множествами:

- моменты времени T ,
- входные сигналы X ,
- выходные сигналы Y ,
- внутренние параметры системы H ,
- состояния системы Z в конкретные моменты времени $t \in T$.

Также возможно описать множества внешних сигналов управления агрегатами G . Однако, разработанная методология не подразумевает отдельное управление каждым из элементов, поэтому управляющих сигналов для агрегатов A -схемы не будет.

Работа агрегативной схемы представляет собой процесс непрерывной обработки информации, которая разделяется на два основных типа: внешнюю и внутреннюю. Внешняя информация поступает в систему из окружающей её среды от взаимодействующих с ней объектов. Внутренняя же информация формируется непосредственно агрегатами в ходе их функционирования. Основным механизмом, обеспечивающим как взаимодействие A -схемы с внешней средой, так и связь между отдельными агрегатами внутри системы, является обмен специальными сообщениями - сигналами. В рамках методологии фундаментальной единицей такого обмена выступают элементарные сигналы, означающие передачу конкретных документов. Их передача осуществляется мгновенно и независимо друг от друга по специально выделенным каналам связи, отображаемым на схеме. Архитектура связей в A -схеме подчиняется строгим структурным правилам, обеспечивающим детерминированность и однозначность информационных потоков. Так, каждый входной контакт любого элемента системы может быть соединён не более чем с одним элементарным каналом, что исключает конфликты при получении данных. В то же время выходной контакт агрегата способен выдавать сигнал одновременно по любому конечному числу каналов. Критически важным дополнительным условием является правило, согласно которому множественные каналы, исходящие от различных источников, не могут быть направлены на один и тот же входной контакт другого элемента; каждый вход получает данные только из одного строго определённого источника. Это гарантирует отсутствие коллизий и чёткую прослеживаемость путей прохождения информации в сложной системе.

Переходы агрегата A_i из состояния $z_i(t_1)$ в $z_i(t_2)$ определяются внутренними параметрами $h_i(t) \in H$ и входными сигналами $x_i(t) \in X$. Но поскольку в данном случае интерес представляет только структурная составляющая A -схемы, без учета динамики функционирования, внутренние параметры, влияющие на совершение переходов в пространстве состояний, можно не рассматривать.

Взаимодействие элементов A -схемы и внешней среды представляется как обмен сигналами между агрегатами A_n и A_o . То есть внешняя среда представляется в виде отдельного агрегата системы (A_o), который, так же, как и остальные агрегаты, содержит входные и выходные контакты.

Каждый агрегат A_n формально описывается двумя множествами, использующимися для описания их сопряжения с прочими элементами A -схемы и внешней средой:

- входные контакты $X_i(n) = \{X_1(n), X_2(n), \dots, X_b(n)\}$,
- выходные контакты $Y_j(n) = \{Y_1(n), Y_2(n), \dots, Y_c(n)\}$.

Для описания сопряжения элементов в A -схеме используется оператор R в совокупности с множествами $X_i(n)$ и $Y_j(n)$. Этот оператор сопряжения R задается в виде таблицы, в которой на пересечении строк с номерами элементов (агрегатов) n и столбцов с номерами контактов i располагаются пары чисел, указывающие номер элемента и номер контакта, с которым соединен контакт $X_i(n)$.

Поставим в соответствие каждому из элементов методологии проектирования коммуникационных протоколов конкретный агрегат A -схемы. Соотношение агрегатов и названий этапов методологии представлено в таблице 1.

На рисунке 2 представлена A -схема, отображающая процесс проектирования, соответствующий методологии проектирования коммуникационных протоколов.

Оператор сопряжения R представлен в таблице 2.

Сами по себе агрегаты составленной схемы могут быть реализованы при помощи разных математических схем и формальных методов. При этом главное – соблюдать условия, указанные в операторе сопряжения и наборы сигналов, которые должны поступать на вход и выход каждого агрегата. Таким образом, межмодульные интерфейсы полностью специфицированы: четко известно, какая информация передается на вход и выход (множества X и Y), а также через какой выходной порт и на какой входной порт какого агрегата (по оператору сопряжения R).

ПОДХОД К АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ ПРОТОКОЛОВ НА БАЗЕ АГРЕГАТИВНОЙ МОДЕЛИ

Сбор технических требований на внутреннюю функциональность протокола и технических требований к сетевой функциональности (агрегаты A_1 и A_2) может быть реализован при помощи различных методик сбора и формирования требований [5, 6, 7] и статистических методов анализа [8].

Таблица 1. Соотношение агрегатов и названий этапов методологии

Агрегат	Название этапа в методологии
A_0	Внешняя среда (инициатор создания протокола)
A_1	Сбор технических требований на внутреннюю функциональность протокола
A_2	Сбор технических требований к сетевой функциональности
A_3	Разработка / доработка спецификации протокола
A_4	Разработка поуровневой модели стека протоколов
A_5	Разработка сетевой модели стека протоколов
A_6	Настройка поуровневой модели
A_7	Настройка сетевой модели
A_8	Моделирование точка-точка
A_9	Моделирование работы сети
A_{10}	Проверка на соответствие требованиям внутренней функциональности
A_{11}	Проверка на соответствие требованиям к сетевой функциональности
A_{12}	Анализ найденных ошибок и фиксация требуемых изменений
A_{13}	Финализация спецификации

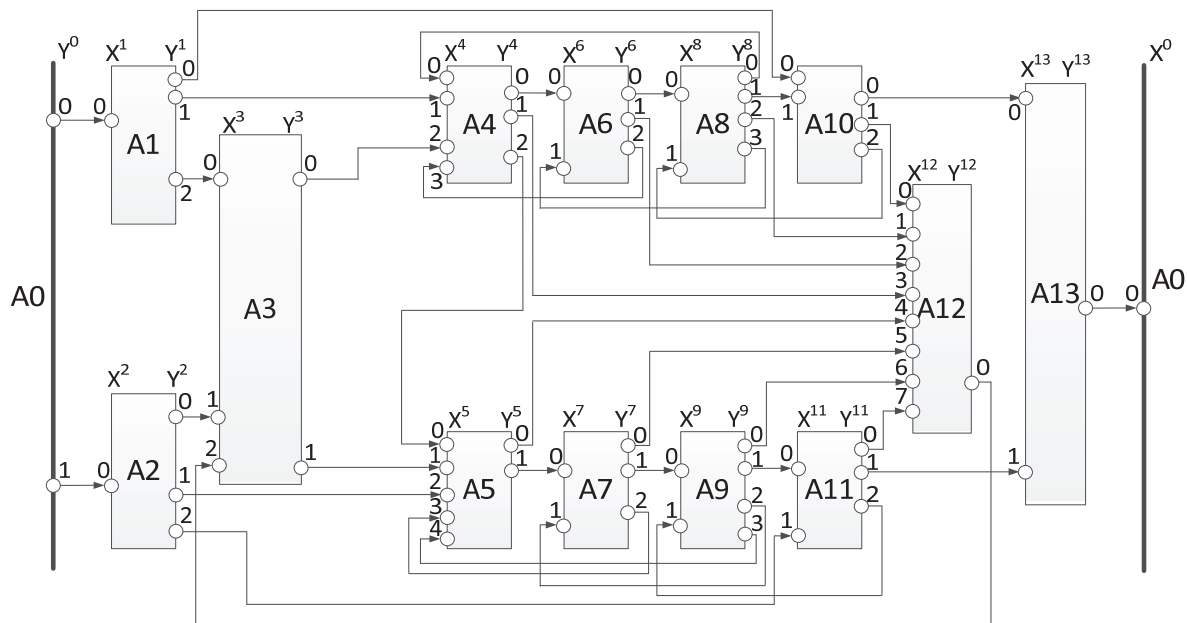


Рис. 2. А-схема процесса проектирования коммуникационных протоколов

При этом требования сразу могут быть записаны как формулы в терминах темпоральных логик. В этом случае будет применен разработанные методы формальной верификации (*model checking*). Это автоматизированная проверка математической модели системы (часто в виде графа или автомата) на соответствие формальным требованиям (спецификациям) [9, 10]. Проведение этой операции позволит математически доказать отсутствие ошибок. Использование математического подхода темпоральными логиками превращает сбор требований в полностью автоматическую процедуру, сделать требования однозначными (формализация), выявить противоречия (логический анализ), доказать выполнимость требований (верификация) еще на ранней стадии проекта.

Следующий этап после сбора требований – это автоматическое написание технических спецификаций (агрегат A_3). По сути, если технические требования формализованы в виде темпоральных логик, это трансляция информации из формального, машиночитаемого представления (моде-

Таблица 2. Оператор сопряжения R

Номер агрегата	Номер входного контакта							
	0	1	2	3	4	5	6	7
0	13,0	-	-	-	-	-	-	-
1	0,0	-	-	-	-	-	-	-
2	0,1	-	-	-	-	-	-	-
3	1,2	2,0	12,0	-	-	-	-	-
4	8,0	1,1	3,0	6,2	-	-	-	-
5	4,2	3,1	2,1	7,2	9,3	-	-	-
6	4,0	8,3	-	-	-	-	-	-
7	5,1	9,3	-	-	-	-	-	-
8	6,0	10,2	-	-	-	-	-	-
9	7,1	11,2	-	-	-	-	-	-
10	1,0	8,1	-	-	-	-	-	-
11	9,1	2,2	-	-	-	-	-	-
12	10,1	8,2	6,1	4,1	5,0	7,0	9,0	11,0
13	10,0	11,1	-	-	-	-	-	-

ли, графы, логические формулы) в привычный технический документ (спецификацию) по строгим правилам [11, 12]. Это также позволит избавиться от рутинного, подверженного ошибкам переписывания одних и тех же данных в разных документах, обеспечивая при этом безошибочность и лаконичность итоговой спецификации. После этого использование алгоритмов автоматического извлечения формальных требований и ключевых понятий из текстов, которое состоит из распознавания именованных сущностей (поиска в тексте упоминания компонентов, параметров, стандартов), далее семантический анализ и извлечение отношений (определение связей между сущностями), а также генерация связанных текстовых описаний для разделов спецификации. Это часто делается с помощью шаблонов, заполняемых данными из модели.

Задача этапов, связанных с агрегатами A_4 и A_5 – автоматическое создание программной модели на базе спецификации. Подходы к созданию поуровневой модели (агрегат A_5), когда в наличии уже есть формализованная спецификация и технические требования в значительной степени проработаны. Например, методы перехода от спецификаций протоколов передачи данных к архитектурным диаграммам программных моделей, затем формальное описание раскрашенными сетями Петри и верификация архитектурной диаграммы программной модели [13]. Кроме того, при разработке поуровневой модели могут быть использованы формальные техники моделирования, такие как элементы теории автоматов с генерацией кода в язык SDL [14].

Разработка сетевой модели стека протоколов (агрегат A_4) – это сложная многофакторная задача, которая должна учитывать не только принципы, описанные в спецификации протокола, но и различные физические свойства сетей, принципы их организации и т.п. Здесь необходимы дополнительные методы и алгоритмы, которые автоматизируют процесс составления сети из компонентов (узлы, коммутаторы, каналы), при необходимости дорабатывают до требуемой отказоустойчивости, качественно прокладывают маршруты передачи данных, настраивают параметры обеспечения качества сервиса, а также дают возможность запустить на моделирование и собрать статистику для анализа. Совокупности таких методов и алгоритмов представлены в публикациях [15, 16, 17]. Подобные подходы применены и отработаны в составе САПР SANDS.

Для автоматической настройки (агрегаты A_6 и A_7) моделей создано несколько подходов. Выбор метода зависит от размерности пространства параметров, сложности реализации оценки, наличия ограничений на применение, а также характера параметров (непрерывные, дискретные, стохастические и т.п.). Например, на первоначальных этапах настройка моделей может проводиться вручную, а далее могут быть использованы цифровые двойники. Цифровой двойник настраивает параметры не как одноразовую задачу, а как непрерывный процесс адаптивного оценивания. Цель его использования здесь – наиболее точное отражение поведения и свойств физической системы-оригинала на протяжении всего ее жизненного цикла, что позволяет делать надежные прогнозы и оптимальные управляющие воздействия. При этом цифровой двойник не настроит модель однократно, он будет делать это непрерывно или периодически, подстраиваясь под изменения реального объекта. Именно поэтому применение цифрового двойника для настройки модели выглядит

максимально перспективным методом. Однако, для проектирования протоколов возможно использовать и более простые подходы, не использующие концепции искусственного интеллекта [18, 19].

Под моделированием точка-точка и моделированием работы сети (агрегаты A_8 и A_9) подразумевается именно процесс симуляции, то есть запуск на исполнение моделей, разработанных и настроенных на предыдущих этапах. Здесь используются формальные методы и алгоритмы, позволяющие ускорить процесс работы программного кода, методы совместного моделирования [20, 21].

Проверка на соответствие требованиям (агрегаты A_{10} и A_{11}) может осуществляться при помощи формальных методов верификации, таких как *model checking* [22]. Агрегаты A_8 - A_{11} , по сути, отражают процесс верификации. Если модель на предыдущих этапах уже была представлена как сеть Петри или как граф, то есть возможность преобразовать ее в модель Крипке [23], а затем применить методики формальной верификации и проверить спецификации, записанные формулами темпоральной логики, на полученных моделях Крипке.

Финализация спецификации (агрегат A_{13}) подразумевает работу экспертов и приведение самого стандарта, а также используемой терминологии в соответствии с требованиями ГОСТ, что полностью автоматически выполняется при помощи нейронных сетей.

Таким образом, процесс, представленный А-схемой на рисунке 2, может быть проиллюстрирован следующей схемой, отображенной на рисунке 3.

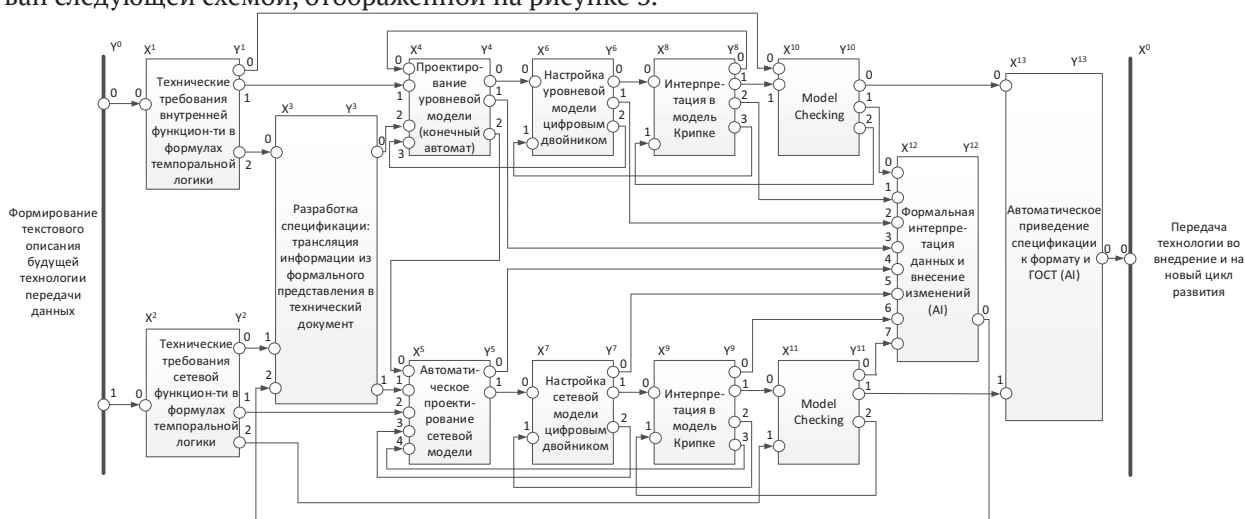


Рис. 3. Автоматизированный процесс эволюции коммуникационных технологий

ЗАКЛЮЧЕНИЕ

Статья посвящена важной для индустрии связи проблеме: автоматизации проектирования коммуникационных технологий, а также четкой структуризации этого процесса. Предложено решение в виде формального агрегативного описания разработанной в более ранних работах автора методологии проектирования коммуникационных протоколов, а также описана стратегия ее полной автоматизации.

Описанная методология проектирования коммуникационных протоколов представляет собой инженерный процесс, основные достоинства которого заключаются в четком распределении этапов, возможности параллельной организации проектирования и моделирования, управление процессом исправления ошибок, четкие и понятные критерии завершения процесса.

В данной работе представлена формализация методологии в виде агрегативной системы (А-схемы). Подобное представление унифицирует сложный, многоэтапный процесс, который теперь представлен в виде формальной конструкции из взаимодействующих агрегатов, каждый из которых соответствует конкретному этапу методологии. Понятным образом специфицированы интерфейсы через множества входных/выходных сигналов и оператор сопряжения. При этом А-схема позволяет рассматривать каждый этап (агрегат) как относительно независимый модуль, каждый агрегат может быть реализован наиболее подходящим формальным методом или инструментом, наполнен самостоятельной формальной методикой, позволяющей автоматизировать эти процессы.

Формализованный автоматизированный процесс может выглядеть следующим образом:

- Автоматизация формирования технических требований с использованием темпоральных логик, что позволяет в дальнейшем применять методы *model checking*.
- Генерация спецификаций: трансляция формальных моделей в текст и извлечение сущностей и отношений из текстовых требований.

- Генерация моделей: автоматическое создание поуровневых моделей (через применение сетей Петри, *SDL*) и сетевых моделей с использованием специфических алгоритмов, применяемым в САПР.
- Настройка и моделирование: использование концепции цифровых двойников для адаптивной настройки, а также методов ускоренного и совместного моделирования.
- Верификация: Прямое применение формальных методов верификации – проверка моделей Крипке на соответствие формулам темпоральной логики).
- Финализация спецификации: использование нейросетевых инструментов для приведения документации в соответствие со стандартами.

Таким образом, представлена целостная концепция трансформации проектирования коммуникационных протоколов. В дальнейшем это позволит разработать прототип автоматизированной системы на основе предложенной *A*-схемы, апробировать методологию на реальных проектах по созданию протоколов для критических промышленных задач, создать библиотеки верифицированных компонентов и шаблонов. Использование такого подхода позволит автоматизировать процесс эволюции современных коммуникационных технологий за счет передачи готового стандарта на вход полученной агрегативной схемы вместе с улучшенными техническими требованиями, обеспечив следующий этап развития технологии. Следствием этого станет ускоренная разработка и развитие коммуникационных технологий для наземных, воздушных и космических коммуникаций, бортового обмена данными, а также реализации концепции и дальнейшей эволюции беспроводного цифрового неба.

СПИСОК ЛИТЕРАТУРЫ

1. Оленев, В.Л. Методология формализованного проектирования коммуникационных протоколов на основе сетей Петри / В.Л. Оленев // Информатика и космос. – 2022. – № 4(2). – С. 37-45.
2. Оленев, В.Л. Проектирование и моделирование коммуникационных систем: монография / В.Л. Оленев, Е.А. Суворова, Н.И. Синёв. – СПб.: ГУАП, 2024. – 328 с.
3. Бусленко, Н.П. Моделирование сложных систем / Н.П. Бусленко. – М.: Наука, 1978. – 402 с.
4. Советов, Б. Я. Моделирование систем: Учеб. для вузов / Б. Я. Советов, С. А. Яковлев. – М.: Высш. шк., 2001. – 343 с.
5. Даниярулы, Ж. Особенности разработки требований к программному обеспечению / Ж. Даниярулы, М. Рыспаева, М. Кажиманова, Ш. Токсынов // Science and technology research: сборник статей II Международной научно-практической конференции. – Петрозаводск, 2021. – С. 47-56.
6. Аксенов К.А., Аксенова О.П., Гончарова Н.В. [и др.]. Свидетельство о государственной регистрации программы для ЭВМ № 2018612978 Российская Федерация. Система генерации технического задания на программное обеспечение (Авто ТЗ) : № 2018610328. 2018.
7. Шевцова, О.С. Проблема выбора методов сбора требований и обоснование их использования на этапах разработки автоматизированных информационных систем / О.С. Шевцова, Л.А. Юнина, А.С. Шевцов, Л.З. Давлеткиреева // Молодежная наука как фактор и ресурс инновационного развития: сборник статей Международной научно-практической конференции. – Петрозаводск, 2020. – С. 19-23.
8. Шорохова, И.С. Статистические методы анализа: учеб. пособие / И.С. Шорохова, Н.В. Кисляк, О.С. Мариев. М-во образования и науки Рос. Федерации, Урал. федер. ун-т. – Екатеринбург: Изд-во Урал. ун-та, 2015. – 300 с.
9. Сергеева, А.С. Формальная верификация механизма транспортных соединений протокола STP-ISS-14 / А.С. Сергеева, Н.Ю. Чумакова // Системный анализ и логистика. – 2023. – № 2(36). – С. 54-63.
10. Чумакова, Н.Ю. Формальная верификация качества сервиса транспортных протоколов / Н.Ю. Чумакова // Совместная III Отраслевая научно-практическая конференция «Созвездие Роскосмоса: траектория науки» и VIII Всероссийская научно-техническая конференция «Актуальные проблемы ракетно-космической техники» («VIII Козловские чтения») : Материалы Совместной III Отраслевой научно-практической конференции и VIII Всероссийской научно-технической конференции. – Самара: Федеральное государственное бюджетное учреждение науки Самарский федеральный исследовательский центр Российской академии наук, 2024. – С. 264-266.
11. Котляров, В.П. Автоматизация формализации требований к программным проектам / Котляров В.П., Петров А.В. // Информатика, телекоммуникации и управление. – 2009. – № 3(80). – С. 236-241.
12. Кулямин, В.В. Формализация требований на практике / В.В. Кулямин, Н.В. Пакулин, О.Л. Петренко, А.А. Сортов, А.В. Хорошилов. – Институт системного программирования РАН. Препринт. – 2006. – № 13. – 69 с. – URL: https://www.ispras.ru/preprints/docs/prep_13_2006.pdf (дата обращения 12.11.2025).
13. Оленев, В.Л. Проектирование программных моделей сетевых протоколов для встроенных систем : дис. ... канд. техн. наук / В.Л. Оленев. – СПб, 2011. – 235 с.
14. Olenev, V. Co-modeling of embedded networks using systemc and SDL: From theory to practice / V. Olenev, A. Rabin, I. Lavrovskaya [et al.] // Advancing Embedded Systems and Real-Time Communications with Emerging Technologies, – 2014. – С. 206-233.
15. Оленев, В.Л. Автоматизированное прокладывание беступиковых маршрутов в сетях с червячной коммутацией / В.Л. Оленев // Научно-технические исследования в космических исследованиях Земли. – 2024. – Т. 16. – № 4. – С. 41-49.
16. Оленев, В.Л. Автоматическое построение отказоустойчивых бортовых сетей / В.Л. Оленев // Научно-технические исследования в космических исследованиях Земли. – 2023. – Т. 15. – № 4. – С. 4-13.

17. Оленев, В.Л. Распределение информационных потоков в сетях с червячной маршрутизацией / В.Л. Оленев, А.А. Карандашев // Научные технологии в космических исследованиях Земли. – 2025. – Т. 17. – № 3. – С. 4-14.
18. Сукманова, Т.В. Цифровой двойник как инструмент моделирования и оптимизации сложных природных и технических систем / Т.В. Сукманова, С.Н. Белов // Вестник Балтийского федерального университета им. И. Канта. Серия: Естественные и медицинские науки. – 2025. – № 1. – С. 40-57.
19. Боровков, А.И. Цифровые двойники в условиях четвертой промышленной революции / А.И. Боровков // Connect WIT. – 2021. – № 01-02. – С. 50-53.
20. Morozkin, P. Integration of SDL Models into a SystemC Project for Network Simulation / P. Morozkin, I. Lavrovskaya, V. Olenev, K. Nedovodeev // SDL 2013, LNCS 7916, Springer-Verlag Berlin Heidelberg. 2013. С. 275-290.
21. Olenev, V. Hierarchical Simulation of Onboard Networks / V. Olenev, I. Lavrovskaya, I. Korobkov, N. Sinyov, Y. Sheynin // Intelligent Distributed Computing XIII. IDC 2019. Studies in Computational Intelligence. V. 868. Springer. 2019. С. 191-196.
22. Карпов, Ю.Г. Model Checking. Верификация параллельных и распределенных программных систем / Ю.Г. Карпов. – СПб.: БХВ-Петербург, 2010. – 560 с.
23. He L., Liu G., Sun Y. PNet: A Petri-Net-Based Model Checker for Computation Tree Logic // 2023 International Conference on Networking, Sensing and Control (ICNSC). Marseille, 2023. С. 1-7.

EVOLUTION OF COMMUNICATION TECHNOLOGIES BY AUTOMATION OF PROTOCOL DEVELOPMENT PROCESSES

© 2026 V.L. Olenev

Saint Petersburg State University of Aerospace Instrumentation, Saint Petersburg, Russia

The modern process of designing communication protocols is highly complex and remains largely informal and empirical. This leads to long development cycles, high costs, and the risk of latent errors in finished communications equipment. Existing approaches do not provide an end-to-end formalized process from requirements generation to verified specifications, complicating automation and error elimination. The goal of this study is to formalize and develop a strategy for fully automating an end-to-end communication protocol design methodology to ensure reliability, reduce development time, and eliminate human error. The methodology proposed in the previous stages of the study, covering all stages from requirements gathering to specification finalization, is presented in this paper using an aggregative approach (A-schemes). Each stage of the methodology is represented as an aggregate with clearly defined sets of input/output signals and states, along with a coupling operator, creating a unified model of the entire process of developing new protocols. To automate each of the 13 aggregates, a set of formal methods was analyzed and proposed. These methods can solve the specific problem of a given stage while being sufficiently formalized to achieve automation of the entire process. This approach also enables a modular and clear specification of interfaces between stages. The proposed approach enables the transformation of protocol design into a rigorous, manageable, and automated engineering process. This will lead to a reduction in the time to market for new communications equipment, a significant reduction in development costs, and a significant increase in the reliability of protocols and standards through early error detection. The results of this work lay the foundation for the creation of specific CAD tools, the rapid development of protocol families, and the accelerated evolution of communications technologies.

Keywords: protocol design, automation, communication technologies, aggregation schemes, A-schemes.

DOI: 10.37313/1990-5378-2026-28-2-185-195

EDN: CWBOPE

REFERENCES

1. Olenev, V.L. Metodologiya formalizovannogo proektirovaniya kommunikacionnyh protokolov na osnove setej Petri / V.L. Olenev // Informaciya i kosmos. – 2022. – № 4(2). – С. 37-45.
2. Olenev, V.L. Proektirovanie i modelirovanie kommunikacionnyh sistem: monografiya / V.L. Olenev, E.A. Suvorova, N.I. Sinyov. – SPb.: GUAP, 2024. – 328 s.
3. Buslenko, N.P. Modelirovanie slozhnyh sistem / N.P. Buslenko. – M.: Nauka, 1978. – 402 s.
4. Sovetov, B. Ya. Modelirovanie sistem: Ucheb. dlya vuzov / B. Ya. Sovetov, S. A. Yakovlev. – M.: Vyssh. shk., 2001. – 343 s.
5. Daniyaruly, Zh. Osobennosti razrabotki trebovanij k programnomu obespecheniyu / Zh. Daniyaruly, M. Ryspaeva, M. Kazhimanova, Sh. Toksynov // Science and technology research: sbornik statej II Mezhdunarodnoj nauchno-prakticheskoy konferencii. – Petrozavodsk, 2021. – С. 47-56.
6. Aksenov K.A., Aksenova O.P., Goncharova N.V. [i dr.]. Svidetel'stvo o gosudarstvennoj registracii programmy dlya EVM № 2018612978 Rossijskaya Federaciya. Sistema generacii tekhnicheskogo zadaniya na programnoe obespechenie (Avto TZ) : № 2018610328. 2018.
7. Shevcova, O.S. Problema vybora metodov sbora trebovanij i obosnovanie ih ispol'zovaniya na etapah razrabotki avtomatizirovannyh informacionnyh sistem / O.S. Shevcova, L.A. Yunina, A.S. Shevcov, L.Z. Davletkireeva //

- Molodezhnaya nauka kak faktor i resurs innovacionnogo razvitiya: sbornik statej Mezhdunarodnoj nauchno-prakticheskoj konferencii. – Petrozavodsk, 2020. – S. 19-23.
8. *Shorohova, I.S.* Statisticheskie metody analiza: ucheb. posobie / I.S. Shorohova, N.V. Kislyak, O.S. Mariev. M-vo obrazovaniya i nauki Ros. Federacii, Ural. feder. un-t. – Ekaterinburg: Izd-vo Ural. un-ta, 2015. – 300 s.
 9. *Sergeeva, A.S.* Formal'naya verifikaciya mekhanizma transportnyh soedinenij protokola STP-ISS-14 / A.S. Sergeeva, N.Yu. Chumakova // Sistemnyj analiz i logistika. – 2023. – № 2(36). – S. 54-63.
 10. *Chumakova, N.Yu.* Formal'naya verifikaciya kachestv servisa transportnyh protokolov / N.Yu. Chumakova // Sovmestnaya III Otrasleyaya nauchno-prakticheskaya konferenciya «Sozvezdie Roskosmosa: traektoriya nauki» i VIII Vserossijskaya nauchno-tehnicheskaya konferenciya «Aktual'nye problemy raketno-kosmicheskoy tekhniki» («VIII Kozlovskie chteniya») : Materialy Sovmestnoj III Otrasleyoy nauchno-prakticheskoj konferencii i VIII Vserossijskoj nauchno-tehnicheskoy konferencii. – Samara: Federal'noe gosudarstvennoe byudzhethoe uchrezhdenie nauki Samarskij federal'nyj issledovatel'skij centr Rossijskoj akademii nauk, 2024. – S. 264-266.
 11. *Kotlyarov, V.P.* Avtomatizaciya formalizacii trebovanij k programmnyh projektam / Kotlyarov V.P., Petrov A.V. // Informatika, telekommunikacii i upravlenie. – 2009. – № 3(80). – S. 236-241.
 12. *Kulyamin, V.V.* Formalizaciya trebovanij na praktike / V.V. Kulyamin, N.V. Pakulin, O.L. Petrenko, A.A. Sortov, A.V. Horoshilov. – Institut sistemnogo programirovaniya RAN. Preprint. – 2006. – № 13. – 69 s. – URL: https://www.ispras.ru/preprints/docs/prep_13_2006.pdf (data obrashcheniya 12.11.2025).
 13. *Olenev, V.L.* Proektirovanie programmnyh modelej setevykh protokolov dlya vstroennykh sistem : dis. ... kand. tekhn. nauk / V.L. Olenev. – SPb, 2011. – 235 s.
 14. *Olenev, V.* Co-modeling of embedded networks using systemc and SDL: From theory to practice / V. Olenev, A. Rabin, I. Lavrovskaya [et al.] // Advancing Embedded Systems and Real-Time Communications with Emerging Technologies, – 2014. – C. 206-233.
 15. *Olenev, V.L.* Avtomatizirovanoe prokladyvanie bestupikovykh marshrutov v setyah s chervyachnoj kommutaciej / V.L. Olenev // Naukoemkie tekhnologii v kosmicheskikh issledovaniyakh Zemli. – 2024. – T. 16. – № 4. – S. 41-49.
 16. *Olenev, V.L.* Avtomaticheskoe postroenie otkazoustojchivykh bortovykh setej / V.L. Olenev // Naukoemkie tekhnologii v kosmicheskikh issledovaniyakh Zemli. – 2023. – T. 15. – № 4. – S. 4-13.
 17. *Olenev, V.L.* Raspredelenie informacionnykh potokov v setyah s chervyachnoj marshrutizaciej / V.L. Olenev, A.A. Karandashev // Naukoemkie tekhnologii v kosmicheskikh issledovaniyakh Zemli. – 2025. – T. 17. – № 3. – S. 4-14.
 18. *Sukmanova, T.V.* Cifrovoy dvojniki kak instrument modelirovaniya i optimizacii slozhnykh prirodnykh i tekhnicheskikh sistem / T.V. Sukmanova, S.N. Belov // Vestnik Baltijskogo federal'nogo universiteta im. I. Kanta. Seriya: Estestvennye i medicinskie nauki. – 2025. – № 1. – S. 40-57.
 19. *Borovkov, A.I.* Cifrovye dvojniki v usloviyakh chetvortoj promyshlennoj revolyucii / A.I. Borovkov // Connect WIT. – 2021. – № 01-02. – S. 50-53.
 20. *Morozkin, P.* Integration of SDL Models into a SystemC Project for Network Simulation / P. Morozkin, I. Lavrovskaya, V. Olenev, K. Nedovodeev // SDL 2013, LNCS 7916, Springer-Verlag Berlin Heidelberg. 2013. C. 275-290.
 21. *Olenev, V.* Hierarchical Simulation of Onboard Networks / V. Olenev, I. Lavrovskaya, I. Korobkov, N. Sinyov, Y. Sheynin // Intelligent Distributed Computing XIII. IDC 2019. Studies in Computational Intelligence. V. 868. Springer. 2019. C. 191-196.
 22. *Karpov, Yu.G.* Model Checking. Verifikaciya paralel'nykh i raspredelennykh programmnykh sistem / Yu.G. Karpov. – SPb.: BHV-Peterburg, 2010. – 560 s.
 23. *He L., Liu G., Sun Y.* PNet: A Petri-Net-Based Model Checker for Computation Tree Logic // 2023 International Conference on Networking, Sensing and Control (ICNSC). Marseille, 2023. C. 1-7.