

Анализ инвариантных последовательностей процессов на основе вероятностно-взвешенных графов и символьных вычислений

О. А. Жаркова • Р. В. Насыров

Уфимский университет науки и технологий

В данной статье рассматривается проблема расширения возможностей структурного анализа процессов, возникающая при использовании традиционных графических нотаций (UML, BPMN) в сторону формирования количественных оценок, характеризующих выявленные процессы. Авторы предлагают методику преобразования статических диаграмм указанных нотаций в графы, нагруженные вероятностными и ресурсными величинами как в символьных обозначениях с последующей символьной обработкой, так и в численной форме с последующим получением величин оценок качества процесса. Предлагаемый подход позволяет перейти от описательного моделирования к численной оценке эффективности процессов. Предложена архитектура адаптивного цикла анализа, включающая вычислительный и валидационный контуры, реализующие итеративное уточнение модели. Предлагаемая методика реализована в виде микросервисного веб-приложения на базе клиент-серверной архитектуры с использованием FastAPI, Python и библиотеки Vis.js, обеспечивающего визуальное построение графов, формирование матриц вероятностей и весов, а также получение аналитических выражений в обратной польской нотации. На примере типового процесса оформления онлайн-заказа в интернет-магазине продемонстрирована возможность выявления узких мест и количественной оценки необходимого уровня оптимизации параметров. Предложенный подход позволяет интегрировать различные методы визуального структурного моделирования с символьными и количественными вычислениями, предоставляя инструмент для обоснованного принятия решений на этапе проектирования информационных систем.

Микросервисы; FastAPI; Python; UML; BPMN; вычисления на графах; сервис.

ВВЕДЕНИЕ

При проектировании информационных систем и бизнес-процессов широкое распространение получили графические нотации — IDEF0, UML (диаграммы последовательности, активности), BPMN и другие. С их помощью проектировщики, проводя анализа контекста задачи, фиксируют инвариантные последовательности операций, то есть устойчивые, повторяющиеся структуры, из которых складывается деятельность [Мир25, Сем25]. На практике использование таких моделей зачастую ограничивается качественной верификацией: проверяется полнота охвата функций, логическая непротиворечивость, соответствие потокам данных. Поскольку исходно эти методики разрабатывались для первичной формализации объекта проектирования и деятельности, связанной с ним, то количественные характеристики в этих методах представлены слабо. Количественный анализ — оценка того, с какой вероятностью выполняются те или иные ветви процесса и какие ресурсы (время, стоимость) при этом затрачиваются — остаётся за рамками визуального моделирования. Это приводит к рассогласованию между структурным проектированием (созданием схемы) и параметрическим (назначением числовых характеристик), хотя именно их совместный анализ позволяет выявить узкие

Жаркова О. А., Насыров Р. В. Анализ инвариантных последовательностей процессов на основе вероятностно-взвешенных графов и символьных вычислений // СИИТ. 2026. Т. 8, № 3(27). С. 33-44. DOI: [10.54708/SIIT-2026-no3-p33](https://doi.org/10.54708/SIIT-2026-no3-p33). EDN: [XBVQLP](https://edn.sci.ugatu.su/XBVQLP).

Zharkova O. A., Nasyrov R. V. "Analysis of invariant process sequences based on probabilistically weighted graphs and symbolic computations" // SIIT. 2026. Vol. 8, no. 3(27), pp. 33-44. DOI: [10.54708/SIIT-2026-no3-p33](https://doi.org/10.54708/SIIT-2026-no3-p33). EDN: [XBVQLP](https://edn.sci.ugatu.su/XBVQLP). (In Russian).

места и оптимизировать процесс ещё на этапе моделирования. Актуальной задачей является построение методики количественного анализа этих схем для получения численных оценок эффективности, что позволило бы ещё на ранних стадиях проектирования оценивать такие параметры, как вероятность исполнения ветвей процесса и необходимые ресурсы. Предлагаемый подход позволяет устранить это рассогласование, позволяя построить на основе структурного описания процесса в граф со взвешенно-количественными показателями. Это позволяет применять различные алгоритмы анализа взвешенных графов, как в численном, так и в аналитическом виде. В целом, такая формализация даёт возможность перейти от статической диаграммы, имеющей описательный характер, к количественному анализу показателей эффективности выявленных последовательностей.

ПРЕДЛАГАЕМОЕ РЕШЕНИЕ

Упомянутые ранее схемы представляют компоненты деятельности, которые обычно входят в более общую систему деятельности, реализуемый специалистами в предметной области. Такая деятельность как правило носит повторяющийся характер и ее удобно представить в виде циклической последовательности или схемы с обратной связью. Сам процесс анализа тоже можно представить в виде последовательно содержащей ряд вложенных циклов.

Авторами предлагается представить процесс анализа эффективности в виде замкнутого цикла исследования эффективности, представленного на рис. 1. Предлагается строить этот цикл в виде взаимодействия двух контуров: вычислительного (синтез) и валидационного (анализ).

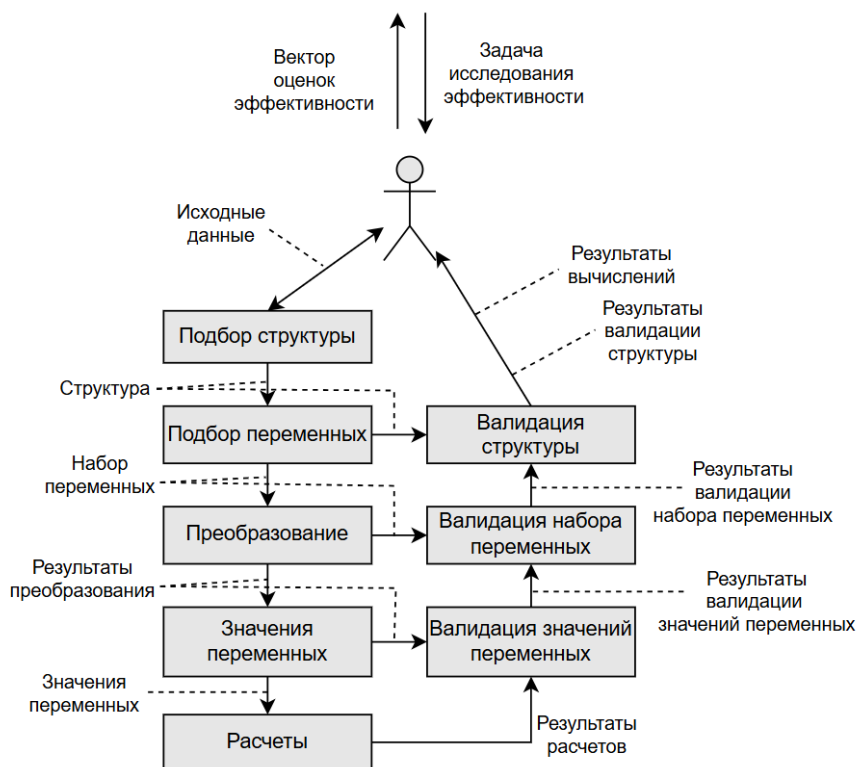


Рис. 1. Общая схема

Начальной точкой является блок ввода исходных данных, которые инициируют работу основного алгоритма. На первом этапе происходит подбор архитектуры модели и определение ключевых переменных. Далее следует этап преобразования, где абстрактные переменные наполняются конкретными значениями, формируя таким образом корректный массив для последующих вычислений. Результатом этого вычислительного этапа является вектор результирующих показателей, характеризующий текущий уровень эффективности системы.

Параллельно функционирует нижний контур — блок валидации. Он выполняет функции «контролера качества», последовательно верифицируя результаты каждого этапа основного цикла: проверяется логическая состоятельность структуры, адекватность набора переменных и достоверность полученных значений. Это позволяет отсеивать методологические ошибки на ранних стадиях, не допуская их до финальных расчетов.

Ключевая особенность схемы — наличие нескольких уровней обратной связи. Если полученный вектор показателей не соответствует целевым критериям, система не останавливается, а возвращается к этапу коррекции структуры модели. Запускается новый виток итерации, в ходе которого модель уточняется с учетом результатов предыдущего расчета и данных валидации.

Таким образом, представленная архитектура реализует принцип адаптивного управления: она не просто выполняет однократный расчет, а обеспечивает многократное уточнение модели, постепенно приближая выходные показатели к требуемым значениям за счет встроенных механизмов валидации и обратной связи.

МОДЕЛЬ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ – ПРЕДЛАГАЕМЫЕ ТЕХНОЛОГИИ

В общей схеме этапу «Исходные данные» соответствует UML/BPMN модель в схеме use case; «Подбор структуры / Структура» реализуется через действие «Транслировать UML/BPMN модель в граф»; «Набор переменных» соответствует действию «Задать вероятности P и веса M », а построение матриц P и M является промежуточным результатом этого этапа; этап «Расчеты» соответствует получению символьных выражений в RPN; наконец, «Результаты расчета / Реализации» ведут к выявлению узких мест.

Применение этой методики возможно для достижения различных целей. В качестве примера рассмотрим диаграмму вариантов использования для применения в академических целях. Использование данного веб-приложения позволяет транслировать структуру связей диаграммы последовательности (UML) или потока управления (BPMN) во взвешенный ориентированный граф. Задание вероятностей (P) переходов отражает частоту срабатывания конкретного маршрута, а веса (M) — ресурсоёмкость (время, стоимость, объем данных). Матричное представление и последующая символьная обработка позволяют перейти от абстрактной картинке к аналитическим выражениям, описывающим суммарную стоимость или пропускную способность всего процесса, выявляя узкие места на уровне инвариантных структур.

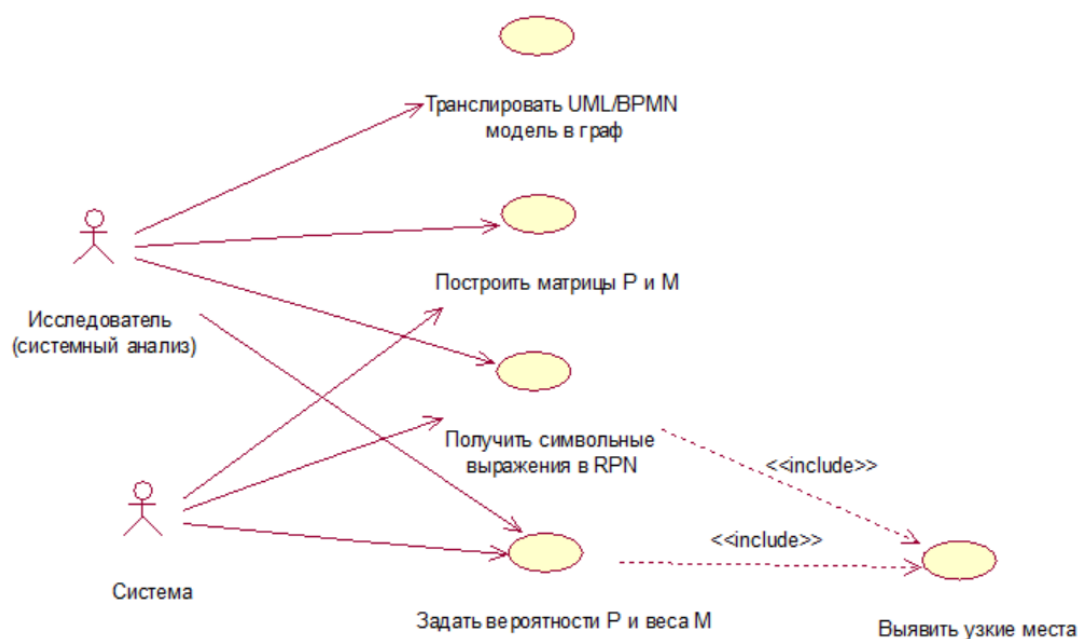


Рис. 2. Use Case для академических целей

Опишем сценарии данных вариантов использования:

Основных действующих лиц два: исследователь (инициирует действия); система (автоматически выполняет построение матриц и символьные преобразования). Сценарии представлены в табл. 1.

Таблица 1

Сценарии вариантов использования

Основные сведения этапа	Основной поток
<p>UC1: Транслировать UML/BPMN модель в граф <i>Описание:</i> Исследователь переносит структуру процесса из внешней нотации (UML, BPMN) в ориентированный граф системы путём ручного создания узлов и рёбер или импорта файла.</p>	<ol style="list-style-type: none"> 1. Исследователь на вкладке «Граф» создаёт узлы (кнопкой «Добавить узел» или двойным кликом) и рёбра, задавая для каждого ребра вероятность P и вес M в полях ввода. 2. При необходимости исследователь импортирует ранее сохранённый граф (форматы JSON, DOT) с помощью кнопки «Загрузить граф». 3. Система отображает граф, обновляет статистику (количество узлов и рёбер).
<p>UC2: Построить матрицы P и M из графа <i>Описание:</i> Система преобразует текущий граф в две матрицы размером $N \times N$ (N – число узлов). Выполняется только по команде исследователя.</p>	<ol style="list-style-type: none"> 1. Исследователь нажимает кнопку «Построить матрицы» на вкладке «Граф». 2. Система анализирует все рёбра, для каждого определяет индексы узлов (по возрастанию id) и заполняет матрицы P и M. 3. Строки матрицы P нормируются так, чтобы сумма вероятностей в строке равнялась 1 (если есть активные рёбра). 4. Система переключается на вкладку «Матрицы» и отображает заполненные таблицы; неиспользуемые ячейки остаются неактивными (кнопки с номером).
<p>UC3: Получить символьные выражения в RPN <i>Описание:</i> Исследователь инициирует символьную обработку матриц для получения аналитических выражений в обратной польской нотации.</p>	<ol style="list-style-type: none"> 1. Исследователь переходит на вкладку «Результаты» и нажимает «Вычислить результат». 2. Система собирает данные только из активных ячеек матриц и формирует два массива чисел, где каждая активная ячейка кодируется уникальным числовым идентификатором (100 + смещение для P, 200 + смещение для M). 3. Система отправляет POST-запрос на эндпоинт /sostavexp с этими матрицами. 4. Сервер выполняет алгоритм сокращения и возвращает четыре строки в RPN-формате (exp11, exp12, exp21, exp22). 5. Клиент заменяет числовые идентификаторы в строках на символы вида P_{ij} и M_{ij}. 6. Система сохраняет полученные выражения в текущем графе и отображает их на вкладке «Результаты» (в расширенном режиме – шаг 2).
<p>UC4: Задать вероятности P и веса M <i>Описание:</i> Исследователь указывает значения P и M для рёбер графа непосредственно при их создании или редактирует их позже (через матрицы).</p>	<ol style="list-style-type: none"> 1. При создании ребра исследователь вводит P (0–1) и M (целое ≥ 0) в соответствующие поля. 2. Система сохраняет эти параметры в объекте ребра (поле data). 3. Если исследователь желает изменить параметры существующего ребра, он переходит на вкладку «Матрицы», активирует нужную ячейку и редактирует значение.
<p>UC5: Выявить узкие места <i>Описание:</i> Исследователь анализирует полученные символьные выражения и соответствующие числовые значения для обнаружения критических элементов (например, рёбер с наибольшим весом или вероятностью).</p>	<ol style="list-style-type: none"> 1. Исследователь изучает числовые результаты (шаг 3 из интерфейса) и сопоставляет их с рёбрами графа. 2. Система не предоставляет автоматического выделения узких мест, поэтому исследователь вручную интерпретирует результаты. 3. При необходимости исследователь возвращается к графу и модифицирует параметры, повторяя вычисления.

В большинстве практических случаев определение значений параметров базируется на ретроспективном анализе выполненных проектов: из накопленных данных извлекаются статистические оценки того, как часто и с какими результатами реализовывались те или иные этапы. Вероятности переходов P при этом отражают наблюдаемую частоту успешного завершения одного этапа и перехода к следующему, а веса M отражают сведения о типичных временных

затратах или стоимости, зафиксированных в проектной документации. Такой подход позволяет опираться не только рассматривать некоторые теоретические построения, но и использовать эмпирически данные реально выполненных комплексов действий и проектов.

РЕАЛИЗАЦИЯ

На основе ранее проведенных исследований [Бак93, Нас94, Нас21, Нас22, Сад20, Хам23] рассмотрим общую концепцию программного проекта. С системной точки зрения проект представляет собой полнофункциональное веб-приложение для визуализации, обработки и анализа ориентированных графов и связанных с ними матриц вероятностей и весов. Приложение реализует комплексный подход к работе с графами, включающий их визуальное построение, параметризацию, реализация выполнения математических вычислений и разные способы визуализации результатов [Кор22]. Фрагмент интерфейса проекта представлен на рис. 3.

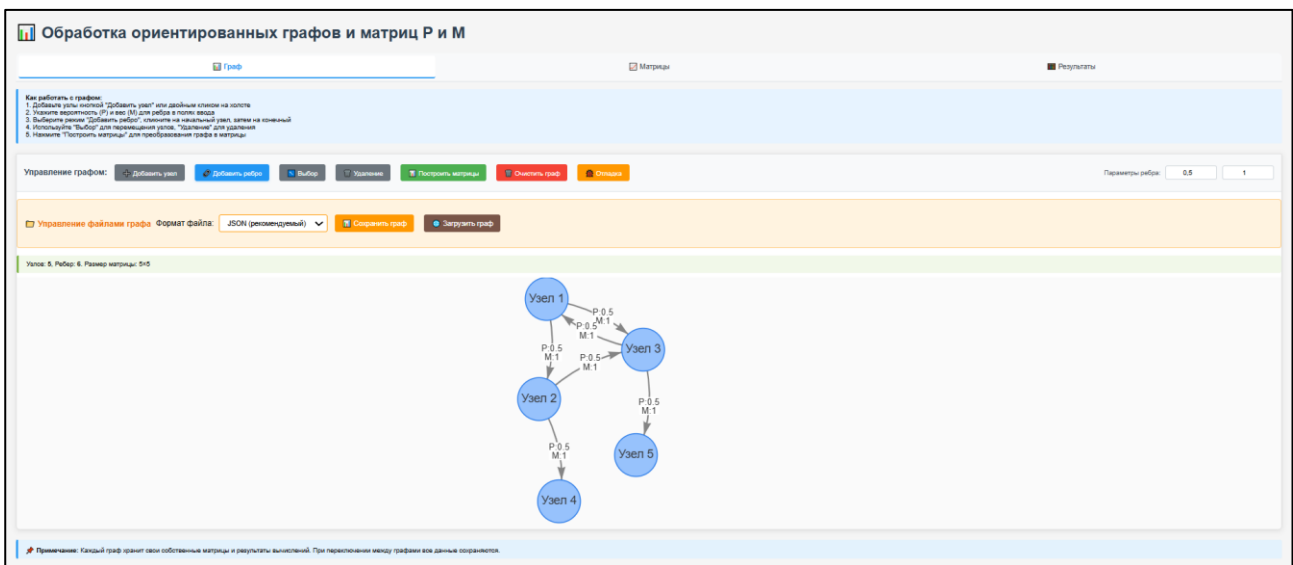


Рис. 3. Фрагмент интерфейса пользователя

Архитектура проекта представляет собой микросервисную реализацию [Нью21, Рич19, Сед22]:

1. Клиентская часть (index.html) представлена в табл. 2.

Таблица 2

Клиентская часть

Технологии	Цель	Ключевые особенности
HTML5, CSS3, JavaScript (ES6+), Vis.js	Интерактивный UI	Трехпанельный интерфейс (вкладки: «Граф», «Матрицы», «Результаты»)

Основные функции клиентской части:

- визуализация графов (Vis.js): интерактивное создание узлов и рёбер; несколько режимов взаимодействия (добавление, выбор, удаление); автоматическая расстановка узлов с физическим движком;
- управление матрицами: автоматическое преобразование графа в матрицы; интеллектуальное заполнение ячеек с проверкой корректности;
- работа с файлами: экспорт/импорт графов в форматах JSON и DOT; экспорт/импорт матриц в форматах JSON и CSV; предпросмотр загружаемых файлов;

• вычислительный модуль: отправка данных на сервер для символьной обработки; вычисление числовых выражений; разные режимы отображения результатов.

2. Серверная часть (main.py) представлена в табл. 3.

Таблица 3

Серверная часть

Технологии	Цель	Базовый стек
Python, FastAPI ¹ , Pydantic	Математическая обработка	Асинхронная обработка запросов, автоматическая валидация

Основные эндпоинты:

- `/sostavexpr` (POST) - символьная обработка матриц: принимает две матрицы одинакового размера; выполняет серию алгоритмических преобразований; возвращает символьные выражения в обратной польской нотации (RPN);

- `/calculate_rpn` (POST) - вычисление RPN выражений: принимает список выражений в RPN формате; выполняет математические вычисления; обрабатывает ошибки и исключительные ситуации;

- `/save_matrices_csv` (POST) - генерация CSV файлов: конвертирует матрицы в CSV формат; поддерживает разные форматы чисел;

- вспомогательные эндпоинты для работы с файлами.

Алгоритм работы приложения представлен в табл. 4.

Таблица 4

Алгоритм работы приложения

Шаг	Действие	Результат / Проверки
1. Построение графа	Создание узлов и ребер; ввод P (0–1) и M (целое ≥ 0) для каждого ребра.	Граф в памяти браузера.
2. Преобразование	Система строит матрицы P и M размером $N \times N$ (N — число узлов).	Сумма P в строке = 1; M — целые неотрицательные.
3. Символьная обработка	Клиент отправляет матрицы на сервер (<code>/sostavexpr</code>). Сервер возвращает RPN-строки, клиент заменяет ID на символы (P_{ij} , M_{ij}).	4 выражения в RPN-формате с символьными переменными.
4. Числовые вычисления	Клиент отправляет RPN на сервер (<code>/calculate_rpn</code>). Сервер возвращает числовые значения.	Итоговые числа для анализа (отображаются на вкладке «Результаты»).

В качестве основных компонентов стека технологий в рамках выбранной архитектуры были использованы:

Разделение компонентов на клиентскую и серверную части:

- *Клиент*: UI/UX, визуализация (Vis.js), управление состоянием графа.
- *Сервер (FastAPI)*: сложные математические алгоритмы.
- *Выгода*: снижение нагрузки на браузер, масштабируемость бэкенда, безопасность логики.

Компоненты технологического стека:

- *FastAPI*: высокая производительность, асинхронность, автодокументация (Swagger).

¹ FastAPI: веб-фреймворк для быстрого создания API на Python [Электронный ресурс] / Официальная документация. URL: <https://fastapi.tiangolo.com/ru> (дата обращения: 15.12.2024).

- *Vis.js*: специализированная библиотека для графов с физическим движком.
- *Vanilla JS*: минимум зависимостей, скорость работы.

Реализованные алгоритмические решения:

- *Обратная польская нотация (RPN)*: устраняет приоритеты операций, легко вычисляется через стек.
- *Символьная обработка*: позволяет анализировать структуру выражений до подстановки чисел, упрощает отладку.

ПРИМЕР ПРИМЕНЕНИЯ

Выбор предметной области: для анализа выбран процесс «Оформление онлайн-заказа в интернет-магазине» – типичная последовательность действий, с которой сталкиваются миллионы пользователей ежедневно. Этот процесс хорошо структурирован, имеет четкие этапы и возможные точки отказа, что делает его идеальным для анализа методом цепи Маркова.

Актуальность анализа: По данным Statista² [Sta23], средний показатель конверсии в электронной коммерции составляет 1.5–3.5%, что означает высокий процент отказов на различных этапах оформления заказа. Анализ эффективности этого процесса позволяет выявить проблемные места и предложить меры по оптимизации.

Рассмотрим детальное представление процесса по этапам [Гро18]. Предположим, что процесс оформления состоит из 5 основных этапов:

1. Выбор товара – пользователь добавляет товар в корзину.
2. Переход к оформлению – нажатие кнопки «Оформить заказ».
3. Ввод данных – заполнение формы с контактными данными и адресом доставки.
4. Выбор способа оплаты – выбор и подтверждение метода оплаты.
5. Подтверждение заказа – финальное подтверждение и получение номера заказа.

Каждый этап может завершиться либо переходом к следующему этапу, либо отказом (нецелевое состояние). Вероятности и временные оценки основаны на агрегированных данных исследований пользовательского поведения (Baymard Institute, 2023; Google Analytics benchmarks, 2024). Последовательность в виде диаграммы Use Case представлена на рис. 4.

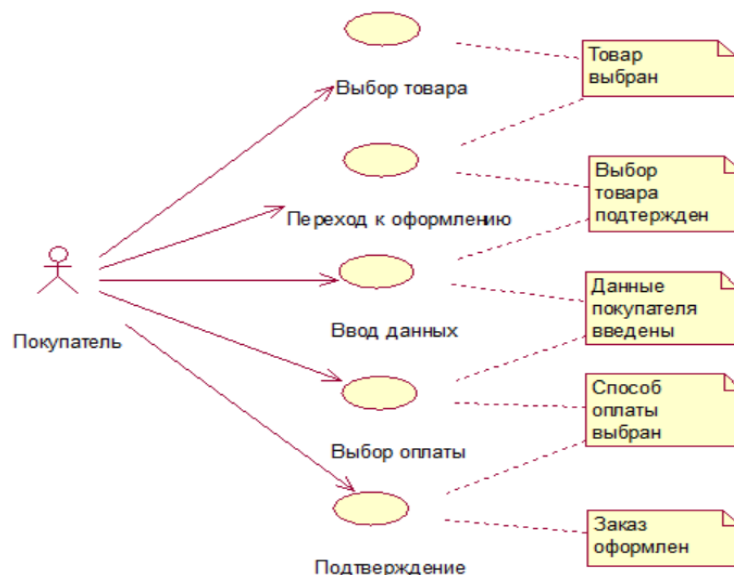


Рис. 4. Use Case диаграмма «Оформление онлайн-заказа в интернет-магазине»

² Электронный маркетинг во всем мире — статистика и факты. URL: <https://www.statista.com/topics/1446/e-mail-marketing> (дата обращения 03.04.2026).

Поскольку схема достаточно простая то её можно достаточно однозначно представить как последовательность вершин и связей графа. Дополнив её некоторыми характерными структурными элементами и параметрами, можно получить структур в форме схемы цепи Маркова.

Рассмотрим обоснование количественных оценок параметров строенной модели, в частности вероятностей переходов по этапам.

1. Выбор товара → Оформление (0.75): По данным Baymard Institute, только 75% пользователей, добавивших товар в корзину, переходят к оформлению заказа.

2. Переход к оформлению → Ввод данных (0.85): 15% пользователей покидают процесс на этапе перехода, чаще всего для сравнения цен или чтения отзывов.

3. Ввод данных → Выбор оплаты (0.70): Самый критичный этап – 30% отказов связаны со сложностью формы, необходимостью регистрации или concerns о безопасности данных.

4. Выбор оплаты → Подтверждение (0.90): При наличии удобных способов оплаты отказ происходит лишь у 10% пользователей.

5. Подтверждение → Цель (0.95): При условии корректной работы системы, 95% подтверждений успешны.

6. Проблема F → Ввод данных (0.60): 60% пользователей готовы повторить попытку при временной ошибке системы.

Аналогично рассмотрим оценки времени выполнения этапов. В качестве источников данных рассматривались следующие:

- Юзабилити-тестов среднего интернет-магазина [Наз22, Гор20, Зен19].
- Данных аналитики поведения пользователей [Лав14].
- Эргономических норм на выполнение действий [Ста16].

На основании проведенного анализа указанных источников были получены следующие средние значения затрат времени:

- Выбор товара: 2 мин (просмотр, выбор количества).
- Переход к оформлению: 1 мин (ожидание загрузки, осмысление).
- Ввод данных: 3 мин (заполнение 5–7 полей).
- Выбор оплаты: 2 мин (выбор метода, ввод данных карты).
- Подтверждение: 1 мин (проверка, нажатие кнопки).

Аналогичным образом были получены средние значения времени при отказах:

- Быстрый отказ: 0.5–1 мин
- Системная ошибка: 2–5 мин

Подробный граф процесса «Оформление онлайн-заказа в интернет-магазине» представлен на рис. 5. В каждом процессе можно выделить один или несколько этапов, которые могут рассматриваться как критические процессы или в качестве «бутылочного горлышка», т. е. такие, которые существенно ограничивают его эффективность. В рассматриваемом примере таким процессом является «ввод данных». Для изучения степени влияния не общую эффективность и определения порогового значения реализуем параметрический анализ результирующей эффективности от эффективности данного этапа. Для этого будем рассматривать все значения как константы (т. е. «закрепим» их). А вероятность выделенного этапа рассматриваем как параметр. Результаты вычислений сведем в табл. 5.

Графическая форма результатов вычислений представлена на рис. 6 и 7.

Полученные результаты расчетов, представленные на графиках рис. 6 и 7 показывают зависимость эффективности реализации процесса от эффективности одного из этапов процесса – «Ввод данных». Увеличение безошибочности ввода и затрат времени на него возможно реализовать повышением качества интерфейса информационной системы до обеспечения приемлемых значений исследованных характеристик. При этом целевое значение уровня вероятности – примерно 0,9. Конкретные технологические решения зависят от технологического стека системы и выходят за рамки тематики данной статьи. Аналогичный анализ можно провести для других этапов с целью оптимизации процесса в целом.

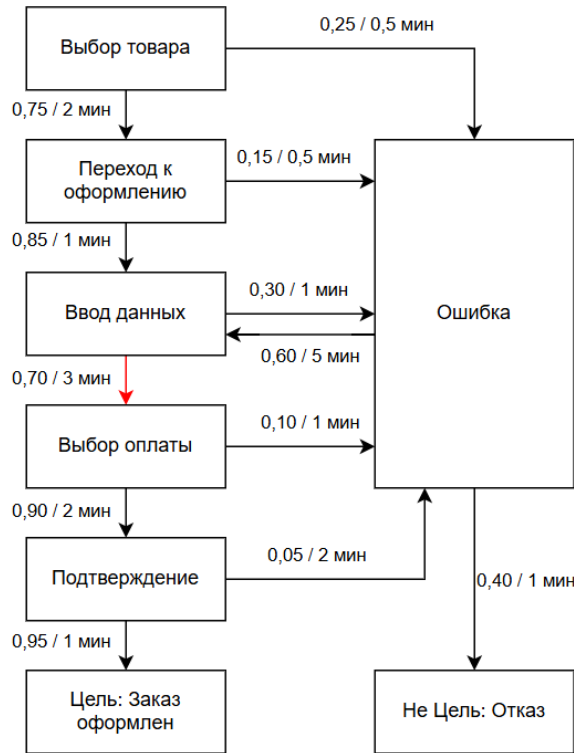


Рис. 5. Подробный граф процесса «Оформление онлайн-заказа в интернет-магазине»

Таблица 5

Результаты вычислений

Критический элемент – «Ввод данных»	Целевая вероятность	Нецелевая вероятность	Целевые ресурсы	Нецелевые ресурсы
0,1	0,16	0,84	5,16	9,24
0,2	0,29	0,71	5,63	7,07
0,3	0,4	0,6	6,07	5,72
0,4	0,48	0,52	6,46	4,85
0,5	0,56	0,44	6,81	4,26
0,6	0,62	0,38	7,1	3,85
0,7	0,67	0,33	7,36	3,54
0,8	0,72	0,28	7,58	3,3
0,9	0,76	0,24	7,77	3,11

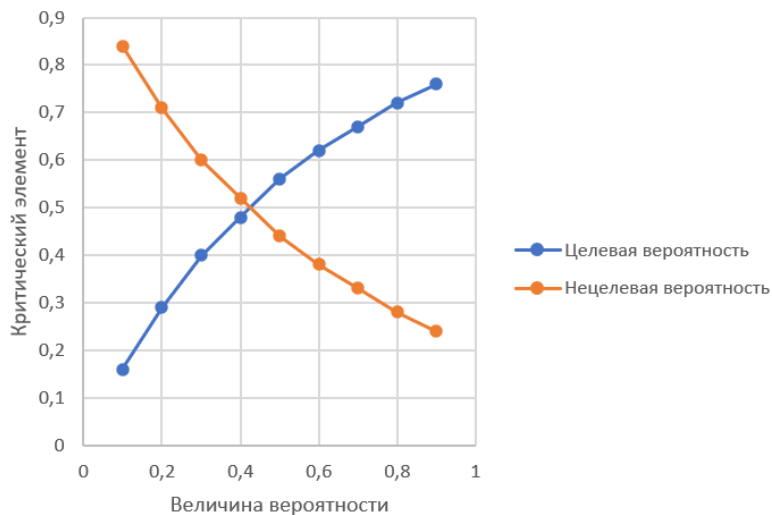


Рис. 6. График вероятностей

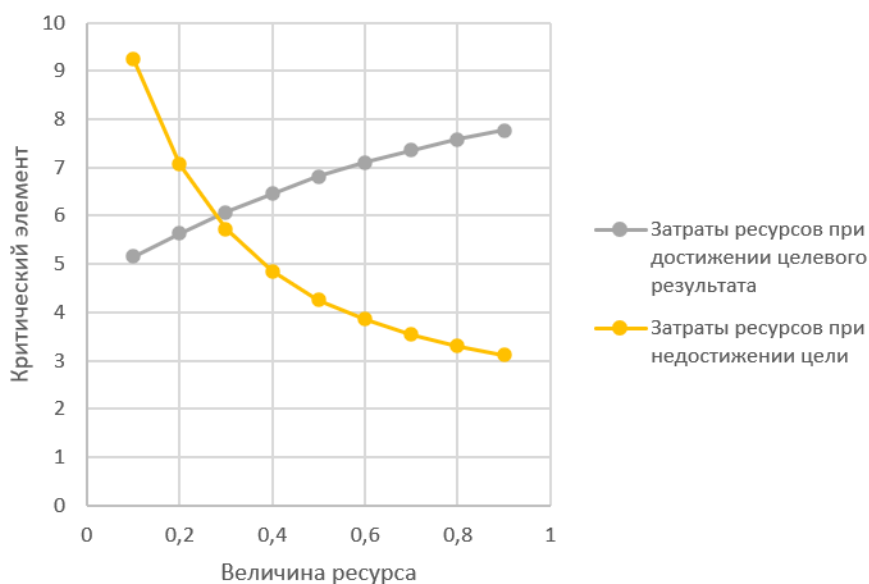


Рис. 7. График затрат ресурсов

Таким образом, предложенная методика позволяет не только точно определять узкие места, но и количественно оценить необходимый уровень их оптимизации для достижения целевых показателей всего процесса. Это дает возможность ранжировать этапы по степени их влияния на итоговую эффективность и обоснованно распределять ресурсы при совершенствовании системы.

ВЫВОДЫ И ЗАКЛЮЧЕНИЕ

В ходе проведенного исследования была решена задача устранения рассогласования между структурным и количественным анализом процессов путем разработки методики преобразования визуальных моделей в вероятностно-взвешенные графы с последующей символьной обработкой. Традиционные нотации (BPMN, UML) эффективны для визуализации инвариантных структур, но не позволяют оценивать вероятностные и ресурсные характеристики. Предложенный подход трансформирует статические диаграммы в объект анализа.

В отличие от существующих подходов, которые сводятся к качественному моделированию либо к статистической обработке эмпирических данных, предлагаемая методика сочетает визуальную интерпретацию структурных диаграмм с аналитическими возможностями символьных преобразований. Данный подход обеспечивает не только получение численных оценок (вероятность достижения целевого состояния, ожидаемые временные или стоимостные затраты), но и содержательный анализ влияния отдельных элементов процесса на результирующие показатели, что позволяет осуществлять локализацию проблемных элементов на стадии проектирования.

В процессе исследования были получены следующие основные результаты:

1. Предложена архитектура адаптивного цикла анализа с вычислительным и валидационным контурами, обеспечивающая итеративное уточнение модели и контроль достоверности результатов.
2. Разработана методика перевода UML/BPMN-диаграмм в вероятностно-взвешенный граф с матричным представлением вероятностей (P) и весов (M), что создает основу для количественного анализа.
3. Реализован механизм символьной обработки, позволяющий получать аналитические выражения в обратной польской нотации. Это дает возможность выявлять влияние отдельных элементов на итоговые показатели инвариантных структур процесса.

4. Создано веб-приложение, реализующее полный цикл: визуальное построение графа, ввод эмпирических данных, автоматическое формирование матриц, их символьную обработку и численный расчет показателей.

Предложенное решение интегрирует визуальное моделирование с символьными вычислениями, обеспечивая инструмент для выявления узких мест и оптимизации параметров процессов на этапе проектирования. Дальнейшее развитие предполагает расширение поддерживаемых нотаций, автоматизацию поиска узких мест и интеграцию с внешними источниками данных.

СПИСОК ЛИТЕРАТУРЫ | REFERENCES

- [Бак93] Бакусов Л. М., Насыров Р. В., Лебедев Е. Г. Причинный анализ для принятия решений: Учеб. пособие. Уфа: УГАТУ, 1993. 96 с. Bakusov L. M., Nasyrov R. V., Lebedev E. G. Causal Analysis for Decision-Making: A Tutorial. Ufa: Ufa State Aviation Technical University, 1993. (In Russian).
- [Гор20] Горячкин Б. С., Хитрин Д. М. Эргономичность пользовательских web-интерфейсов через призму юзабилити // E-Scio. 2020. № 3 (42). PQHSMA. Goryachkin B. S., Khitrin D. M. Ergonomics of user web interfaces through the prism of usability // E-Scio. 2020. No. 3 (42). (In Russian). PQHSMA.
- [Гро18] Гросс Д., Шортли Дж. Ф. и др. Основы теории массового обслуживания. М.: Ленанд, 2018. 448 с. Gross D., Shortle J. F., et al. (2018). Fundamentals of Queueing Theory (5th ed.). Wiley.
- [Зен19] Зенг В. А. Оценка качества проектирования пользовательских интерфейсов нового поколения // Известия ТулГУ. Технические науки. 2019. № 12. ZCMSSJ. Zeng Valeria Andreevna. Assessment of the design quality of new-generation user interfaces // Bulletin of Tula State University. 2019. No. 12. (In Russian). ZCMSSJ.
- [Кор22] Кормен Т. Х., Лейзерсон Ч. И. и др. Алгоритмы: построение и анализ. М.: Вильямс, 2022. 1328 с. Cormen T. H., Leiserson C. E., et al. (2022). Introduction to Algorithms (4th ed.). MIT Press.
- [Лав14] Лавлинский Н. Е. Применение систем веб-аналитики для исследования поведения пользователей сайта // Научные труды Вольного экономического общества России. 2014. Т. 186. С. 289-293. VKOSKB. Lavlinsky N. E. Application of web analytics systems for studying the behavior of website users // Scientific works of the Free Economic Society of Russia. 2014. Vol. 186. P. 289-293. (In Russian). VKOSKB.
- [Мир25] Миронов В. В., Гусаренко А. С., Тугузбаев Г. А. Персонализация графических конструкторских документов в ситуационно-ориентированной среде: использование XSL-трансформации // СИИТ. 2025. Т. 7, № 2(21). С. 163-176. JEGOKH. Mironov V. V., Gusarenko A. S., Tuguzbaev G. A. Personalization of graphic design documents in a situation-oriented environment: using XSL transformation // SIIT. 2025. Vol. 7, No. 2(21). P. 163-176. (In Russian). JEGOKH.
- [Наз22] Назаров Д. М. Методы оценки юзабилити и доступности веб-сайтов электронной коммерции // Integral. 2022. № 4. JAQTII. Nazarov D. M. Methods for assessing usability and accessibility of e-commerce websites // Integral. 2022. No. 4. (In Russian). JAQTII.
- [Нас21] Насыров Р. Оценка эффективности процессов разработки программного обеспечения на основе вероятностных причинно-следственных моделей // 3-я Междунар. конф. по системам управления, математическому моделированию, автоматизации и энергоэффективности (SUMMA), 10-12 нояб. 2021. YLGXBL. Nasyrov R. Evaluating the effectiveness of software development processes based on probabilistic causal models // 2021 3rd Int. Conf. on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), 10-12 Nov. 2021. YLGXBL. (In Russian).
- [Нас22] Насыров Р. В. Причинный подход к построению бионических вычислений на основе рекурсивных моделей анализа данных // СИИТ. 2022. Т. 4, № 1(8). С. 27-36. UOMMOU. Nasyrov R. V. Causal approach to the construction of bionic computations based on recursive models of data analysis // SIIT. 2022. Vol. 4, No. 1(8). P. 27-36. (In Russian). UOMMOU.
- [Нас94] Насыров Р. В. Автоматизированная система принятия решений на основе причинных моделей проблемных ситуаций: Автореф. дис. ... канд. техн. наук. Уфа: УГАТУ, 1994. ZIRFMR. Nasyrov R.V. Automated decision-making system based on causal models of problem situations: Abstract of Cand. Sci. (Eng.) Dissertation. Ufa, Ufa State Aviat. Univ., 1994. (In Russian). ZIRFMR.
- [Нью21] Ньюмен С. Создание микросервисов: проектирование детализированных систем. СПб.: Питер, 2021. 624 с. Newman S. (2021). Building Microservices: Designing Fine-Grained Systems (2nd ed.). O'Reilly Media.
- [Рич19] Ричардсон К. Шаблоны микросервисов: с примерами на Java. М.: Диалектика, 2019. 544 с. Richardson, C. (2018). Microservices Patterns: With examples in Java. Manning Publications.
- [Сад20] Садыкова Л. Р., Насыров Р. В. Приложение оценки эффективности обобщенного сценария действий: Свид. гос. рег. программ для ЭВМ № 2020666733 РФ; заявл. 25.11.2020. CDVUQX. Sadykova L. R., Nasyrov R. V. Application for assessing the effectiveness of a generalized action scenario: Certif. computer program No. 2020666733 of the Russian Federation; declared 25.11.2020. (In Russian). CDVUQX.
- [Сед22] Седерер П. Разработка микросервисов на Python: создание, развертывание и управление микросервисами с помощью Python. М.: ДМК Пресс, 2022. 440 с. Ceder P. (2021). Python Microservices Development: Build, deploy, and manage microservices with Python. Packt Publishing.

- [Сем25] Семенова В. А. Формирование контекста для вывода формальных понятий из неполных и противоречивых данных // СИИТ. 2025. Т. 7, № 1(20). С. 59-67. [TRCDPY](#).
Semenova V. A. Formation of context for the derivation of formal concepts from incomplete and contradictory data // SIIT. 2025. Vol. 7, No. 1(20). P. 59-67. (In Russian). [TRCDPY](#).
- [Ста16] Стахно Р. Е., Алексеев С. А. Эргономические принципы технологии распределения функций между пользователем и средствами автоматизации в органах внутренних дел // European science. 2016. № 11 (21). [WYHAID](#).
Stakhno R. E., Alekseev S. A. Ergonomic principles of the technology of distributing functions between the user and automation tools in internal affairs agencies // European Science. 2016. No. 11 (21). (In Russian). [WYHAID](#).
- [Хам23] Хамзин Р. З., Насыров Р. В. Мониторинг эффективности производства предприятия: KPI и OEE // Академическая публицистика, № 5-2/2023. С. 76-91. [FAPSWW](#).
Khamzin R. Z., Nasyrov R. V. Monitoring the production efficiency of an enterprise: KPI and OEE // Academic Publication, No. 5-2/2023. P. 76-91. (In Russian). [FAPSWW](#).

ОБ АВТОРАХ | ABOUT THE AUTHORS

ЖАРКОВА Оксана Андреевна

Уфимский университет науки и технологий, Россия.
oksana.zharkova19@yandex.ru ORCID: 0009-0009-2408-4003.
Студ. спец. «Информатика и вычислительная техника».

ZHARKOVA Oksana Andreevna

Ufa University of Science and Technology, Russia.
oksana.zharkova19@yandex.ru ORCID: 0009-0009-2408-4003.
Student of the specialty "Computer science and engineering".

НАСЫРОВ Рашит Вильевич

Уфимский университет науки и технологий, Россия.
nrash@yandex.ru ORCID: 0000-0002-5402-9525.
Канд. техн. наук, доцент.

NASYROV Rashit Vilevich

Ufa University of Science and Technology, Russia.
nrash@yandex.ru ORCID: 0000-0002-5402-9525.
Candidate of Technical Sciences, Associate Professor.

МЕТАДАННЫЕ | METADATA

Заглавие: Анализ инвариантных последовательностей процессов на основе вероятностно-взвешенных графов и символьных вычислений.

Авторы: Жаркова О. А., Насыров Р. В.

Аннотация: В данной статье рассматривается проблема расширения возможностей структурного анализа процессов, возникающая при использовании традиционных графических нотаций (UML, BPMN) в сторону формирования количественных оценок, характеризующих выявленные процессы. Авторы предлагают методику преобразования статических диаграмм указанных нотаций в графы, нагруженные вероятностными и ресурсными величинами как в символьных обозначениях с последующей символьной обработкой, так и в численной форме с последующим получением величин оценок качества процесса. Предлагаемый подход позволяет перейти от описательного моделирования к численной оценке эффективности процессов. Предложена архитектура адаптивного цикла анализа, включающая вычислительный и валидационный контуры, реализующие итеративное уточнение модели. Предлагаемая методика реализована в виде микросервисного веб-приложения на базе клиент-серверной архитектуры с использованием FastAPI, Python и библиотеки Vis.js, обеспечивающего визуальное построение графов, формирование матриц вероятностей и весов, а также получение аналитических выражений в обратной польской нотации. На примере типового процесса оформления онлайн-заказа в интернет-магазине продемонстрирована возможность выявления узких мест и количественной оценки необходимого уровня оптимизации параметров. Предложенный подход позволяет интегрировать различные методы визуального структурного моделирования с символьными и количественными вычислениями, предоставляя инструмент для обоснованного принятия решений на этапе проектирования информационных систем.

Ключевые слова: Микросервисы; FastAPI; Python; UML; BPMN; вычисления на графах; сервис.

Язык: Русский.

Статья поступила в редакцию 28 марта 2026 г.

Title: Analysis of invariant process sequences based on probabilistically weighted graphs and symbolic computations.

Authors: Zharkova O. A., Nasyrov R. V.

Abstract: This article addresses the challenge of expanding the capabilities of structural process analysis, which arises when using traditional graphical notations (UML, BPMN), to include the generation of quantitative assessments characterizing the identified processes. The authors propose a method for transforming static diagrams of the specified notations into graphs loaded with probabilistic and resource quantities both in symbolic notations with subsequent symbolic processing and in numerical form with the subsequent generation of process quality assessment values. The proposed approach enables the transition from descriptive modeling to a numerical assessment of process efficiency. An architecture for an adaptive analysis cycle is proposed, including computational and validation loops implementing iterative model refinement. The proposed method is implemented as a microservice web application based on a client-server architecture using FastAPI, Python, and the Vis.js library, providing visual graph construction, the generation of probability and weight matrices, and the generation of analytical expressions in reverse Polish notation. Using a typical online ordering process in an online store as an example, we demonstrate the possibility of identifying bottlenecks and quantifying the required level of parameter optimization. The proposed approach integrates various visual structural modeling methods with symbolic and quantitative computations, providing a tool for informed decision-making during the information system design phase.

Key words: Microservices; FastAPI; Python; UML; BPMN; graph computing; service.

Language: Russian.

The article was received by the editors on 28 March 2026.