

УДК 004.85, 004.421, 519.688

**ГЕНЕРАЦИЯ UML-МОДЕЛЕЙ С ПОМОЩЬЮ ИСКУССТВЕННОГО
ИНТЕЛЛЕКТА В КОНТЕКСТЕ РАЗВИТИЯ ИТ-ОБРАЗОВАНИЯ**

Киреев Н.Б., Кашникова И.В.

Институт информационных технологий БГУИР, г. Минск, Республика Беларусь,

kireev@bsuir.by, kashnikava@bsuir.by

В статье исследуется применение ИИ для генерации UML-моделей в ИТ-образовании. Анализируется преобразование текстовых описаний в формальные модели и оценивается соответствие результатов профессиональным стандартам.

Ключевые слова: искусственный интеллект, UML-моделирование, ИТ-образование, автоматизация проектирования, образовательные технологии.

Современная разработка программного обеспечения характеризуется ростом сложности и требований к скорости выпуска продуктов. В условиях Agile и DevOps традиционные, трудоемкие методы анализа и проектирования все чаще становятся «узким местом», не успевая за темпом итеративных циклов разработки.

В то время как применение систем искусственного интеллекта для автоматизации рутинных задач кодирования, таких как генерация кода, комментирование и рефакторинг (на примере GitHub Copilot, Amazon CodeWhisperer и аналогов), стало широко распространенной практикой, задача его применения на более ранних, концептуальных стадиях жизненного цикла ПО — этапах анализа требований и проектирования архитектуры — остается комплексной и менее изученной. Известно, что ИИ способны генерировать код по UML-диаграммам с качеством, сопоставимым с уровнем junior-разработчика. Однако обратная задача — генерация самих UML-моделей на основе текстовых описаний — исследована в значительно меньшей степени.

Данное исследование фокусируется на оценке возможностей ИИ именно в этой области: моделировании структур и процессов ПО с использованием графических нотаций, в частности, UML. Для решения задачи генерации диаграмм предлагается использовать в качестве посредника между ИИ и человеком PlantUML — инструмент с открытым исходным кодом, который позволяет создавать диаграммы в виде текстового кода и визуализировать их.

Цель работы — экспериментально оценить, достаточно ли когнитивных способностей современных ИИ-систем для построения адекватных UML-моделей на основе текстовых описаний предметной области и пользовательских сценариев (user story), используя PlantUML в качестве промежуточного звена.

В качестве основного метода исследования был выбран эксперимент, позволяющий объективно оценить возможности и ограничения ИИ-систем в решении конкретной прикладной задачи. Для преобразования текстовых запросов (промптов) в графические модели использовался инструмент PlantUML — система с открытым исходным кодом, позволяющая описывать диаграммы в виде текстового кода на своем Domain-Specific Language (DSL) и визуализировать их. Данный выбор обусловлен ключевым технологическим ограничением: современные крупные языковые модели не способны напрямую генерировать графические файлы (изображения), но демонстрируют высокую эффективность при работе со структурированными текстовыми языками, к которым относится язык PlantUML. Это делает его идеальным посредником (прокси) между ИИ и конечным пользователем-модельером.

Для достижения цели исследования и решения поставленных задач была разработана и применена следующая процедура эксперимента, состоящая из последовательных этапов:

- Формулировка текстового промпта. На этом этапе создавалось детализированное текстовое описание на естественном языке (русском), моделирующее фрагмент предметной области или сценарий взаимодействия. Описание включало ключевые сущности, их атрибуты, методы и взаимосвязи для диаграмм классов; акторов и их цели — для диаграмм вариантов использования; последовательность сообщений между объектами — для диаграмм последовательностей.

- Генерация кода для PlantUML. Сформулированный промпт передавался на вход одной из тестируемых ИИ-систем (Mistral, GitHub Copilot, ChatGPT) с явным указанием требования сгенерировать код на языке PlantUML для конкретного типа диаграммы.

- Визуализация полученного кода. Сгенерированный ИИ текстовый код проверялся на синтаксическую корректность и затем визуализировался в стандартном онлайн-редакторе PlantTextUML. Это позволяло получить итоговое графическое представление диаграммы.

- Анализ результата. Заключительный этап включал двухуровневый анализ,

состоящий из визуального и структурного компонентов. Визуальный анализ предусматривал проверку читаемости и общей логичности компоновки элементов диаграммы, в то время как структурный анализ охватывал детальную проверку соответствия полученной модели исходному текстовому запросу с точки зрения семантической адекватности, а также соответствия правилам нотации UML с точки зрения синтаксической корректности. Все результаты фиксировались для последующего сравнительного анализа.

В первом эксперименте, посвященном сравнительному анализу генерации диаграмм классов, всем тестируемым ИИ-системам (Mistral, Copilot и ChatGPT) был представлен одинаковый промпт: «Сгенерируй UML-диаграмму классов для типового приложения бронирования театральных билетов. Отобрази ассоциации между классами. Для вывода используй код PlantUML. Язык описания — русский». Все системы успешно справились с задачей, сгенерировав синтаксически корректный код, который был визуализирован в читаемые диаграммы.

Ключевым наблюдением стали существенные структурные различия в предложенных моделях: каждая ИИ-система представила уникальный набор классов и ассоциаций. Это наглядно демонстрирует вероятностный характер преобразования текстового описания в формальную модель и зависимость результата от обучающих данных конкретной системы. Наиболее проработанной и логически полной оказалась диаграмма, сгенерированная ChatGPT, включающая такие сущности, как «Билет», «Сцена», «Платеж», с корректно отображенными связями между ними. Результаты от Mistral и Copilot были более абстрактными и менее детализированными.

Таким образом, эксперимент подтвердил, что современные ИИ-системы способны генерировать базовые диаграммы классов, качество которых сопоставимо с работой аналитиков начального уровня. Однако полученные результаты обязательно требуют верификации и возможной доработки экспертом-архитектором.

В ходе второго эксперимента исследовалась возможность генерации двух типов UML-диаграмм — последовательности и классов — на основе табличного описания бизнес-сценария, структурированного по этапам "ввод, расчет, представление данных". Результаты показали выраженную разницу в качестве генерации для разных типов диаграмм. Так, диаграмма последовательности, созданная с помощью Copilot, была построена на высоком уровне детализации: система корректно идентифицировала ключевые объекты (Пользователь, Интерфейс, Контроллер, БазаДанных) и точно отразила последовательность сообщений между ними, адекватно воплотив логику исходного сценария. В то же время, диаграмма классов, сгенерированная на основании того же описания, оказалась существенно более абстрактной, содержала лишь базовые элементы и требовала значительной доработки для практического применения, хотя и могла служить концептуальной основой для дальнейшего проектирования.

В третьем эксперименте оценивалась способность ИИ-систем генерировать корректные диаграммы вариантов использования на основе детального текстового описания процесса аутентификации пользователя. Результаты показали, что даже при четко сформулированном промпте возможности протестированных систем (на примере Copilot) для построения данного типа диаграмм остаются крайне ограниченными. Сгенерированная диаграмма оказалась неудовлетворительной и демонстрировала фундаментальное непонимание синтаксиса и семантики нотации UML для Use Case: вместо корректных акторов и прецедентов ИИ представил акторов в виде классов, а сценарии — в качестве операций внутри этих классов. Кроме того, диаграмма игнорировала ключевые отношения включения и расширения между прецедентами, что делает её непригодной для практического использования. Полученный результат указывает на существенные пробелы в обучающих данных, связанных со специализированными правилами построения диаграмм вариантов использования, и подтверждает, что данная область требует дальнейшего совершенствования языковых моделей либо разработки специализированных инструментов.

В ходе четвертого эксперимента исследовалась способность ИИ-систем к трансформации моделей на основе графического контекста. В качестве исходных данных

Copilot получил изображение диаграммы вариантов использования и текстовый запрос на генерацию диаграммы деятельности для процесса авторизации. Результат показал, что при наличии достаточного контекстного окружения ИИ успешно справляется с задачей преобразования одной модели в другую: сгенерированная диаграмма деятельности была построена корректно с точки зрения, как синтаксиса, так и семантики. Система адекватно отобразила логику процесса авторизации, используя полный спектр необходимых элементов нотации UML — включая узлы «начало/конец потока», блоки «решения» с охватывающими условиями, последовательность «действий» и направленные потоки управления. Данный эксперимент демонстрирует, что предоставление релевантного визуального контекста существенно повышает точность и адекватность генерации, позволяя ИИ эффективнее интерпретировать предметную область и строить формальные модели, соответствующие стандартам.

Описанная выше методика применения AI с выводом результатов в виде кода *PlantUML* и его отображением в виде UML-диаграмм может успешно применяться на этапах анализа и проектирования ПО. Она позволяет сократить время разработки, найти нетривиальные архитектурные решения в диалоге «человек – ИИ», обеспечить документирование и хранение принятых архитектурных решений в удобном виде текстового кода, в противовес файлам графических форматов.

Однако, на данном этапе развития систем ИИ, полная автоматизация процессов анализа и проектирования ПО не представляется возможными. Следует также учитывать, что уровень исполнения UML-диаграмм разных типов с помощью ИИ может существенно различаться. Например, в описанных выше примерах наряду с достаточно высоким уровнем построения диаграмм классов и диаграмм последовательности присутствует диаграмма вариантов использования, не соответствующая исходному процессу и нотации языка UML.

Эффективность генерации моделей зависит от типа диаграммы: ИИ демонстрирует высокую результативность при создании диаграмм классов и последовательностей, приемлемую для диаграмм деятельности и низкую для диаграмм вариантов использования. В среднем, генерируемые модели соответствуют уровню junior-специалиста, предоставляя качественную основу для дальнейшей доработки и экономия времени аналитиков и разработчиков. Кроме того, различные ИИ-системы предлагают альтернативные модели по одному и тому же запросу, что открывает возможности для архитекторов рассматривать разнообразные варианты проектных решений. Текстовое представление моделей посредством *PlantUML* позволяет удобно хранить архитектурные решения в виде текстового кода, что является преимуществом для систем контроля версий и командной работы.

Таким образом, несмотря на существующие ограничения, системы ИИ уже сегодня являются мощным вспомогательным инструментом в арсенале бизнес-аналитиков и software-архитекторов. Они не заменяют экспертов, но значительно усиливают их возможности, беря на себя рутинные операции по первичному моделированию. Дальнейшее развитие моделей, целенаправленно обученных на корпусах текстовых требований и соответствующих им UML-моделей, позволит преодолеть выявленные недостатки и повысить качество автоматизированного проектирования. Актуальность данной темы для конференции "Непрерывное профессиональное образование лиц с особыми потребностями" подчеркивает важность применения современных технологий в образовательных практиках, направленных на поддержку различных категорий специалистов.

Полученные результаты исследования имеет смысл интегрировать в образовательный процесс, для совершенствования программы подготовки специалистов и предоставления им современных инструментов для анализа и проектирования программного обеспечения. Это не только повысит уровень обучаемости, но и подготовит будущих профессионалов к вызовам, связанным с растущей сложностью и динамичностью разработки ПО.

Литература

1. Буч Г. Язык UML. Руководство пользователя [Текст] / Г. Буч, Дж. Рамбо, А. Джекобсон. — 2-е изд. — М.: Вильямс, 2007. — 496 с. — ISBN 978-5-8459-1085-9.
2. *PlantUML Language Reference Guide (Version 1.2025.0)*: Справочное руководство по языку

PlantUML [Электронный ресурс]. — URL: https://pdf.plantuml.net/PlantUML_Language_Guide_ru.pdf (дата обращения: 15.10.2023).

3. Резуник Л. Подход к созданию сервиса генерации программного кода мобильных приложений с использованием больших языковых моделей [Текст] / Л. Резуник, Д. В. Александров // ИТ-Стандарт. — 2024. — № 4(41). — С. 34-41. — EDN EKMBXI.

GENERATION OF UML MODELS USING ARTIFICIAL INTELLIGENCE IN THE CONTEXT OF IT EDUCATION DEVELOPMENT

Kireev N.B., Kashnikava I.V.

Institute of Information Technologies, BSUIR, Minsk, Republic of Belarus

The article explores the application of AI for generating UML models in IT education. The study analyzes the conversion of textual descriptions into formal models and evaluates the compliance of the results with professional standards.

Keywords: artificial intelligence, UML modeling, IT education, design automation, educational technologies.