

УДК: 004.89

**Худайберидева Г. Б., магистр, ассистент кафедры
«Информатика и информационные технологии»**

Московский Политехнический Университет,

Россия, г. Москва

**Кожухов Д. А., магистр, ассистент кафедры
«Информатика и информационные технологии»**

Московский Политехнический Университет,

Россия, г. Москва

**Пименкова А. А., студент-бакалавр кафедры
«Информатика и информационные технологии»**

Московский Политехнический Университет,

Россия, г. Москва

КОНТЕКСТНО-ОСОЗНАЮЩИЕ ИИ-ПАРТНЕРЫ ДЛЯ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ДОКУМЕНТАЦИИ С УЧЕТОМ БИЗНЕС-ЛОГИКИ

Аннотация: Статья посвящена исследованию концепции контекстно-осознающих искусственных интеллектуальных систем, предназначенных для генерации комплексной документации, интегрирующей технические аспекты программного обеспечения с бизнес-логикой и стратегическими целями организации. Анализируются ограничения существующих инструментов автоматизированной генерации документации, фокусирующихся преимущественно на технической составляющей. Предлагается модель ИИ-партнера, способного к глубокому пониманию контекста требований, целевой аудитории и бизнес-процессов. Ключевым аспектом концепции является двунаправленная интеграция: способность системы не только анализировать программный код, но и формировать рекомендации по оптимизации требований на основе выявленных несоответствий. Рассматриваются фундаментальные вызовы, связанные с необходимостью сложного синтеза знаний из

областей бизнес-аналитики и программной инженерии. Формулируются направления для дальнейших исследований в данной неисследованной области.

***Ключевые слова:** автоматическая генерация документации, искусственный интеллект, бизнес-логика, контекстное понимание, двунаправленная интеграция, программная инженерия, бизнес-аналитика, техническая документация, бизнес-документация, ИИ-партнер.*

Khudaiberideva G. B.

**master and department assistant at the department of
"Computer Science and Information Technology"
Moscow Polytechnic University
Moscow, Russia**

Kozhukhov D. A.

**master and department assistant at the department of
"Computer Science and Information Technology"
Moscow Polytechnic University
Moscow, Russia**

Pimenkova A. A.

**bachelor's student at the department of
"Computer Science and Information Technology"
Moscow Polytechnic University
Moscow, Russia**

CONTEXT-AWARE AI PARTNERS FOR AUTOMATIC DOCUMENTATION GENERATION BASED ON BUSINESS LOGIC

***Annotation:** The article is devoted to the study of the concept of context-aware artificial intelligent systems designed to generate complex documentation that integrates technical aspects of software with business logic and strategic goals of the organization. The limitations of existing automated documentation generation tools, focusing mainly on the*

technical component, are analyzed. A model of an AI partner capable of a deep understanding of the context of requirements, target audience and business processes is proposed. A key aspect of the concept is bidirectional integration: the ability of the system not only to analyze the program code, but also to form recommendations for optimizing requirements based on identified inconsistencies. The fundamental challenges associated with the need for a complex synthesis of knowledge from the fields of business intelligence and software engineering are considered. Directions for further research in this unexplored area are formulated.

***Keywords:** automatic documentation generation, artificial intelligence, business logic, contextual understanding, bidirectional integration, software engineering, business analytics, technical documentation, business documentation, AI partner.*

Введение

Актуальность проблемы генерации актуальной, полной и согласованной документации в процессе разработки программного обеспечения не вызывает сомнений. Документация служит критически важным артефактом для разработчиков, тестировщиков, технической поддержки, конечных пользователей и менеджмента [1]. Традиционные подходы к созданию документации, особенно в части, касающейся бизнес-логики, требований и архитектурных решений, характеризуются высокой трудоемкостью, подверженностью ошибкам и быстрым устареванием по мере эволюции кодовой базы [2]. Существующие инструменты автоматической генерации технической документации, такие как средства извлечения комментариев (например, Javadoc, Doxygen), статического анализа кода или шаблонизированной отчетности, демонстрируют определенную эффективность в описании структурных элементов программы, сигнатур методов и базовых потоков данных [3]. Однако их принципиальное ограничение заключается в неспособности осмыслить генерируемое содержание в более широком контексте бизнес-целей, требований заинтересованных сторон и специфики предметной области [4]. Возникает существенный разрыв между технической реализацией и ее обоснованием с точки зрения бизнес-ценности. Данная проблема

обуславливает необходимость разработки принципиально новых подходов, основанных на применении искусственного интеллекта с расширенными возможностями контекстного понимания.

Анализ ограничений существующих подходов

Современные генеративные модели, применяемые для ассистирования в разработке, включая генерацию кода и комментариев, демонстрируют значительный прогресс [5]. Тем не менее, их применение для создания полноценной документации, особенно выходящей за рамки чисто технических аспектов, сопряжено с рядом фундаментальных проблем. Основной недостаток заключается в поверхностной обработке контекста [6]. Существующие системы, как правило, оперируют в рамках локального контекста фрагмента кода или ограниченного набора файлов, не обладая доступом к целостной картине бизнес-процессов, стратегическим документам организации, специфике целевой аудитории для различных типов документации или истории принятия проектных решений [7]. Генерация ограничивается преимущественно синтаксическим уровнем, воспроизводя структуру кода без глубокого проникновения в семантику бизнес-операций, которые данный код реализует [8]. Следствием этого является документация, технически корректная, но лишенная бизнес-смысла, не отражающая "почему" было реализовано именно так, и не связанная явно с исходными требованиями [9]. Кроме того, текущие инструменты функционируют преимущественно в режиме однонаправленной трансляции: из кода в документацию, без механизмов обратного влияния на уточнение или совершенствование самих требований на основе выявленных в процессе анализа несоответствий или оптимизационных возможностей [10].

Концепция контекстно-осознающего ИИ-партнера

В качестве ответа на выявленные ограничения предлагается концепция контекстно-осознающего ИИ-партнера (КОИИП) для генерации документации. Данная концепция предполагает создание интеллектуальной системы, способной к синтезу информации из разнородных источников, выходящих далеко за пределы кодовой базы [11]. Ключевыми источниками контекста для КОИИП должны выступать: репозитории требований (например, пользовательские истории, use cases, BPMN-диаграммы), стратегические документы компании (видение, миссия, бизнес-цели), документация по архитектуре предприятия, профили заинтересованных сторон, исторические данные о коммуникациях и принятых решениях, а также специфические шаблоны и гайдлайны по документированию, принятые в организации [12]. Фундаментальной характеристикой КОИИП является способность к многоуровневому пониманию контекста: от технической реализации отдельных модулей до бизнес-процессов, в которые они интегрированы, и стратегических задач, которые они обслуживают [13]. Это позволяет системе адаптировать стиль, уровень детализации и содержание генерируемой документации под конкретную целевую аудиторию – разработчиков, бизнес-аналитиков, конечных пользователей или руководство [14]. Например, документация для разработчиков будет акцентировать технические детали и связи между компонентами, в то время как отчет для бизнес-заинтересованных лиц должен четко отображать соответствие функциональности исходным требованиям и ее вклад в достижение бизнес-показателей.

Инновационный аспект двунаправленной интеграции

Принципиальным отличием и инновационным аспектом концепции КОИИП является реализация механизма двунаправленной интеграции [15]. В отличие от пассивных инструментов генерации, КОИИП предполагает активную роль системы в совершенствовании самого процесса разработки.

Анализируя код в сопоставлении с исходными требованиями и бизнес-контекстом, система способна выявлять различные категории несоответствий: функциональные отклонения реализации от заявленных требований, потенциальные оптимизации бизнес-процессов, реализованные в коде, но не отраженные в документации по требованиям, избыточные или дублирующиеся функции, противоречащие принципам эффективности, а также потенциальные риски для бизнеса, связанные с конкретными техническими решениями [16]. На основе выявленных инсайтов КОИИП генерирует не только документацию, но и аргументированные рекомендации по актуализации требований, оптимизации бизнес-логики или пересмотру архитектурных решений [17]. Таким образом, система становится не просто генератором текстов, а активным участником цикла обратной связи между разработкой, документированием и бизнес-анализом, способствуя повышению согласованности и качества как программного продукта, так и его описания.

Фундаментальные вызовы и неисследованность области

Реализация концепции контекстно-осознающего ИИ-партнера сопряжена с рядом сложных научно-технических вызовов, обуславливающих неисследованность данной области. Первый вызов заключается в необходимости разработки моделей и методов для глубокого семантического анализа и связывания информации из принципиально разных доменов: формализованных спецификаций программного кода и часто неструктурированных или полуструктурированных бизнес-артефактов (требований, стратегий, диаграмм процессов) [18]. Требуется создание онтологий или иных механизмов представления знаний, способных установить семантические связи между элементами кода, бизнес-правилами, целями и ролями заинтересованных сторон [19]. Второй вызов связан с достижением

подлинного понимания системой бизнес-контекста. Это подразумевает не только извлечение фактов, но и способность к интерпретации стратегических приоритетов, пониманию специфики предметной области, учету организационной культуры и неявных знаний [20]. Современные генеративные модели, несмотря на их мощь, демонстрируют ограничения в устойчивом поддержании сложного контекста и глубине рассуждений, необходимых для такой интерпретации [21]. Третий вызов – реализация эффективной двунаправленной интеграции. Разработка алгоритмов для выявления не просто ошибок в коде, а именно несоответствий между технической реализацией и бизнес-интенциями, а также для формулирования практически ценных и обоснованных рекомендаций по оптимизации бизнес-процессов на основе технического анализа, представляет собой нетривиальную задачу на стыке искусственного интеллекта, программной инженерии и бизнес-информатики [22]. Сложность синтеза знаний из этих дисциплин на уровне, необходимом для функционирования КОИИП, является основным барьером для развития данного направления.

Заключение

Представленная концепция контекстно-осознающего ИИ-партнера для генерации документации предлагает новый подход к решению проблемы создания комплексных, релевантных и ценных документационных артефактов. Ключевым отличием от существующих инструментов является фокус на глубоком понимании системой не только технической реализации, но и бизнес-логики, стратегических целей и контекста использования. Центральным инновационным элементом выступает принцип двунаправленной интеграции, при котором ИИ-система не только пассивно генерирует документацию на основе кода, но и активно анализирует соответствие реализации бизнес-требованиям, выявляя несоответствия и генерируя рекомендации по оптимизации как

самой документации, так и лежащих в ее основе требований и процессов. Анализ выявил существенные ограничения современных генеративных моделей и инструментов в части контекстного понимания и интеграции бизнес-знаний. Фундаментальные трудности реализации концепции КОИИП лежат в плоскости сложного синтеза методов семантического анализа, представления знаний из разнородных доменов, глубокого понимания бизнес-контекста и разработки алгоритмов для формирования обратной связи, влияющей на бизнес-процессы. Данные вызовы определяют неисследованность области и формируют актуальные направления для дальнейших научных изысканий в области искусственного интеллекта, программной инженерии и бизнес-аналитики. Успешная разработка и внедрение подобных систем потенциально способны значительно повысить эффективность процессов разработки и сопровождения программного обеспечения за счет обеспечения постоянной согласованности между технической реализацией, документацией и бизнес-целями организации.

СПИСОК ЛИТЕРАТУРЫ:

1. Rahman M.M., Roy S.K. Automatic generation of software documentation using source code summarization // Journal of Systems and Software. 2022. Т. 191. С. 111384.
2. Forward A., Lethbridge T.C. The relevance of software documentation, tools and technologies: a survey // Труды конференции ACM SIGDOC'02. 2002. С. 26–33.
3. Moreno L., Aponte J., Sridhara G. и др. Automatic generation of natural language summaries for Java classes // Труды конференции IEEE ICPC. 2013. С. 23–32.
4. Castro J.W., Acuna S.T., Juristo N. Integrating the production of software documentation into software processes // J. UCS. 2010. Т. 16, № 17. С. 2476–2498.

5. Chen M., Tworek J., Jun H. и др. Evaluating large language models trained on code // arXiv preprint arXiv:2107.03374. 2021.
6. Johnson D.D., Roemmele M., Etzioni O. Execution-guided neural program generation // Труды воркшопа ICLR. 2018.
7. Sein M.K., Henfridsson O., Puro S. и др. Action design research // MIS Quarterly. 2011. Т. 35, № 1. С. 37–56.
8. Gupta R., Pal S., Kanade A., Shevade S. DeepFix: Fixing common C language errors by deep learning // Труды конференции AAAI. 2017. Т. 31, № 1.
9. Cleland-Huang J., Chang C.K., Christensen M. Event-based traceability for managing evolutionary change // IEEE TSE. 2003. Т. 29, № 9. С. 796–810.
10. Lee S., Yang Y., Ryu Y. и др. Learning to generate code comments from class hierarchies // Труды конференции IEEE SANER. 2021. С. 422–433.
11. Vasilescu B., Casalnuovo C., Devanbu P. Recovering clear, natural identifiers from obfuscated JS names // Труды конференции ESEC/FSE. 2017. С. 683–693.
12. Glinz M. On non-functional requirements // Труды конференции IEEE RE. 2007. С. 21–26.
13. Leite J.C.S.P., Freeman P.A. Requirements validation through viewpoint resolution // IEEE TSE. 1991. Т. 17, № 12. С. 1253–1269.
14. Pohl K. Requirements Engineering: Fundamentals, Principles, and Techniques. Springer, 2010.
15. Zave P., Jackson M. Four dark corners of requirements engineering // ACM TOSEM. 1997. Т. 6, № 1. С. 1–30.
16. Van Lamsweerde A. Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley, 2009.
17. Jackson M. Problem Frames: Analysing and Structuring Software Development Problems. Addison-Wesley, 2001.