

УДК 004.89

ИНТЕРПРЕТАЦИЯ ДИАГРАММ ПРОЦЕССОВ ДЛЯ ГЕНЕРАЦИИ АЛГОРИТМОВ И ПРОГРАММНОГО КОДА

Воробьев Ф.М. (ГУАП)

Научный руководитель – ассистент, Вершинин В.К. (ИТМО)

Введение. Графические нотации процесса, такие как блок-схемы (flowcharts), диаграммы деятельности UML (UML Activity) и Business Process Model and Notation (BPMN) представляют собой общепринятые визуальные языки описания алгоритмов и бизнес-процессов, используемые аналитиками для документирования логики решений человеком читаемым способом. В промышленной практике эти диаграммы выступают связующим звеном между бизнес-аналитиками, формализующими требования, и разработчиками, реализующими алгоритмы в коде. При этом автоматизированное преобразование таких диаграмм в структурированные алгоритмические представления и программный код остаётся нерешённой задачей из-за сложности топологии, многообразия нотаций, пересечений, слияний потоков и ошибок в визуальных образах.

В литературе предложены методы прямой реконструкции программного кода из изображений диаграмм с использованием vision-language моделей (VLM), однако такие модели подвержены галлюцинациям – генерации структур, не соответствующих исходному изображению. Существующие детерминированные методы реконструкции графов демонстрируют формальную корректность для простых схем, однако они часто теряют управление потоком при сложных топологиях, пересечениях и слияниях, что приводит к ошибкам в восстановлении логики диаграмм [1].

Целью данной работы является разработка архитектуры гибридной системы интерпретации диаграмм процессов, сочетающей детерминированную реконструкцию графа и использование легковесных языковых моделей (LLM) для семантической валидации, что позволяет генерировать алгоритмические описания и программный код с измеримой структурной корректностью.

Основная часть. Предлагаемый подход включает несколько модульных этапов.

С помощью свёрточной нейронной сети (CNN) сначала классифицируется конкретная нотация диаграммы: flowchart, UML Activity или BPMN – что позволяет адаптировать последующие этапы анализа под правила соответствующей нотационной грамматики и экономить ресурсы системы.

Затем выполняется детекция элементов диаграммы. Используется дообученная модель объектной детекции на архитектуре трансформер RF-DETR-Seg-Nano для локализации базовых графических объектов: блоков процессов, условий, блоков начала и конца и стрелочных элементов. В отличие от классических CNN-детекторов семейства YOLO или Faster R-CNN, которые полагаются на локальные признаки и пост-обработки, как non-maximum suppression, трансформер-детекторы формулируют задачу обнаружения как end-to-end set-prediction с механизмом self-attention, что обеспечивает моделирование глобального контекста сцены и более устойчивую локализацию объектов в сложных визуальных структурах. Кроме того, стрелки на диаграммах традиционно сложно детектировать, поскольку они могут иметь разные формы, размеры, углы поворота и взаимное перекрытие с узлами или другими линиями, а стандартные локальные признаки CNN нередко ошибочно классифицируют стрелку как часть линии или узла [2].

На основе бинарной маски линий проводится skeleton-обработка (skeletonization) для построения центра линий, определения точек разветвления и слияния и формирования ориентированного графа, описывающего поток управления.

Текст внутри узлов и на переходах извлекается с использованием технологии автоматического распознавания текста (OCR), после чего интегрируется в графовое представление как свойства узлов и рёбер – это стандартная практика современных

пайплайн-систем для извлечения структуры и текста из диаграмм.

После чего выполняется семантическая валидация с помощью легковесной (7B-класс) языковой модели, которая применяется для проверки согласованности текста в графе, устранения ошибок OCR и выявления потенциально некорректных логических переходов, противоречивых условий или пропущенных ветвей. Такой гибридный подход базируется на сочетании детерминированного восстановления структуры и семантической интерпретации, избегающей полностью black-box мультимодальных решений, что демонстрируется в работах типа Flowchart2Mermaid, где использование чистой VLM требует дальнейшего контроля корректности [3].

Важной особенностью архитектуры является модульность: каждый этап может быть независимо оценён и расширен для поддержки новых нотаций или логических проверок, а промежуточное представление (JSON-граф) обеспечивает воспроизводимость и измеримость результатов. Такая модульная система согласуется с рекомендациями по созданию структурно интерпретируемых пайплайн-архитектур, отмеченными в обзоре методов извлечения содержимого из технических диаграмм [1].

Предварительные эксперименты, проведённые на наборе диаграмм из открытых источников, показали, что топологически-ориентированная реконструкция снижает число структурных ошибок по сравнению с эвристиками на основе bounding box, особенно в диаграммах с пересечениями и узлами слияния, где геометрические методы дают нестабильные результаты.

Семантическая валидация LLM позволяет исправлять ошибки OCR и улучшать согласованность логических условий в графе, что важно для последующей генерации алгоритмического описания и программного кода.

Выводы. Предложена гибридная архитектура интерпретации диаграмм процессов, которая сочетает детерминированные методы реконструкции ориентированного графа и использование легковесных языковых моделей для семантической проверки.

Анализ решения демонстрирует потенциал гибридного подхода, который позволяет улучшить структурную корректность реконструируемых графов по сравнению с традиционными эвристическими методами и обеспечивает более контролируемые результаты по сравнению со сквозными VLM-решениями.

Практическая значимость заключается в том, что предлагаемый инструмент позволяет разработчикам программного обеспечения автоматизировать часть рутинного преобразования диаграмм, созданных аналитиками, в формальные алгоритмы и программный код, снижая количество ошибок внедрения и повышая эффективность разработки.

В дальнейших исследованиях планируется расширение набора поддерживаемых нотаций, включение механизма проверки логической корректности диаграммы с точки зрения допустимости состояний и переходов, а также формализация метрик для сравнительного анализа методов реконструкции.

Список использованных источников:

1. Nick Bray, Michael Hempel, Matthew Boeding, Hamid Sharif, Decoding Technical Diagrams: A Survey of AI Methods for Image Content Extraction and Understanding, Information // mdpi.com, 2026.

2. Schäfer B., Keuper M., Stuckenschmidt H. Arrow R-CNN for handwritten diagram recognition // IJDAR, 2021.

3. Pritam Deka and Barry Devereux, Flowchart2Mermaid: A Vision-Language Model Powered System for Converting Flowcharts into Editable Diagram Code // arXiv.org, 2025.

Автор _____ Воробьев Ф.М.

Научный руководитель _____ Вершинин В.К.