



Примена алата заснованих на вештачкој интелигенцији у реинжењерингу информационих система

Application of Artificial Intelligence-Based Tools in Information Systems Reengineering

Анастасија Савић, Факултет техничких наука, Нови Сад

Студијски програм – СОФТВЕРСКО ИНЖЕЊЕРСТВО И ИНФОРМАЦИОНЕ ТЕХНОЛОГИЈЕ

Кратак садржај – У раду је приказан процес реинжењеринга болничког информационог система уз примену алата заснованих на вештачкој интелигенцији (AI). Истраживање обухвата примену AI асистента (Google Gemini) и AI агента (OpenAI Codex) у фазама анализе изворног кода, генерисања дијаграма обједињеног језика за моделовање (UML), реинжењеринга базе података и модернизације апликационог слоја. Приказани резултати показују могућности примене AI алата у реконструкцији документације, моделовању система и подршци у фазама реинжењеринга, уз неопходну верификацију добијених резултата од стране стручњака.

Кључне речи: вештачка интелигенција, реинжењеринг информационих система, AI асистенти, AI агенти

Abstract – This paper presents the process of re-engineering a hospital information system using artificial intelligence (AI)-based tools. The research involves the application of an AI assistant (Google Gemini) and an AI agent (OpenAI Codex) in the phases of source code analysis, unified modeling language (UML) model generation, database reengineering, and application layer modernization. The presented results demonstrate the potential of AI tools for documentation reconstruction, system modeling, and supporting various phases of the re-engineering process, while emphasizing the necessity of expert verification of the generated results.

Keywords: artificial intelligence, information systems re-engineering, AI assistants, AI agents

НАПОМЕНА: Овај рад проистекао је из мастер рада чији ментор је била др Славица Кордић, ван. проф.

1. УВОД

Реинжењеринг информационих система представља сложен процес модернизације постојећих софтверских решења са циљем побољшања њихове одрживости, ефикасности и прилагодљивости савременим

технолошким захтевима [1], [2]. Традиционални приступи реинжењеринга често захтевају значајан ангажман стручњака и дуг временски период за анализу постојећег система, реконструкцију документације и трансформацију софтверских компоненти.

Развој вештачке интелигенције (AI) и великих језичких модела омогућила је примену нових приступа у софтверском инжењерству [3]. Савремени AI алати помажу програмерима при писању програмског кода, способни су да разумеју целокупне базе кода, предлажу архитектонске трансформације и самостално решавају комплексне инжењерске проблеме [4].

Циљ овог рада је испитивање могућности примене AI алата у процесу реинжењеринга на примеру болничког информационог система [5], у фазама анализе система, реинжењеринга базе података и модернизације апликационог слоја.

2. ПРЕГЛЕД АКТУЕЛНОГ СТАЊА У ОБЛАСТИ ИСТРАЖИВАЊА

Савремени AI алати засновани на великим језичким моделима све чешће се користе у процесима софтверског инжењерства [3], [4]. У пракси се могу разликовати AI асистенти и AI агенти.

AI асистенти, као што је Google Gemini [6], пружају интерактивну подршку кориснику кроз анализу захтева, генерисање препорука и креирање садржаја на основу задатих упита.

Са друге стране, AI агенти, као што је OpenAI Codex [7], поседују већи степен аутономије и способни су за анализу сложенијих софтверских система, идентификацију њихових компоненти и извршавање задатака у више корака.

Због различитих карактеристика ових приступа, за потребе истраживања у оквиру овог рада, коришћена су оба типа алата. Gemini је примењен за формулисање смерница током моделовања система, миграције базе података и анализе резултата статичке анализе кода, док је Codex коришћен за анализу изворног кода, генерисање модела, трансформацију базе података и рефакторисање апликационог слоја.

3. АНАЛИЗА ПОСТОЈЕЋЕГ СИСТЕМА

Као студија случаја коришћен је постојећи информациони систем за подршку раду болнице [5]. Систем је реализован као вишеслојна веб апликација која користи *React* на клијентској страни, *ASP.NET Core Web API* на серверској страни и *PostgreSQL* систем за управљање базом података.

3.1. Анализа кода

Пре почетка процеса реинжењеринга било је неопходно извршити детаљну анализу постојећег решења. У традиционалном приступу ова фаза подразумева ручно проучавање документације и анализу великог броја изворних датотека, што представља временски захтеван процес. У овом истраживању испитана је могућност примене *AI* агента *Codex* ради аутоматизације ових активности.

Анализом је утврђено да је постојећи болнички информациони систем [5] организован као слојевита архитектура са јасно издвојеним слојевима корисничког интерфејса, пословне логике и приступа подацима. *AI* алат је успешно идентификовао кључне доменске ентитете као што су пацијент, доктор, медицински картон, преглед, операција и лек. Поред идентификације ентитета, издвојене су и њихове међусобне зависности, као и главни токови података кроз систем.

Слика 1 представља апстрактни модел анализираног болничког информационог система [5], добијен применом *AI* алата *Codex*, који приказује његове главне компоненте и њихове међусобне односе и омогућава лакше разумевање његове архитектуре и организације.



Слика 1. Апстрактни модел система генерисан применом *AI* алата *Codex*

Анализом изворног кода добијен је приказ основне структуре болничког информационог система [5], укључујући кључне компоненте, доменске ентитете и њихове међусобне зависности. Добијени резултати послужили су као основа за даље активности моделовања и реинжењеринга.

3.2. Моделовање система

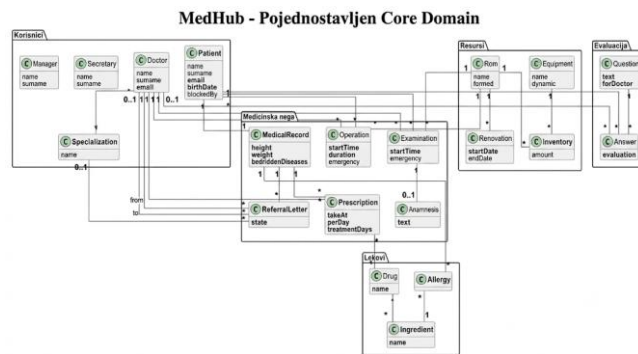
Поред анализе архитектуре, *AI* алати су коришћени за генерисање дијаграма обједињеног језика за моделовање (*UML*) [8].

Пре почетка генерисања *UML* дијаграма коришћен је алат *Gemini* ради избора најпогоднијег приступа за аутоматизовано моделовање система. На основу анализе захтева и карактеристика расположивих алата предложена је употреба *PlantUML* технологије [9] као посредног формата за опис *UML* модела. Након тога је *Codex* генерисао *PlantUML* скрипте на основу изворног кода, док је за њихову визуелизацију и даљу обраду коришћен алат *draw.io*.

Као резултат овог процеса генерисани су дијаграм класа, дијаграм секвенци и дијаграм случајева коришћења. Генерисани *UML* дијаграми успешно су

приказали структуру и функционалност система из различитих перспектива.

Исправност генерисаног класног дијаграма, приказаног на слици 2, проверена је поређењем са постојећом имплементационом шемом базе података. Утврђено је да су кључни ентитети идентификовани у *UML* моделу присутни као одговарајуће табеле у бази података, док приказане релације одговарају везама реализованим путем страних кључева. Добијени резултати указују на висок степен подударности између генерисаног *UML* модела и имплементираних структуре базе података и доменског модела у изворном коду.



Слика 2. Дијаграма класа генерисан путем *AI* алата *Codex*

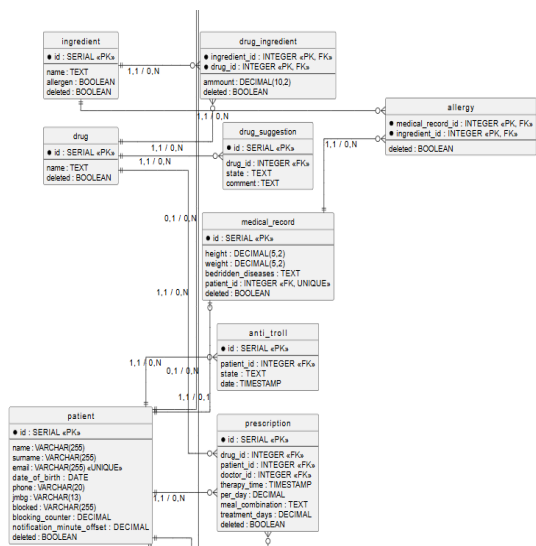
4. РЕИНЖЕЊЕРИНГ БАЗЕ ПОДАТАКА

Реинжењеринг базе података спроведен је применом фаза реверзног инжењеринга, реструктурирања и инжењеринга према напред [1], [2]. Циљ овог процеса био је разумевање постојеће структуре базе података, идентификација могућности за њено унапређење и припрема за миграцију у друго окружење за управљање базама података.

Полазну основу представљала је *PostgreSQL* база података болничког информационог система [5], која садржи податке о пацијентима, лекарима, медицинским картонима, прегледима, операцијама, терапијама и другим аспектима рада здравствене установе. У циљу анализе структуре базе података извршен је извоз метаподатака из алата *pgAdmin*, који су затим прослеђени *AI* алату *Codex* ради екстракције шеме базе података.

На основу табела, примарних и страних кључева, као и *NOT NULL* и *UNIQUE* ограничења, идентификованих анализом достављених метаподатака, *AI* алат је реконструисао логичке ентитете, релације између њих док су максимални и минимални кардиналитети изведени на основу структуре кључева и претходно идентификованих ограничења. Добијени резултати представљени су ентитети–релације (*ER*) моделом. Део генерисаног *ER* дијаграма приказан је на слици 3.

Поређењем са ручно креираним *ER* моделом утврђено је да су логички ентитети, релације и максимални кардиналитети исправно реконструисани. Додатно, генерисани *ER* модел је омогућио приказ минималних кардиналитета дефинисаних на основу *NOT NULL* ограничења.



Слика 3. Део генерисаног ER дијаграма помоћу AI алата *Codex*

Након реконструкције постојеће шеме спроведена је фаза реструктурирања базе података. *AI* алат је анализирао постојећи модел и предложио измене које обухватају увођење додатних *UNIQUE* и *CHECK* ограничења, семантичке корекције атрибута и стандардизацију домена података уједначавањем типова података, ради отклањања неконзистентности у моделу.

У фази инжењеринга према напред извршена је трансформација реструктуриране *PostgreSQL* шеме у *Oracle Database 26ai* шему. У овој фази посебна пажња посвећена је типовима података и њиховом прилагођавању циљном *Oracle* окружењу.

Након тога, за потребе планирања миграције података, коришћен је *AI* алат *Gemini*, који је предложио више приступа миграције из *PostgreSQL* у *Oracle* окружење, при чему је као најпогодније решење издвојио примену алата *Oracle SQL Developer Migration Wizard*. Наведени алат је затим коришћен за извршавање процеса миграције података, након успостављања конекције са изворном *PostgreSQL* и циљном *Oracle* базом података. Исправност миграције потврђена је поређењем броја записа у свим табелама и успешним функционисањем апликације над мигрираном базом података.

Иако претходно генерисана *Oracle* шема базе података није била неопходна за одабран начин миграције података, она је задржана како би се проверило да ли *Codex* алат може успешно да конвертује *PostgreSQL* шему у *Oracle* шему базе података, јер је код појединих приступа миграцији, као што је извоз података у формат вредности раздвојених зарезом (*CSV*), претходно генерисана шема циљне базе података неопходна.

Током истраживања уочено је да *AI* алати могу значајно убрзати анализу и разумевање сложених база података, као и процес њихове трансформације и миграције. Ипак, поједини генерисани резултати захтевали су додатну ручну проверу, пре свега због разлика у синтакси *Oracle* система и трансформације *PostgreSQL* специфичних типова података у њихове *Oracle* еквиваленте.

5. РЕИНЖЕЊЕРИНГ АПЛИКАЦИОНОГ СЛОЈА

У оквиру модернизације апликационог слоја испитивана је примена *AI* алата у рефакторисању серверског и клијентског дела система за подршку раду болнице [5]. Циљ истраживања био је да се процени у којој мери алати засновани на вештачкој интелигенцији могу подржати анализу постојећег кода, отклањање уочених недостатака и модернизацију софтверске архитектуре уз очување постојећих функционалности система.

5.1. Реструктурирање серверског слоја

У истраживању је коришћен *AI* алат *Codex* за анализу изворног кода, идентификацију проблема и генерисање измена, док је *SonarQube* коришћен за објективну процену квалитета кода пре и након реструктурирања.

Пре почетка refaktorisања спроведена је статичка анализа комплетног серверског слоја. Анализа је обухватила контролере, сервисе и компоненте за приступ подацима и указала на постојање безбедносних рањивости, багова, проблема у структури кода и одсуство аутоматизованих тестова. У циљу лакше интерпретације резултата *SonarQube* анализе коришћен је и *Gemini*, који је предложио начин извоза и организације резултата у облику pogodnom за даљу обраду од стране *Codex* агента.

На основу добијених резултата *Codex* је извршио рефакторисање кода кроз више итерација. Извршене су исправке логичких грешака у пословној логици, уклоњене су хардкодоване лозинке из конфигурационих датотека и имплементирано је чување осетљивих података у *.env* окружењу. Поред тога, извршена је реорганизација појединих делова кода, уклањање сувишних конструкција и побољшање структуре метода које су доприносиле повећању техничког дуга.

У завршној фази *Codex* је генерисао јединичне тестове за кључне функционалности система и извршио иницијалну валидацију њихове исправности. Након поновљене *SonarQube* анализе уочено је смањење броја уочених проблема и повећање тестне покривености.

Резултати показују да *AI* алати могу успешно подржати процес рефакторисања и унапређења квалитета серверског кода, али да је за потврду исправности предложених измена неопходна додатна верификација од стране програмера.

5.2. Модернизација клијентског слоја

На клијентској страни истраживање је обухватило два аспекта модернизације: миграцију постојеће *React* апликације са *JavaScript* на *TypeScript* технологију и унапређење корисничког интерфејса и корисничког искуства.

У првој фази *AI* алат *Codex* је анализирао постојећу *JavaScript* имплементацију и извршио њену постепену миграцију на *TypeScript*. Током процеса конвертоване су постојеће компоненте, уведена статичка типизација и дефинисани типови за податке, форме и компоненте корисничког интерфејса. Након сваке фазе миграције спроведена је аутоматизована провера исправности

кроз *TypeScript* анализу и процес изградње (*build*) апликације. Резултати су показали да је могуће аутоматизовати значајан део процеса миграције уз очување постојеће функционалности система.

Након техничке модернизације кода, *Codex* је коришћен за анализу корисничког интерфејса и идентификацију могућности за његово унапређење. Алат је препознао проблеме који се односе на конзистентност навигације, распоред елемената интерфејса и респонзивност апликације на уређајима различитих димензија. На основу анализе предложена су и имплементирана побољшања која су обухватила стандардизацију навигационих компоненти, унапређење распореда елемената применом *flex layout* механизма и побољшање визуелних индикатора интеракције.

Посебно значајан резултат односи се на унапређење респонзивности корисничког интерфејса, при чему су отклоњени проблеми са преклапањем елемената и ограниченом доступношћу појединих контрола. Иако су уочена значајна побољшања, анализа је показала да поједине странице и даље захтевају додатне итерације оптимизације.

6. ЗАКЉУЧАК

Основни научни допринос рада огледа се у евалуацији могућности примене *AI* асистената и *AI* агената у различитим фазама реинжењеринга информационих система и сагледавању њихове улоге у модернизацији наслеђених софтверских решења.

Резултати истраживања показали су да алати *Google Gemini* и *OpenAI Codex* могу да убрзају активности анализе, реконструкције модела, трансформације базе података и модернизације апликационог слоја над постојећим болничким информационим системом [5]. Посебна вредност ових алата огледа се у способности аутоматизације дела инжењерских активности и генерисању предлога који могу послужити као подршка током процеса реинжењеринга.

Истовремено, истраживање је показало да квалитет добијених резултата зависи од расположивог контекста и квалитета улазних података. Иако су *AI* алати показали висок ниво корисности, поједини резултати захтевали су додатну ручну проверу и валидацију, што указује да улога софтверског инжењера остаје кључна у доношењу коначних одлука и обезбеђивању исправности решења.

Добијени резултати указују да *AI* алати представљају значајну подршку у процесу реинжењеринга информационих система, али не и потпуну замену за инжењерско знање и експертизу.

Будућа истраживања могу бити усмерена ка анализи напреднијих *AI* платформи и њиховој примени у сложенијим сценаријима модернизације софтверских система.

7. ЛИТЕРАТУРА

[1] E. J. Chikofsky and J. H. Cross, "Reverse engineering and design recovery: A taxonomy", *IEEE software*, vol. 7, no. 1, pp. 13–17, 1990.

[2] C. Wagner, *Model-Driven Software Migration: A Methodology: Reengineering, Recovery and Modernization of Legacy Systems*. Springer Science & Business Media, 2014.

[3] M. Adawadkar, "The Evolution of Generative and Agentic AI: From Rule-Based Systems to Autonomous Intelligence", *American Journal of Technology Advancement*, vol. 2, no. 11, 2025.

[4] A. Ndiaye, X. Taraj, B. Robeznik, G. Herzog, C. Gerber, and H. Purwins, *Generative AI in software engineering: transforming the software development process*. DFKI & Accenture, 2025.

[5] A. Savić, *Softverski paket za podršku rada bolnice*, B.Sc. thesis, Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia, 2024.

[6] Google Gemini documentation <https://ai.google.dev/> (pristupljeno u martu 2026.)

[7] OpenAI Codex documentation <https://openai.com/index/openai-codex/> (pristupljeno u martu 2026.)

[8] G. Reggio, M. Leotta, F. Ricca, and D. Clerissi, "What are the used UML diagrams? A Preliminary Survey", in *EESSMod@ MoDELS*, pp. 3–12, 2013.

[9] S. Carruthers, A. Thomas, L. Kaufman-Willis, and A. Wang, "Growing an accessible and inclusive systems design course with plantuml", in *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, pp. 249–255, 2023.

Кратка биографија:



Анастасија Савић рођена је у Зрењанину 2002. год. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – Софтверско инжењерство и информационе технологије одбранила је 2026.год

Контакт:

anastasijas557@gmail.com