

ИЗВЕШТАЈ О ОЦЕНИ ДОКТОРСKE ДИСЕРТАЦИЈЕ

I ПОДАЦИ О КОМИСИЈИ		
1. Датум и орган који је именовао комисију: Решење декана Факултета техничких наука у Новом Саду, бр. 012-40/186-2026 од 16.07.2026. 2. Састав комисије у складу са <i>Правилима докторских студија Универзитета у Новом Саду</i>:		
1.	Дејановић др Игор	редовни професор
	презиме и име	звање
	Универзитет у Новом Саду, Факултет техничких наука	Примењене рачунарске науке и информатика, 16.05.2022.
	установа у којој је запослен-а	ужа научна област и датум избора
		председник
2.	Бабарогић др Слађан	редовни професор
	презиме и име	звање
	Универзитет у Београду, Факултет организационих наука	Информациони системи, 17.11.2021.
	установа у којој је запослен-а	ужа научна област и датум избора
		члан
3.	Милошевић др Немања	доцент
	презиме и име	звање
	Универзитет у Новом Саду, Природно-математички факултет	Рачунарске науке, 12.7.2022.
	установа у којој је запослен-а	ужа научна област и датум избора
		члан
4.	Вуковић др Жељко	доцент
	презиме и име	звање
	Универзитет у Новом Саду, Факултет техничких наука	Примењене рачунарске науке и информатика, 01.07.2025.
	установа у којој је запослен-а	ужа научна област и датум избора
		члан
5.	Милосављевић др Гордана	редовни професор
	презиме и име	звање
	Универзитет у Новом Саду, Факултет техничких наука	Примењене рачунарске науке и информатика, 21.10.2020.
	установа у којој је запослен-а	ужа научна област и датум избора
		ментор
		функција у комисији

II ПОДАЦИ О КАНДИДАТУ
1. Име, име једног родитеља, презиме: Ненад, Перо, Тодоровић 2. Датум рођења, општина, држава: 18.12.1993, Суботица, Србија 3. Назив факултета, назив претходно завршеног нивоа студија и стечени стручни/академски назив: Факултет техничких наука, Универзитет у Новом Саду; мастер академске студије, студијски програм Рачунарство и аутоматика, мастер инжењер електротехнике и рачунарства 4. Година уписа на докторске студије и назив студијског програма докторских студија: 2017, Рачунарство и аутоматика (докторске академске студије), Факултет техничких наука, Универзитет у Новом Саду
III НАСЛОВ ДОКТОРСКЕ ДИСЕРТАЦИЈЕ:
Аутоматизација развоја API-базираних генератора рударењем идиома из изворног кода (Automating the Development of API-Based Generators Using Source Code Idioms Mining)
IV ПРЕГЛЕД ДОКТОРСКЕ ДИСЕРТАЦИЈЕ:
Навести кратак садржај са назнаком броја страница, поглавља, слика, схема, графикона и сл. Докторска дисертација Ненада Тодоровића написана је на енглеском језику, у формату Б5. Укупно има 149 страна, од чега су 104 нумерисане стране. Основни текст организован је у девет поглавља на 92 стране, након којих следе списак литературе са 120 референци (стр. 93–104) и биографија кандидата. Дисертација садржи 35 слика, 3 табеле и 3 алгоритама. Уводни део чине насловна страна, кључна документацијска информација на српском и енглеском језику (Образац 5а), проширени резиме на српском језику (17 страна), апстракт на енглеском језику, садржај, списак алгоритама, списак табела, списак слика и списак скраћеница. На крају је приложен план третмана података. Садржај дисертације по поглављима је следећи: 1. Увод (Introduction) 2. Теоријске основе (Theoretical Foundations) 3. Преглед литературе и сродних истраживања (Background and Related Work) 4. Преглед SAGED приступа (Overview of SAGED) 5. Закључивање идиома из кода (Code Idiom Inference) 6. RoseLib 7. Примена SAGED приступа (Application of SAGED) 8. Евалуација (Evaluation) 9. Закључак (Conclusions)
V ВРЕДНОВАЊЕ ПОЈЕДИНИХ ДЕЛОВА ДОКТОРСКЕ ДИСЕРТАЦИЈЕ:
Дисертација садржи све елементе прописане важећим прописима. Девет поглавља логички су распоређена и добро структурирана. Прва три поглавља постављају проблем истраживања, теоријске основе и стање у области. Поглавља 4–7 представљају предложени приступ, његове математичке основе и имплементацију. Поглавље 8 представља евалуацију, а поглавље 9 сумира закључке, доприносе и правце даљих истраживања. Истраживање је спроведено по методологији истраживања у науци о дизајну (Design Science Research Methodology – DSRM) у варијанти Пеферса и сарадника, а пресликавање корака DSRM на поједине делове дисертације је експлицитно наведено. Прво поглавље уводи контекст истраживања, предмет истраживања, потребу за истраживањем, циљеве, примењене методе и могућност примене очекиваних резултата. Кандидат позиционира истраживање у оквиру софтверског инжењерства вођеног моделима (Model-Driven Software Engineering – MDSE), у коме се модели третирају као примарни артефакти развоја, а извршиве апликације изводе се из њих трансформацијама, при чему се за генерисање кода користе трансформације модела у текст (M2T). Кључни практични проблем је постепено увођење MDSE у постојеће софтверске пројекте, при којем ручне измене у коду морају бити очуване између циклуса генерисања кода, без наметања нове архитектуре пројекту. API-базирани генератори (ABG) идентификовани су као средство које такав прелазак омогућава. Пошто ефикасност ABG-а директно

зависи од нивоа апстракције његовог API-ја, а како су ранија истраживања показала да је развој генератора са таквим ABG-овима знатно спорији него са обрађивачима шаблона, формулисан је следећи циљ истраживања: дефинисати приступ и пројектовати имплементацију за полуаутоматско откривање понављајућих идиома у коду који су добри кандидати за генерисање кода, као и за полуаутоматско креирање API-ја дефинисаног у терминима тих идиома. Потреба за истраживањем образложена је чињеницом да у литератури недостаје скалабилна, потпуно специфицирана имплементација рударења идиома кода, као и било какав рад који се тиче аутоматизације развоја ABG-ова. На основу изнетог, формулисане су општа хипотеза X0 и изведене хипотезе X1–X4. Поглавље се завршава прегледом структуре дисертације.

Друго поглавље представља теоријске основе двеју области на којима почива допринос: MDSE и рударења идиома кода. MDSE је позициониран у односу на сродне методе (MBSE, MDSD, MDA) и објашњени су његови кључни концепти који се користе у наставку дисертације. Различити начини за генерисање кода су детаљно анализирани, као и механизми за интеграцију ручно писаног и генерисаног кода. Идиоми су дефинисани као мали, често понављајући фрагменти кода са једном семантичком улогом, који могу садржати метапроменљиве и који одговарају често понављаним подстаблима синтаксних стабала. У наставку се детаљно дискутује зашто су погодни као основа за проширење ABG-ова и зашто су алтернативне технике (детекција клонова, рударење API-ја, рударење честих подстабала) мање подесне за постављени циљ. Поглавље је јасно написано, добро илустровано (осам слика) и пружа појмовни апарат који је потребан за наредна поглавља.

Треће поглавље даје преглед сродних истраживања: користи од ABG-ова, изазови њихове употребе, рударење идиома из кода и аутоматизација развоја MDSE алата. Кандидат показује како ABG-ови омогућавају неометан MDSD развојни процес, ослањајући се на раније искуство у индустрији своје истраживачке групе. Приступ Аламаниса и Сатона, који закључује пробабилистичку граматiku супституције стабала (PTSG) из програмског кода, идентификован је као најсавременија техника рударења идиома, а документован је и недостатак описа у литератури како применити Type-Based MCMC на овај домен. Радови о аутоматизацији развоја MDSE алата (детекција клонова за извођење QVT шаблона и DSL-ова, учење правила превођења по примеру, LSTM мреже и велики језички модели), анализирани су и показано је да ниједан од њих не разматра аутоматизацију развоја ABG-ова. Предност разумљивих и лако измењивих граматика супституције стабала над неуронским моделима „црне кутије“ убедљиво је аргументована. Преглед литературе је актуелан, релевантан и критички вреднован и јасно утврђује истраживачки јаз којим се дисертација бави.

Четврто поглавље представља централни методолошки допринос дисертације: SAGED (Semi-automatic API-based Generators Development), нови приступ за полуаутоматски развој API-базираних генератора. Приступ се састоји од две фазе и четири корака: фаза 1, откривање концепата, обухвата корак S1 (закључивање идиома кода) и корак S2 (курирање), а фаза 2, проширење ABG-а, обухвата корак S3 (генерисање API метода) и корак S4 (њихову интеграцију у ABG). Избор непараметарске Бајесове PTSD методе, апроксимираних прилагођених Type-Based MCMC, добро је мотивисан: реч је о ненадгледаном учењу које не захтева аотиран код, метода је заснована на статистичком моделу и стога закључује значајније идиоми од метода заснованих искључиво на учесталости, а закључени фрагменти нужно поштују граматiku језика. Кандидат детаљно разматра предуслове и ограничења развијене методе (улазни код мора показивати понављање, синтаксна исправност идиома не подразумева његову потпуност, а закључени идиоми теже да буду плитки) и из њих изводи неопходност корака курирања, у коме корисник прегледа, спаја и мења идиоми. Поглавље такође специфицира пет захтева које резултујући ABG мора да задовољи (прецизна селекција, инкременталне измене, очување синтаксне исправности, раздвајање ручно писаних и генерисаних API метода и свеобухватна подршка за измену кода) и објашњава како се флексибилна ABG основа проширује генерисаним методама. Приступ је дефинисан независно од конкретног програмског језика и ABG-а.

Пето поглавље даје математичке основе методе закључивања и садржи главни научни допринос дисертације у овој области. Полазећи од контекстно слободних граматика и њихове пробабилистичке варијанте, кандидат уводи граматике супституције стабала и PTSD, чији фрагменти кодирају нелокални контекст и чији нетерминални листови природно представљају метапроменљиве, и своди проблем проналажења идиома на учење PTSD која проширује познату граматiku и добро објашњава корпус за обучавање. Објашњено је Бајесово закључивање са Дирихлеовим процесом као априорном расподелом над фрагментима, заједно са базном расподелом. Пошто се апостериорна расподела не

може тачно израчунати, она се апроксимира Гибсовим узоркивачем, који за сваки чвор одлучује да ли је корен новог фрагмента. Пошто Гибсов узоркивач који обрађује чвор по чвор није скалабилан на велике базе кода, кандидат преузима Type-Based MCMC Лианга и сарадника (карактеристичан по томе што чворове истог типа групише у блок и заједно их ажурира) и детаљно га тумачи и разрађује за домен рударења идиома кода. Кључни оригинални допринос је поједностављење које омогућава да се чува само делимична информација о окружењу чвора уместо целог вектора промена бројача, чиме се значајно смањују вођење евиденције и утрошак меморије. Поглавље се завршава предобрадом парсираних синтаксних стабала (бинаризација, повезивање зависних конструкција и фиксирање изабраних врста чворова као корена или не-корена), чиме се закључивање усмерава ка идиомима које ABG основа може да прихвати. Сви алгоритми су формално дефинисани, објашњени на пратећим примерима и илустровани, а излагање је ригорозно и прецизно.

Шесто поглавље описује RoseLib, ABG отвореног кода за језик C# развијен у ранијим истраживањима групе као слој изнад Roslyn Syntax Tree API-ја, који у дисертацији служи као ABG основа. Поглавље објашњава механизме обиласка и композиције, језик CSPath за везе следљивости, инспирисан језиком XPath и имплементиран метајезиком textX, и провере интегритета ради откривања ручних измена између два циклуса генерисања. Поглавље заокружује пример генератора који се ослања на CPath и провере интегритета. Опис је потпун, технички прецизан и довољан за разумевање како RoseLib задовољава захтеве формулисане у четвртом поглављу.

Седмо поглавље представља имплементацију приступа SAGED за програмски језик C#. Корак S1 реализују Предобрађивач, који припрема улазне датотеке, и Инференцер, који језички независно језгро закључивања проширује подршком за C# засновано на Roslyn-у. Језгро закључивања детаљно је описано заједно са алгоритмом обучавања. Корак S2 подржавају Визуализатор, који приказује пронађене идиооме, њихову учесталост и везе у коду, и Руковалац, који омогућава спајање идиома уз очување синтаксне исправности. Фазу 2 реализује Трансформатор са својом базом знања, који из изабраног идиома генерише декларацију и тело API методе и додаје је у парцијалну класу одговарајућег композера, чиме је обезбеђено захтевано раздвајање ручно писаних и генерисаних метода. Показано је да тако проширен RoseLib задовољава свих пет захтева из четвртог поглавља. На крају поглавља представљени су алати за кориснике: додаток за Visual Studio Code, Визуализатор идиома и алат за мерење StatEval који се користи у евалуацији. Имплементација је усклађена са приступом описаним у четвртом поглављу, њене компоненте покривене су јединичним тестовима, а развијени софтвер доступан је у репозиторијумима отвореног кода, што обезбеђује поновљивост истраживања.

Осмо поглавље евалуира приступ кроз три истраживачка питања: да ли фаза откривања концепата производи релевантне идиооме кода (RQ1), да ли је та фаза скалабилна (RQ2) и да ли су API методе изведене приступом SAGED ефикасније за употребу од постојећих API метода ABG-а (RQ3). Квантитативна евалуација мерила је прецизност, покривеност, просечну дужину идиома и скалабилност на четрнаест C# пројеката величине од око 4,5 хиљаде до преко 600 хиљада линија кода, обједињених у јавно доступан Mendeley Data скуп. У просеку је 40% закључених идиома пронађено у тест корпусима, идиоми су објаснили 39% чворова тест корпуса, а време обучавања расло је линеарно са величином базе кода. Квалитативна евалуација са спољном валидацијом потврдила је релевантност закључених идиома уз ограничен обим ручног курирања. Уочена ограничења методе су детаљно наведена. Евалуација ефикасности показала је да је за развој истог MDSE алата проширеном верзијом библиотеке RoseLib написано 24% мање линија кода уз око 20% краће време развоја, а поређење са ранијим индустријским истраживањем групе показало је у просеку 15,5 наспрам 20 човек-часова по генератору. Сва три истраживачка питања потврдно су одговорена, чиме су потврђене хипотезе X1–X4 и, посредно, X0, а претње валидности идентификоване су заједно са мерама за њихово адресирање. Евалуација је методолошки утемељена, свеобухватна и одговарајуће осмишљена за проверу постављених хипотеза.

Завршно поглавље сумира резултате, групише доприносе дисертације на методолошке, техничке и емпиријске и даје табеларни преглед потврде сваке хипотезе заједно са местом у дисертацији на којем се налазе докази. Разматране су импликације за истраживање и праксу и назначени правци даљег развоја. Закључци су јасно образложени, усклађени са представљеним резултатима, а предложени правци даљих истраживања су реални и добро утемељени.

Списак литературе (120 референци) је релевантан за тему дисертације и обухвата класичне и савремене изворе из области MDSE и машинског учења над изворним кодом, укључујући и најновије

радове о великим језичким моделима. Проширени резиме на српском језику верно одражава садржај дисертације.
VI СПИСАК НАУЧНИХ И СТРУЧНИХ РАДОВА КОЈИ СУ ОБЈАВЉЕНИ ИЛИ ПРИХВАЋЕНИ ЗА ОБЈАВЉИВАЊЕ НА ОСНОВУ РЕЗУЛТАТА ИСТРАЖИВАЊА У ОКВИРУ РАДА НА ДОКТОРСКОЈ ДИСЕРТАЦИЈИ: Таксативно навести називе радова, где и када су објављени. Прво навести најмање један рад објављен или прихваћен за објављивање у складу са <i>Правилима докторских студија Универзитета у Новом Саду</i> који је повезан са садржајем докторске дисертације. У случају радова прихваћених за објављивање, таксативно навести називе радова, где и када ће бити објављени и приложити потврду уредника часописа о томе. Рад у врхунском међународном часопису (M21): 1. Todorović, N., Lukić, A., Todorović, N., Dragaš, B., Milosavljević, G.: Automating the development of API-based generators using code idioms mining. <i>Software and Systems Modeling</i>, 25(1), 2026, pp. 81–108, (објављено онлајн 6. 6. 2025), https://doi.org/10.1007/s10270-025-01296-z Рад у истакнутом међународном часопису (M22): 1. Dragaš, B., Todorović, N., Rajačić, T., Milosavljević, G., Vuković, Ž.: A Novel Approach to Integration of Manual Changes in Generated Code: SeamlessMDD. <i>Journal of Web Engineering</i>, 24(4), 2025, pp. 499–528. https://doi.org/10.13052/jwe1540-9589.2442 Саопштења са међународних скупова штампана у целини (M33): 1. Dragaš, B., Todorović, N., Rajačić, T., Milosavljević, G.: SeamlessMDD: Framework for Seamless Integration of Generated and Hand-Written Code. In: <i>Web Engineering, ICWE 2024, Lecture Notes in Computer Science</i>, vol. 14629, Springer, 2024, pp. 163–177. https://doi.org/10.1007/978-3-031-62362-2 2. Todorović, N., Dragaš, B., Milosavljević, G.: Supporting Integrative Code Generation with Traceability Links and Code Fragment Integrity Checks. In: Trajanovic, M., Filipovic, N., Zdravkovic, M. (eds.) <i>Disruptive Information Technologies for a Smart Society, ICIST 2023, Lecture Notes in Networks and Systems</i>, vol. 872, Springer, 2024, pp. 490–501. https://doi.org/10.1007/978-3-031-50755-7_46 3. Todorović, N., Lukić, A., Zoranović, B., Vadera, R., Vuković, Ž., Stoja, S.: RoseLib: A Library for Simplifying .NET Compiler Platform Usage. In: <i>ICIST 2018 Proceedings</i>, Vol. 1, 2018, pp. 216–221. ISBN 978-86-85525-22-3 Поред наведених, рад кандидата Ненада Тодоровића објављен у часопису <i>Software and Systems Modeling</i>, био је изабран и презентован у journal-first програму на престижној конференцији: 28th International Conference on Model Driven Engineering Languages and Systems (MODELS 2025), Grand Rapids, MI, USA, October 5–10, 2025: https://conf.researchr.org/details/models-2025/models-2025-journal-first/8/Automating-the-Development-of-API-Based-Generators-Using-Code-Idioms-Mining
VII ЗАКЉУЧЦИ ОДНОСНО РЕЗУЛТАТИ ИСТРАЖИВАЊА: Докторска дисертација бави се аутоматизацијом развоја API-базираних генератора кода ради лакшег увођења софтверског инжењерства вођеног моделима у животни циклус постојећих софтверских пројеката. Истраживање је структурирано према методологији истраживања у науци о дизајну и резултовало је следећим закључцима и резултатима: Методолошки резултати 1. Дефинисан је SAGED, нови приступ за полуаутоматски развој ABG-ова који се састоји од две фазе (откривање концепата и проширење ABG-a) и четири корака, заједно са захтевима које резултујући ABG мора да задовољи. Према сазнању комисије, ниједно раније истраживање није се бавило аутоматизацијом развоја ABG-ова нити употребом идиома кода у ту сврху. 2. Систематизована је литература о закључивању идиома кода. Дисертација детаљно објашњава како се метода Type-Based MCMC практично примењује на рударење идиома кода, уз поједностављење дефиниције типа чвора које значајно смањује утршак меморије. Технички резултати 1. Развијено је језички независно језгро закључивања отвореног кода, проширено за језик C# и интегрисано у развојно окружење, а показано је да RoseLib задовољава све постављене захтеве као

ABG основа која се може аутоматизовано проширивати новим API методама. Успешна имплементација система конструктивно потврђује хипотезу X3.
Емпиријски резултати 1. Квантитативна евалуација на четрнаест пројеката различите величине показала је да је у просеку 40% закључених идиома пронађено у тест корпусима, да идиоми објашњавају 39% чворова тест корпуса и да време обучавања расте линеарно са величином базе кода, чиме су потврђене хипотезе X1 и X2. 2. Квалитативна евалуација са спољном валидацијом показала је да курирање захтева ограничен напор, да је само 12% идентификованих идиома захтевало благе измене и да су искусни спољни програмери препознали семантику идиома без контекста. Уочена ограничења методе документована су као правци даљег рада. 3. Евалуација ефикасности показала је да је проширени ABG захтевао 24% мање линија кода и око 20% мање времена развоја него основни, а у поређењу са ранијим индустријским искуством 15,5 наспрам 20 човек-часова по генератору, чиме је потврђена хипотеза X4. Узете заједно, потврђене хипотезе X1–X4 и успешна имплементација система потврђују општу хипотезу X0: могуће је креирати решење за аутоматизацију развоја API-базираних генератора кода прилагођених техничком домену задатом базом кода. Дисертација тако показује да се употребљивост ABG-ова битно повећава када се њихов API дефинише у терминима концепата вишег нивоа изведених из идиома постојећег кода.
VIII ОЦЕНА НАЧИНА ПРИКАЗА И ТУМАЧЕЊА РЕЗУЛТАТА ИСТРАЖИВАЊА: Експлицитно навести позитивну или негативну оцену начина приказа и тумачења резултата истраживања.
Докторска дисертација урађена је систематично, уз примену ригорозног научног приступа. Истраживачки проблем, хипотезе, циљеви и методе јасно су формулисани у уводном поглављу, а наредна поглавља следе логичан ток од теоријских основа и стања у области, преко дефинисања приступа и његових математичких основа, до имплементације и евалуације. Математички апарат методе закључивања изложен је прецизно, уз формалне дефиниције, алгоритме и пратеће примере који захтевну тему чине приступачном. Архитектура и имплементација илустроване су јасним дијаграмима и примерима кода, а резултати истраживања представљени су нумерички, табеларно и графички и објективно су протумачени, уз отворено навођење уочених ограничења методе. Закључци представљају концизну и кохерентну синтезу резултата и непосредно су поткрепљени изнетим доказима, што потврђује да су хипотезе критички проверене и да су циљеви дисертације остварени. Рад кандидата у потпуности одговара дефинисаној теми и наслову, како по обиму, тако и по методологији, а резултати пружају солидну основу за даља истраживања у овој области. Дисертација је проверена у софтверу за детекцију плагијаризма <i>iThenticate</i>. На основу наведеног, комисија позитивно оцењује начин приказа и тумачења резултата истраживања.
IX КОНАЧНА ОЦЕНА ДОКТОРСКЕ ДИСЕРТАЦИЈЕ: Експлицитно навести да ли дисертација јесте или није написана у складу са наведеним образложењем, као и да ли она садржи или не садржи све битне елементе. Дати јасне, прецизне и концизне одговоре на 3. и 4. питање. Навести нумеричке податке о резултатима провере оригиналности рада и дати текстуално образложење.
1. Да ли је дисертација написана у складу са образложењем наведеним у пријави теме? Докторска дисертација је написана у складу са образложењем наведеним у пријави теме.
2. Да ли дисертација садржи све битне елементе? Дисертација садржи све битне елементе.
3. По чему је дисертација оригиналан допринос науци? Докторска дисертација Ненада Тодоровића представља оригиналан научни допринос у пресеку софтверског инжењерства вођеног моделима и машинског учења над програмским кодом. Њени оригинални доприноси су:

• SAGED: нови приступ за полуаутоматски развој API-базираних генератора, независан од језика и ABG-а, који закључује идиоме кода из нелабелираног изворног кода, подржава њихово куирање и аутоматизује креирање API метода ABG-а. Аутоматизација развоја ABG-ова помоћу идиома кода до сада није обрађивана у литератури. • Детаљно објашњење и прилагођавање методе Type-Based MCMC домену рударења идиома кода, укључујући поједностављење дефиниције типа чвора које значајно смањује утрошак меморије и чини методу применљивом на базе кода од више стотина хиљада линија. • Језички независно језгро закључивања отвореног кода, прошириво на произвољне програмске језике, у потпуности имплементирано за језик C#, са обезбеђеним механизмом проширивања ABG основе генерисаним API методама и алатима за кориснике. • Емпиријски резултати: референтне вредности метрика закључивања на четрнаест пројеката, објављене заједно са јавним скупом података, доказ линеарне скалабилности, спољна валидација релевантности закључених идиома и показивање да API методе изведене приступом SAGED смањују обим кода за 24% и време развоја за око 20%. Ови резултати представљају научни допринос у примени непараметарских Бајесових метода на изворни код и у аутоматизацији развоја MDSE алата, као и допринос од практичне вредности за увођење MDSE у постојеће, зреле софтверске пројекте, у виду имплементираних алата отвореног кода.	4. Који су недостаци дисертације и какав је њихов утицај на резултат истраживања? Комисија није уочила недостатке дисертације који би утицали на резултате истраживања. 5. Образложење резултата провере оригиналности рада (нумерички и наративно): Провера оригиналности докторске дисертације спроведена је софтвером iThenticate у библиотеци Факултета техничких наука. Проверен је текст од 24912 речи, откривено је 200 поклапања из 43 извора и утврђен је индекс сличности од 62%. Од тога 53% (13652 речи) потиче од рада кандидата објављеног у часопису Software and Systems Modeling (Todorović и сарадници, 2026) и 4% (1143 речи) од рада кандидата са конференције (Todorović и сарадници, 2024) који се тиче библиотеке RoseLib. Оба рада чине интегралне делове дисертације. Преосталих 5% распоређено је на 41 извор са појединачним преклапањем испод 1% и односи се на стандардну терминологију области (Model-Driven Software Engineering, code generation, Model-to-Text трансформације) и опште формулације. На основу изнете анализе, комисија закључује да је поднета докторска дисертација оригинално дело кандидата Ненада Тодоровића.
---	--

X	ПРЕДЛОГ:
	На основу наведеног, комисија предлаже:
	<u>а) да се докторска дисертација прихвати, а кандидату одобри одбрана;</u> б) да се докторска дисертација врати кандидату на дораду (да се допуни односно измени); в) да се докторска дисертација одбије.

Место и датум: Нови Сад, 14.09.2026.

1. др Игор Дејановић, ред. проф.

_____, председник

2. др Слађан Бабарогић, ред. проф.

_____, члан

3. др Немања Милошевић, доцент

_____, члан

4. др Жељко Вуковић, доцент

_____, члан

5. др Гордана Милосављевић, ред. проф.

_____, ментор

НАПОМЕНА: Члан комисије који не жели да потпише извештај јер се не слаже са мишљењем већине чланова комисије, дужан је да унесе у извештај образложење односно разлоге због којих не жели да потпише извештај и да исти потпише.

DOCTORAL DISSERTATION EVALUATION REPORT

I COMMITTEE INFORMATION		
1. Date and body that assigned the committee: Decision of the Dean of the Faculty of Technical Sciences in Novi Sad, No. 012-40/186-2026 of 16.07.2026.		
2. Committee members' information in accordance with the <i>Rules of Doctoral Studies of the University of Novi Sad</i>:		
1.	Dejanović Igor, PhD	Full Professor
	Surname and name	Title
	University of Novi Sad, Faculty of Technical Sciences	Applied Computer Science and Informatics, 16.05.2022.
	Institution of employment	Scientific subfield and date of election
		Chair
2.	Babarogić Slađan, PhD	Full Professor
	Surname and name	Title
	University of Belgrade, Faculty of Organizational Sciences	Information Systems, 17.11.2021.
	Institution of employment	Scientific subfield and date of election
		Member
3.	Milošević Nemanja, PhD	Assistant Professor
	Surname and name	Title
	University of Novi Sad, Faculty of Sciences	Computer Science, 12.7.2022.
	Institution of employment	Scientific subfield and date of election
		Member
4.	Vuković Željko, PhD	Assistant Professor
	Surname and name	Title
	University of Novi Sad, Faculty of Technical Sciences	Applied Computer Science and Informatics, 01.07.2025.
	Institution of employment	Scientific subfield and date of election
		Member
5.	Milosavljević Gordana, PhD	Full Professor
	Surname and name	Title
	University of Novi Sad, Faculty of Technical Sciences	Applied Computer Science and Informatics, 21.10.2020.
	Institution of employment	Scientific subfield and date of election
		Mentor
		Committee role
II INFORMATION ABOUT THE CANDIDATE		
1. Name, name of one parent, surname:		

Nenad, Pero, Todorović 2. Date, municipality and country of birth: 18.12.1993, Subotica, Serbia 3. Name of the academic institution, previously completed academic program and the title acquired: Faculty of Technical Sciences, University of Novi Sad; Master Academic Studies, study programme Computing and Control Engineering, Master of Science in Electrical and Computer Engineering 4. Year of enrolment into doctoral studies and name of the doctoral studies programme: 2017, Computing and Control Engineering (Doctoral Academic Studies), Faculty of Technical Sciences, University of Novi Sad
III TITLE OF THE DOCTORAL DISSERTATION: Automating the Development of API-Based Generators Using Source Code Idioms Mining (Аутоматизација развоја API-базираних генератора рударењем идиома из изворног кода)
IV OVERVIEW OF THE DOCTORAL DISSERTATION: Provide a brief overview indicating the number of pages, chapters, figures, diagrams, graphs, etc. The doctoral dissertation of Nenad Todorović is written in English and presented in B5 format. It comprises 149 pages, of which 104 are numbered pages. The main text is organized into nine chapters across 92 pages. Following the main text are a bibliography with 120 references (pages 93–104) and the candidate's biography. The dissertation includes 35 figures, 3 tables, and 3 algorithms. The front matter comprises the title page, Key Word Documentation in Serbian and English (Form 5a), an extended summary in Serbian (17 pages), an English abstract, the table of contents, a list of algorithms, a list of tables, a list of figures, and a list of acronyms. A data management plan is appended at the end. The dissertation's chapters are: 1. Introduction 2. Theoretical Foundations 3. Background and Related Work 4. Overview of SAGED 5. Code Idiom Inference 6. RoseLib 7. Application of SAGED 8. Evaluation 9. Conclusions
V EVALUATION OF INDIVIDUAL DISSERTATION CHAPTERS: The dissertation adheres to all prescribed regulations. Its nine chapters are logically arranged and well-structured. The first three chapters establish the research problem, theoretical foundations, and state of the art. Chapters 4–7 detail the proposed approach, its mathematical foundations, and implementation. Chapter 8 presents the evaluation, while Chapter 9 summarizes the conclusions, contributions, and future research directions. The research followed the Design Science Research Methodology (DSRM) variant by Peffers et al., with an explicit mapping of DSRM steps to the dissertation's sections. The first chapter introduces the research context, subject, motivation, objectives, methods, and potential applications of the expected results. The candidate positions the research within Model-Driven Software Engineering (MDSE), where models are primary development artifacts from which executable applications are derived via transformations, specifically model-to-text (M2T) transformations for code generation. A key challenge is the gradual integration of MDSE into existing software projects, requiring the preservation of manual code changes across code-generation cycles without imposing new architectural constraints. API-based generators (ABGs) are identified as the mechanism that makes such transitions possible. However, since ABG efficiency depends on API abstraction levels, and prior research indicates slower generator development with ABGs compared to template engines, the following objective was formulated: to define an approach and design an implementation for the semi-automatic discovery of recurring code idioms suitable for code generation, and for the semi-automatic creation of an API defined by these idioms. The research is justified by the fact that literature lacks a scalable, fully specified code idiom mining implementation and any work on automating ABG development. Based on this, general hypothesis H0 and derived hypotheses H1–H4 are formulated. The chapter concludes with an outline of the dissertation's structure.

The second chapter lays the theoretical foundations of the two fields the dissertation builds upon: MDSE and code idiom mining. MDSE is positioned with respect to related methods (MBSE, MDSD, MDA). The key concepts of MDSE used in the remainder of the dissertation are explained. Various code generation approaches and mechanisms for integrating handwritten and generated code are analyzed in detail. Idioms are defined as small, frequently recurring code fragments with a single semantic role, potentially containing metavariables, and corresponding to frequently repeated subtrees in syntax trees. The chapter then thoroughly discusses why idioms are suitable for extending ABGs and why alternative techniques (clone detection, API mining, frequent subtree mining) are less appropriate for the stated objective. The chapter is clearly written, well-illustrated (eight figures), and provides the conceptual apparatus necessary for subsequent chapters.

The third chapter reviews related research: the benefits of ABGs, the challenges of their use, code idiom mining, and the automation of MDSE tool development. Drawing on his research group's prior industrial experience, the candidate demonstrates how ABGs facilitate a seamless MDSD workflow. The approach by Allamanis and Sutton, which infers a probabilistic tree substitution grammar (PTSG) from program code, is identified as the state-of-the-art idiom mining technique, and the absence in the literature of a description for applying Type-Based MCMC to this domain is documented. Work on automating MDSE tool development (clone detection for deriving QVT templates and DSLs, by-example learning of translation rules, LSTM networks, and large language models) is analyzed, revealing that none address ABG development automation. The advantage of comprehensible and easily modifiable tree substitution grammars over black-box neural models is convincingly argued. The literature review is up to date, relevant, and critically evaluated, and clearly establishes the research gap addressed by the dissertation.

The fourth chapter presents the dissertation's central methodological contribution: SAGED (Semi-automatic API-based Generators Development), a novel approach for the semi-automatic development of API-based generators. The approach comprises two phases and four steps: Phase 1, Concept Discovery, includes step S1 (code idiom inference) and step S2 (curation), while Phase 2, ABG Extension, includes step S3 (generation of API methods) and step S4 (their integration into the ABG). The choice of the non-parametric Bayesian PTS method, approximated by an adapted Type-Based MCMC, is well-motivated: it is unsupervised learning requiring no annotated code, based on a statistical model that infers more meaningful idioms than purely frequency-based methods, and the inferred fragments inherently respect the language's grammar. The candidate discusses in detail the preconditions and limitations of the developed method (input code must exhibit repetitiveness, syntactic correctness of an idiom does not imply its completeness, and inferred idioms tend to be shallow), from which the necessity of the curation step (where the user reviews, joins, and modifies idioms) is derived.

The chapter also specifies five requirements for the resulting ABG (precise selection, incremental changes, preservation of syntactic correctness, separation of handwritten and generated API methods, and comprehensive support for code modification) and explains how a flexible ABG basis is extended with generated methods. The approach is defined independently of any specific programming language and ABG.

The fifth chapter provides the mathematical foundations of the inference method and contains the dissertation's main scientific contribution in this area. Starting with context-free grammars and their probabilistic variant, the candidate introduces tree substitution grammars and PTSs, whose fragments encode non-local context and whose non-terminal leaves naturally represent metavariables. The idiom-finding problem is reduced to learning a PTS that extends a known grammar and explains the training corpus well. Bayesian inference with a Dirichlet process prior over fragments is explained, along with the base distribution and the Gibbs sampler that approximates the intractable posterior by sampling, for each node, whether the node is the root of a new fragment. Since a node-by-node Gibbs sampler does not scale to large codebases, the candidate adopts Type-Based MCMC by Liang et al., which groups nodes of the same type into a block and updates them jointly, and interprets and elaborates it in detail for the domain of code idiom mining. The key original contribution is a simplification that allows only partial information about a node's surroundings to be stored instead of the entire vector of count changes, significantly reducing bookkeeping and memory consumption. The chapter concludes with the preprocessing of parsed syntax trees (binarization, linking of dependent constructs, and fixing of selected node kinds as roots or non-roots), which guides the inference toward idioms acceptable by an ABG basis. All algorithms are formally defined, explained with running examples, illustrated, and presented rigorously and precisely.

The sixth chapter describes RoseLib, an open-source ABG for C# developed in the group's earlier research as a layer above the Roslyn Syntax Tree API, serving as the ABG basis in this dissertation. The chapter explains its traversal and composition mechanisms, the CPath language for traceability links

(inspired by XPath and implemented with the textX meta-language), and integrity checks for detecting manual changes between two generation cycles. An example of a generator relying on CSPATH and integrity checks rounds off the chapter. The description is complete, technically precise, and sufficient for understanding how RoseLib satisfies the requirements formulated in Chapter 4.

The seventh chapter presents the implementation of the SAGED approach for the C# programming language. Step S1 is realized by the Preprocessor, which prepares input files, and the Inferencer, which extends the language-independent inference core with Roslyn-based C# support. The inference core is described in detail along with the training algorithm. Step S2 is supported by the Visualizer, which displays found idioms, their frequency, and their links in the code, and the Handler, which allows idioms to be joined while preserving syntactic correctness. Phase 2 is realized by the Transformer with its Knowledge Base, which generates the declaration and body of an API method from a selected idiom and adds it to a partial class of the corresponding composer, thereby ensuring the required separation of handwritten and generated methods. It is demonstrated that RoseLib, extended in this manner, satisfies all five requirements from Chapter 4. At the end of the chapter, end-user tools are presented: a Visual Studio Code extension, the idiom Visualizer, and the StatEval measurement tool used in the evaluation. The implementation is consistent with the approach described in Chapter 4, its components are covered by unit tests, and the developed software is available in open-source repositories, ensuring research reproducibility.

The eighth chapter evaluates the approach through three research questions: whether the Concept Discovery phase produces relevant code idioms (RQ1), whether that phase is scalable (RQ2), and whether the API methods derived with the SAGED approach are more efficient to use than existing ABG API methods (RQ3). The quantitative evaluation measured precision, coverage, average idiom length, and scalability on fourteen C# projects ranging from approximately 4.5 thousand to over 600 thousand lines of code, combined into a publicly available Mendeley Data dataset. On average, 40% of the inferred idioms were found in the test corpora, the idioms explained 39% of the test-corpus nodes, and training time grew linearly with codebase size. The qualitative evaluation with external validation confirmed the relevance of the inferred idioms with a limited amount of manual curation. The observed limitations of the method are described in detail. The efficiency evaluation showed that developing the same MDSE tool with the extended version of the RoseLib library required 24% fewer lines of code and about 20% less development time, while a comparison with the group's earlier industrial research showed an average of 15.5 versus 20 person-hours per generator. All three research questions were answered affirmatively, thereby confirming hypotheses H1–H4 and, indirectly, H0. Threats to validity were identified along with measures taken to address them. The evaluation is methodologically sound, comprehensive, and appropriately designed to test the stated hypotheses.

The final chapter summarizes the results, categorizes the dissertation's contributions into methodological, technical, and empirical ones, and provides a tabular overview of each hypothesis's validation, including where the evidence is presented in the dissertation. Implications for research and practice are discussed, and directions for future development are outlined. The conclusions are clearly explained and consistent with the presented results, and the proposed future research directions are realistic and well-founded.

The bibliography (120 references) is relevant to the dissertation's topic, encompassing classical and contemporary sources from MDSE and machine learning on source code, including recent work on large language models. The extended summary in Serbian faithfully reflects the dissertation's content.

VI LIST OF SCIENTIFIC AND PROFESSIONAL MANUSCRIPTS PUBLISHED OR ACCEPTED FOR PUBLICATION, WHICH ARE BASED ON THE RESULTS OF THE DOCTORAL RESEARCH:

List the titles of the manuscripts, where and when they were published. First list at least one manuscript published or accepted for publication in accordance with the *Rules of Doctoral Studies of the University of Novi Sad* which is related to the content of the doctoral dissertation. In the case of manuscripts accepted for publication, list their titles, where and when they will be published, and enclose the confirmation of the journal editor thereof.

Paper in a leading international journal (M21):

1. **Todorović, N., Lukić, A., Todorović, N., Dragaš, B., Milosavljević, G.:** Automating the development of API-based generators using code idioms mining. *Software and Systems Modeling*, 25(1), 2026, pp. 81–108 (published online 6 June 2025), <https://doi.org/10.1007/s10270-025-01296-z>

Paper in an outstanding international journal (M22):

1. Dragaš, B., **Todorović, N.**, Rajačić, T., Milosavljević, G., Vuković, Ž.: A Novel Approach to Integration of Manual Changes in Generated Code: SeamlessMDD. *Journal of Web Engineering*, 24(4), 2025, pp. 499–528. <https://doi.org/10.13052/jwe1540-9589.2442>

Papers presented at international conferences and published in full in proceedings (M33):

1. Dragaš, B., **Todorović, N.**, Rajačić, T., Milosavljević, G.: SeamlessMDD: Framework for Seamless Integration of Generated and Hand-Written Code. In: *Web Engineering, ICWE 2024, Lecture Notes in Computer Science*, vol. 14629, Springer, 2024, pp. 163–177. <https://doi.org/10.1007/978-3-031-62362-2>
2. **Todorović, N.**, Dragaš, B., Milosavljević, G.: Supporting Integrative Code Generation with Traceability Links and Code Fragment Integrity Checks. In: Trajanovic, M., Filipovic, N., Zdravkovic, M. (eds.) *Disruptive Information Technologies for a Smart Society, ICIST 2023, Lecture Notes in Networks and Systems*, vol. 872, Springer, 2024, pp. 490–501. https://doi.org/10.1007/978-3-031-50755-7_46
3. **Todorović, N.**, Lukić, A., Zoranović, B., Vadera, R., Vuković, Ž., Stoja, S.: RoseLib: A Library for Simplifying .NET Compiler Platform Usage. In: *ICIST 2018 Proceedings*, Vol. 1, 2018, pp. 216–221. ISBN 978-86-85525-22-3

In addition to the above, the paper by the candidate Nenad Todorović published in *Software and Systems Modeling* was selected for presentation, and presented in the journal-first programme at the prestigious 28th International Conference on Model Driven Engineering Languages and Systems (MODELS 2025), Grand Rapids, MI, USA, October 5–10, 2025: <https://conf.researchr.org/details/models-2025/models-2025-journal-first/8/Automating-the-Development-of-API-Based-Generators-Using-Code-Idioms-Mining>

VII CONCLUSIONS AND RESEARCH RESULTS:

The doctoral dissertation addresses the automation of API-based code generator development to facilitate the integration of Model-Driven Software Engineering into existing software project lifecycles. The research, structured according to the Design Science Research Methodology, yielded the following conclusions and results:

Methodological results

1. A novel approach SAGED was defined for the semi-automatic development of ABGs. SAGED comprises two phases (Concept Discovery and ABG Extension) and four steps, along with the requirements for the resulting ABG. To the best of the Committee's knowledge, no prior research has addressed the automation of ABG development or the use of code idioms for this purpose.
2. The literature on code idiom inference was systematized. The dissertation provides detailed explanation of how to practically apply the Type-Based MCMC method to code idiom mining, including a simplified definition of a node type that significantly reduces memory consumption.

Technical Results

1. A language-independent, open-source inference core was developed, extended for the C# language, and integrated into a development environment. It was demonstrated that RoseLib satisfies all stated requirements as an ABG basis capable of automatic extension with new API methods. The successful implementation of this system constructively confirms hypothesis H3.

Empirical results

1. The quantitative evaluation across fourteen projects of varying sizes showed that, on average, 40% of the inferred idioms were found in the test corpora. These idioms explained 39% of the test-corpus nodes, and training time grew linearly with codebase size, thereby confirming hypotheses H1 and H2.
2. The qualitative evaluation with external validation indicated that curation required limited effort, only 12% of identified idioms needed minor modifications, and experienced external developers recognized idiom semantics without context. The method's observed limitations are documented as directions for future work.
3. The efficiency evaluation demonstrated that the extended ABG required 24% fewer lines of code and approximately 20% less development time than the basic version. Compared to earlier industrial experience, it required 15.5 person-hours per generator versus 20, confirming hypothesis H4.

Taken together, the confirmed hypotheses H1–H4 and the successful system implementation confirm the general hypothesis H0: it is possible to create a solution for automating the development of API-based code generators tailored to the technical domain specified by a given codebase. The dissertation thus demonstrates

that the usability of ABGs is substantially increased when their API is defined in terms of higher-level concepts derived from existing code idioms.
VIII EVALUATION OF THE METHOD OF PRESENTATION AND INTERPRETATION OF RESEARCH RESULTS: Explicitly state a positive or negative evaluation of the method of presentation and interpretation of the research results. The doctoral dissertation has been prepared systematically, applying a rigorous scientific approach. The research problem, hypotheses, objectives, and methods are clearly formulated in the introductory chapter. The subsequent chapters progress logically from theoretical foundations and the state of the art, through the definition of the approach and its mathematical foundations, to implementation and evaluation. The mathematical apparatus of the inference method is presented precisely, with formal definitions, algorithms, and running examples that make a demanding topic accessible. The architecture and implementation are illustrated with clear diagrams and code examples. Research results are presented in numerical, tabular, and graphical form and interpreted objectively, with the observed limitations of the method reported openly. The conclusions represent a concise and coherent synthesis of the results, directly supported by the presented evidence, confirming that the hypotheses were critically examined and the dissertation's objectives achieved. The candidate's work fully corresponds to the defined topic and title, both in scope and methodology, and the results provide a solid basis for further research in this field. The dissertation was checked using iThenticate plagiarism detection software. Based on the above, the Committee evaluates the manner of presentation and interpretation of the research results positively.
IX FINAL EVALUATION OF DOCTORAL DISSERTATION: Explicitly state whether the dissertation was or was not written in accordance with the stated rationale, and whether it does or does not contain all the important elements. Give clear, precise and concise answers to questions 3 and 4. Provide numerical data on the results of the originality check of the work and give a textual explanation.
1. Was the dissertation written in accordance with the previously submitted dissertation proposal? The doctoral dissertation has been written in accordance with the rationale stated in the topic application.
2. Does the dissertation contain all the important elements? The dissertation contains all the essential elements.
3. Why does this dissertation provide original contributions to science? The doctoral dissertation of Nenad Todorović represents an original scientific contribution at the intersection of Model-Driven Software Engineering and machine learning on program code. Its original contributions are: • SAGED: a novel approach for the semi-automatic development of API-based generators, independent of language and ABG, which infers code idioms from unlabelled source code, supports their curation and automates the creation of ABG API methods. The automation of ABG development by means of code idioms has not previously been addressed in the literature. • A detailed explanation and adaptation of the Type-Based MCMC method to the domain of code idiom mining, including a simplification of the definition of a node type that significantly reduces memory consumption and makes the method applicable to codebases of several hundred thousand lines. • A language-independent, open-source inference core that can be extended to any programming language, fully implemented for C#, with a mechanism for extending the ABG basis with generated API methods and tools for end users. • Empirical results: reference values of the inference metrics on fourteen projects, published together with a public dataset, evidence of linear scalability, external validation of the relevance of the inferred idioms, and a demonstration that API methods derived with the SAGED approach reduce the amount of code by 24% and the development time by about 20%. These results represent a scientific contribution to the application of non-parametric Bayesian methods to source code and to the automation of MDSE tool development, as well as a contribution of practical value for the introduction of MDSE into existing, mature software projects, in the form of the implemented open-source tools.

4. What are the shortcomings of the dissertation and what is their impact on the research results? The Committee has not identified any shortcomings of the dissertation that would affect the research results.	
5. Explanation of the results of the originality check of the work (numerically and narratively): The originality check of the doctoral dissertation was conducted using iThenticate software at the Library of the Faculty of Technical Sciences. A text of 24912 words was checked, revealing 200 matches from 43 sources, and the reported similarity index is 62%. Of this, 53% (13652 words) originates from the candidate's paper published in Software and Systems Modeling journal (Todorović et al., 2026), and 4% (1143 words) from the candidate's conference paper on the RoseLib library (Todorović et al., 2024). Both papers form an integral part of the dissertation. The remaining 5% is distributed across 41 sources with individual overlaps below 1%, and relates to standard terminology of the field (Model-Driven Software Engineering, code generation, Model-to-Text transformations) and general phrasing. Based on this analysis, the Committee concludes that the submitted doctoral dissertation is the original work of the candidate Nenad Todorović.	
X	PROPOSAL:
Based on the indicated information, the committee proposes:	
<u>a) to accept the doctoral dissertation and approve the candidate's defense;</u> b) to return the doctoral dissertation to the candidate for revision (to be supplemented or amended); c) to reject the doctoral dissertation.	

Place and date: Novi Sad, 14.09.2026.

1. Dr Igor Dejanović, Full Professor
_____, Chair of the Committee
2. Dr Slađan Babarogić, Full Professor
_____, Member
3. Dr Nemanja Milošević, Assistant Professor
_____, Member
4. Dr Željko Vuković, Assistant Professor
_____, Member
5. Dr Gordana Milosavljević, Full Professor
_____, Mentor

NOTE: A committee member who does not want to sign the report because they disagree with the majority opinion of the committee is obliged to provide an explanation or reasons why they do not want to sign the report and to sign it accordingly.