

Наставно-научном већу
Факултета организационих наука
Универзитета у Београду

Предмет: Подобност теме и кандидата Дејана Стојимировића за израду докторске дисертације

Одлуком 05-01 бр. 3/25 од 29.4.2020. године именовани смо за чланове Комисије за оцену теме и подобности кандидата Дејана Стојимировића за израду докторске дисертације и научне заснованости теме **Концепт простора имена као начин за савладавање сложености у развоју информационих система.**

На основу материјала приложеног уз Захтев кандидата, Комисија подноси следећи

ИЗВЕШТАЈ

1. Подаци о кандидату

Биографски подаци

Дејан Стојимировић је рођен 18. октобра 1987. године у Пожаревцу. Завршио је основну школу Дража Марковић – Рођа 2002. године у Смољинцу, као ђак генерације и носилац је Вукове дипломе. Након тога је уписао средњу, Економско-трговинску школу у Пожаревцу, смер финансијски техничар. Средњу школу је завршио 2006. У мају те године је освојио прво место на републичком такмичењу из пословне информатике. У јулу 2006. године уписује Факултет организационих наука, одсек за Информационе системе и технологије.

Дипломирао је октобра 2011. године са просечном оценом 8,39 и оценом 10 на дипломском раду „Реализација трансформација модела применом *Smarty* механизма за шаблоне“. Новембра 2011. године уписује мастер студије на Факултету организационих наука, одсек за информационе системе, које је завршио 12.7.2013. године са просеком 9,83 и оценом 10 на завршном раду „Предлог стандардне архитектуре *WEB* апликације на *Node.js* платформи“. У новембру 2013. године, на Факултету организационих наука, уписује докторске студије, студијски програм Информациони системи и квантитативни менаџмент – студијска група Информациони системи. На докторским студијама је положио све испите са просечном оценом 10. Од 2012. године је запослен на Факултету организационих наука као сарадник у настави, а од 2014. ради у звању асистента на Катедри за информационе системе. Учествовао је у више комерцијалних и истраживачких пројеката.

Личне референце

Образовање:

2006-2011: Дипломирани инжењер организационих наука – одсек за информационе системе, Факултет организационих наука Универзитета у Београду

2011-2013: Мастер инжењер организационих наука, Факултет организационих наука Универзитета у Београду

2013 – данас: Студент докторских студија, Факултет организационих наука Универзитета у Београду

Списак положених предмета на докторским студијама са оценама и ЕСП бодовима:

Назив предмета	Оцена	ЕСП
Развој информационих система	10 (десет)	10
Интероперабилност пословних система и апликација	10 (десет)	10
Статистика у менаџменту	10 (десет)	10
Мултиваријациона анализа	10 (десет)	10
Управљање подацима	10 (десет)	10
Структуре података и алгоритми	10 (десет)	10
Софтверске архитектуре	10 (десет)	10
Системи за управљање пословним процесима	10 (десет)	10
Откривање законитости у базама података	10 (десет)	10

Радно искуство:

2014. – данас: асистент на Катедри за информационе системе Факултета организационих наука Универзитета у Београду

2012. – 2014. – сарадник у настави на Катедри за информационе системе Факултета организационих наука Универзитета у Београду

2010. – 2012. – Студент сарадник у Лабораторији за информационе системе „др Бранислав Лазаревић“ на Факултету организационих наука Универзитета у Београду

Педагошко искуство:

- **Основне студије:** Структуре података и алгоритми, Моделовање пословних процеса, Одабрана поглавља из информационих система
- **Мастер студије:** Базе података 2, Изабрана поглавља из информационих система, Методе и алати за аутоматизацију развоја информационих система, Увод у

напредне архитектуре информационих система, Планирање развоја ИС, Интегрисана софтверска решења, Сервисно-орјентисана архитектура

Поред учешћа у настави, учествовао је и у реализацији обука:

- Реализација обуке у оквиру националног програма Преквалификација за ИТ сектор коју су организовали Канцеларија за ИТ и електронску управу Владе Републике Србије и Програм Уједињених нација за развој у Србији (УНДП) на Факултету организационих наука (2019. године)
- Реализација обуке „Развој веб апликација помоћу *PHP* програмског језика“ у Истраживачко-развојном центру Факултета организационих наука, Универзитета у Београду (2017. године)

Учешће на пројектима:

- Идејни пројекат реинжењеринга пословних процеса Агенције за безбедност саобраћаја – сектор за возаче (2019. године)
- Стратегија и развојни план за информациони систем ЈКП Водовод – Ваљево (2011. године)
- Пројектовање и имплементација информационог система за подршку рада Службе за основе студије, као и Службе за мастер и специјалистичке студије Факултета организационих наука, Универзитета у Београду (2017-2020)
- Пројектовање и имплементација система за аутоматско прегледање студенских радова на Факултету организационих наука, Универзитета у Београду (2016. године)
- Израда техничке документације за израду веб, *Android* и *iOS* платформи које би омогућавале *live stream* продукцију са екстерне камере (наочара) путем *bluetooth* конекције, уз подршку система за наплату и комуникацију између стримера и корисника (2019. године)

Публикације:

Радови објављени у зборницима међународних скупова (М30):

1. **Дејан Стојимировић**, Нина Турајлић, Синиша Нешковић, Ненад Аничих, Слађан Бабарогић, ПРИСТУПИ ЗА МОДЕЛОВАЊЕ ВИШЕСТРУКО ПОВЕЗАНИХ КАРАКТЕРИСТИКА У МОДЕЛИМА КАРАКТЕРИСТИКА, Зборник радова са конференције Sym-op-is, University of Belgrade, Faculty of Organizational Sciences, 2019, пп 238-243, ISBN: 978-86-7680-363-7, Кладово, Србија, 15. – 18. септембар, 2019 (М33)

2. Јован Парушић, Урош Шошевић, **Дејан Стојимировић**, DIGITAL DISRUPTION OF AGRICULTURE v4.0, Зборник радова са конференције SymOrg2018, University of Belgrade, Faculty of Organizational Sciences, -1, -, 2018, pp. 1097-1103, ISBN 978-86-7680-361-3, - Србија, 7. – 10. јун, 2018 (M33)
3. **Дејан Стојимировић**, Синиша Нешковић, Нина Турајлић, A POSSIBLE APPROACH TO AUTOMATING THE DESIGN OF NoSQL DOCUMENT-ORIENTED DATABASES, Зборник радова са конференције SymOrg2016, University of Belgrade, Faculty of Organizational Sciences, -1, -, 2016, pp. 1280 - 1288, 978-86-7680-326-2, -, Србија, 10. - 13. Jun, 2016 (M33)
4. Kirchner Kathrin, Nešković Siniša, **Stojimirović Dejan**, “From user-understandable to technical process model: a model-driven approach using CUTA4BPM”, 19. 3rd ICIST, Informaciono društvo Srbije, Kopaonik, 2013, str. 161-167 (M33)

Радови објављени у зборницима скупова од националног значаја (M60):

1. **Дејан Стојимировић**, Синиша Нешковић, Слађан Бабарогић, Предлог поступка пројектовања NoSQL база података заснованих на документима, Зборник радова са конференције YUINFO 2015, Информационо друштво Србије, -1, -, 21, pp. 115 - 120, 978-86-85525-15-5, -, Србија, 8. - 11. Mar, 2015 (M63)
2. Мајсторовић Милосав, **Стојимировић Дејан**, Нешковић Синиша, „Преглед модела карактеристика са посебних освртом на примену апстрактних механизма“, 19. YU INFO конференција, Информационо друштво Србије, Копаник, 2013, стр. 93-99 (M63)
3. Бјелобрк Милош, **Стојимировић Дејан**, Нешковић Синиша, „Методологије софтверских производних линија у развоју ПЛИМ система“, 19. YU INFO конференција, Информационо друштво Србије, Копаник, 2013, стр. 319-325 (M63)

Наставни материјал

1. Коаутор у приручнику групе аутора, Информациони системи и технологије: Приручник за припрему пријемног испита – студијски програм Информациони системи и технологије. Уредник др Зоран Марјановић, Факултет организационих наука, Београд. ISBN: 978-86-7680-342-2, 2017

2. Предмет и циљ истраживања

Један од највећих проблема у развоју софтвера представља сложеност развоја. Имајући у виду да су софтверски системи заправо модели реалних система, очигледно је да сложеност развоја потиче из сложености реалних система, тј. из њихових фундаменталних особина. На ове особине пројектанти не могу да утичу, већ морају да их разумеју како би могли да успешно пројектују и развију софтверски систем.

Сложеност се савладава помоћу апстракција. Она је једини методолошки начин како се сложеност може савладати. Апстракција представља свесно занемаривање детаља на неком нивоу апстракције како би се број концепата са којима се суочавамо редуковао на ниво савладиве сложености.

По Тимфу, постоје три фундаментална апстрактна механизма за савладавање сложености: класификација, агрегација и генерализација (Timpf, S, at all, 1999). Поред ова три апстрактна механизма, неопходно је додати и учаурење, као један од основних концепата објектно-оријентисане парадигме.

Сви језици за моделовање (као што су ПМОВ, *UML*, итд.), као и одговарајући имплементациони језици (*Java*, *C++*, *C#*, итд.) подржавају наведене апстрактне механизме. Концепти као што су функције, процедуре, објекти, класе, модули, наслеђивање и сл. су само неки од начина како језици реализују подршку апстракцијама. Ови концепти су, углавном, добро познати и коришћени у пракси.

Један од језичких концепата који се користи за реализацију апстрактних механизма је и концепт простора имена (енгл. *namespace*). У складу са *UML* дефиницијом (OMG Group, 2017), простор имена је именовани концепт који садржи скуп других именованих концепата који се могу идентификовати преко назива простора имена у коме се налазе. Другим речима, простор имена се може третирати и као контејнер имена. Простор имена се у различитим језицима може реализовати на различите начине. Неки језици имају простор имена као концепт, у неким другим језицима је то пакет, у неким модул, док неки језици имају више концепата који представљају простор имена.

Осим проблема у разумевању и коришћењу концепта простора имена од стране пројектаната и програмера, ова неконзистентност има за последицу и проблем у аутоматизацији развоја софтвера заснованог на моделима (енгл. *Model Driven Development – MDD*). У *MDD* приступу, у коме се развој софтвера заснива на трансформацији модела, трансформација концепта простора имена из једног модела у други, исказаних различитим језицима (нпр. *UML* дијаграм пакета и *Javascript* модула), није баш јасна и једнозначна.

Предмет истраживања је концепт простора имена као механизам за подршку апстракцијама, као и аутоматизација трансформације концепта простора имена из једног језика у други.

Циљ рада је идентификовање начина коришћења појединих језичких концепата као механизма за дефинисање простора имена. Поред самих језичких конструкција, ова анализа ће обухватити и поједине апликационе оквире (енгл. *framework*) као примере добре праксе коришћења простора имена. Осим тога, циљ је и дефинисати правила за

трансформацију концепта имена између појединих језичких конструкција у различитим језицима

3. Полазне хипотезе истраживања

Општа хипотеза: Могуће је урадити систематизацију и формализацију карактеристика простора имена у различитим језицима за моделовање и програмским језицима, њихову генерализацију и дефинисање међусобног пресликавања, на основу чега се могу дефинисати правила трансформације између њих.

Помоћна хипотеза 1: Моделом карактеристика се могу формализовати и исказати карактеристике постојећих концепата простора имена у одабраним језицима, односно сличности и разлике између датих језика.

Помоћна хипотеза 2: Могуће је дефинисати општи метамодел за формализацију концепата простора имена који апстрахује конкретне концепте у одабраним језицима.

Помоћна хипотеза 3: Могуће је, на основу општег метамодела, дефинисати пресликавања концепата простора имена у одабраним језицима.

Помоћна хипотеза 4: На основу дефинисаног метамодела и пресликавања, је могуће дефинисати правила трансформације између концепата простора имена у одабраним језицима.

4. Научне методе истраживања

У циљу реализације истраживања, како би се потврдиле постављене хипотезе, коришћене су следеће методе:

- Прикупљање и систематизација података о доступним решењима на основу литературе, разменом информација и непосредним контактима са релевантним субјектима у свету
- Критичка анализа постојећих решења
- Постављање сопствене методологије за остваривање постављених циљева истраживања синтезом познатих и проверених елемената решења, или њиховом модификацијом, са оригиналним елементима решења
- Експериментална провера предложене методологије израдом прототипа
- Практична провера постављених хипотеза тестирањем реализованог прототипа

5. Очекивани научни допринос докторске дисертације

Очекивани резултати истраживачког процеса у предметној области треба да садрже више научних и стручних доприноса, и то:

- Анализа и систематизација карактеристика постојећих концепата простора имена у одабраним језицима
- Формализација идентификованих концепата простора имена у одабраним језицима исказана преко модела карактеристика
- Дефинисан општи метамодел за формализацију концепата простора имена који апстрахује конкретне концепте у одабраним језицима
- Дефинисана пресликавања и правила трансформације између концепата простора имена у одабраним језицима

6. План истраживања и структура рада

6.1. План истраживања

У циљу израде и одбране докторске дисертације, планиране су следеће активности:

1. Прикупљање информација, анализа и систематизација постојећих научних резултата
2. Формулисање проблема неконзистентне употребе концепата простора имена
3. Дефинисање модела карактеристика за систематизацију и формализацију карактеристика постојећих концепата простора имена
4. Дефинисање модела за складиштење конфигурација модела карактеристика
5. Дефинисање модела за „мапирање“ између концепата конкретних конфигурација модела карактеристика и метамодела
6. Дефинисање правила за трансформацију концепата простора имена из једне платформе у другу
7. Развој прототипа приступа
8. Валидација предложеног приступа применом прототипа
9. Разматрање хипотеза и давање закључка
10. Публиковање резултата истраживања

У табели испод је дат план активности по месецима.

	Месеци											
Фаза	1	2	3	4	5	6	7	8	9	10	11	12
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												

6.2. Оквирни садржај докторске дисертације

1. Увод
 - 1.1. Предмет и циљ истраживања
 - 1.2. Полазне хипотезе
 - 1.3. Структура докторске дисертације
2. Апстракција и механизми апстракције
 - 2.1. Фундаменталне апстракције
 - 2.2. Апстракција учурења
 - 2.3. Врсте простора имена
3. Полазне основе за формализацију концепата имена
4. Модели карактеристика као алат за формализацију простора имена
 - 4.1. Приказ модела карактеристика
 - 4.2. Употреба модела карактеристика као алата за формализацију
5. Модел карактеристика за концепте простора имена
6. Општи метамодел
7. Правила за трансформације
8. Студија случаја
9. Закључак

- 9.1. Научни и стручни доприноси
- 9.2. Правци будућег истраживања
- 10. Литература

6.3. Оквирна листа релевантних библиографских извора

1. Noble, J., Biddle, R., Tempero, E., Potanin, A., & Clarke, D. (2003). Towards a model of encapsulation. In *2003. Presented at the ECOOP 2003 IWACO Workshop on Aliasing, Confinement, and.*
2. Néron, P., Tolmach, A., Visser, E., & Wachsmuth, G. (2015, April). A theory of name resolution. In *European Symposium on Programming Languages and Systems* (pp. 205-231). Springer, Berlin, Heidelberg.
3. Achermann, F., & Nierstrasz, O. (2000, September). Explicit namespaces. In *Joint Modular Languages Conference* (pp. 77-89). Springer, Berlin, Heidelberg.
4. Clarke, D. (2001). Object ownership and containment.
5. Clarke, D., Östlund, J., Sergey, I., & Wrigstad, T. (2013). Ownership types: A survey. In *Aliasing in Object-Oriented Programming. Types, Analysis and Verification* (pp. 15-58). Springer, Berlin, Heidelberg.
6. Boyapati, C., Liskov, B., & Shriram, L. (2003). Ownership types for object encapsulation. *ACM SIGPLAN Notices*, 38(1), 213-223.
7. Czarnecki, K., & Helsen, S. (2006). Feature-based survey of model transformation approaches. *IBM systems journal*, 45(3), 621-645.
8. Rose, L. M., Matragkas, N., Kolovos, D. S., & Paige, R. F. (2012, June). A feature model for model-to-text transformation languages. In *2012 4th International Workshop on Modeling in Software Engineering (MISE)* (pp. 57-63). IEEE.
9. Sebesta, R. W. (2012). *Concepts of programming languages*. Boston: Pearson,.
10. Cardelli, L., & Wegner, P. (1985). On understanding types, data abstraction, and polymorphism. *ACM Computing Surveys (CSUR)*, 17(4), 471-523.
11. Michel, R., Classen, A., Hubaux, A., & Boucher, Q. (2011, January). A formal semantics for feature cardinalities in feature diagrams. In *Proceedings of the 5th Workshop on Variability Modeling of Software-Intensive Systems* (pp. 82-89).

12. Apel, S., & Kästner, C. (2009). An overview of feature-oriented software development. *J. Object Technol.*, 8(5), 49-84.
13. Kang, K. C., Kim, S., Lee, J., Kim, K., Shin, E., & Huh, M. (1998). FORM: A feature-oriented reuse method with domain-specific reference architectures. *Annals of Software Engineering*, 5(1), 143.
14. Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., & Peterson, A. S. (1990). *Feature-oriented domain analysis (FODA) feasibility study* (No. CMU/SEI-90-TR-21). Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst.
15. Czarnecki, K., Helsen, S., & Eisenecker, U. (2005). Formalizing cardinality-based feature models and their specialization. *Software process: Improvement and practice*, 10(1), 7-29.
16. Czarnecki, K., Helsen, S., & Eisenecker, U. (2004, August). Staged configuration using feature models. In *International Conference on Software Product Lines* (pp. 266-283). Springer, Berlin, Heidelberg.
17. Apel, S., Batory, D., Kästner, C., & Saake, G. (2013). A development process for feature-oriented product lines. In *Feature-Oriented Software Product Lines* (pp. 17-44). Springer, Berlin, Heidelberg.
18. Riebisch, M., Böllert, K., Streitferdt, D., & Philippow, I. (2002, June). Extending feature diagrams with UML multiplicities. In *6th World Conference on Integrated Design & Process Technology (IDPT2002)* (Vol. 23, pp. 1-7).
19. Czarnecki, K., & Wasowski, A. (2007, September). Feature diagrams and logics: There and back again. In *11th International Software Product Line Conference (SPLC 2007)* (pp. 23-34). IEEE.
20. Дејан Стојимировић, Нина Турајлић, Синиша Нешковић, Ненад Аничић, Слађан Бабарогић, Приступи за моделовање вишеструко повезаних карактеристикама у моделима карактеристика, Зборник радова са конференције Sym-op-is, University of Belgrade, Faculty of Organizational Sciences, 2019, pp 238-243, ISBN: 978-86-7680-363-7, Кладово, Србија, 15. – 18. септембар, 2019
21. Czarnecki, K., Helsen, S., & Eisenecker, U. (2005). Staged configuration through specialization and multilevel configuration of feature models. *Software process: improvement and practice*, 10(2), 143-169.
22. Matić, R. (2016). *Kontekstno orijentisani samoadaptivni sistemi zasnovani na dinamičkim*

- softverskim proizvodnim linijama* (Doctoral dissertation, Univerzitet u Beogradu, Fakultet organizacionih nauka).
23. Acher, M., Collet, P., Lahire, P., & France, R. (2010, June). Comparing approaches to implement feature model composition. In *European Conference on Modelling Foundations and Applications* (pp. 3-19). Springer, Berlin, Heidelberg.
 24. Atkinson, C., Bayer, J., & Muthig, D. (2000). Component-based product line development: the KobrA approach. In *Software Product Lines* (pp. 289-309). Springer, Boston, MA.
 25. Lee, K., Kang, K. C., & Lee, J. (2002, April). Concepts and guidelines of feature modeling for product line software engineering. In *International Conference on Software Reuse* (pp. 62-77). Springer, Berlin, Heidelberg.
 26. Mendonca, M., Wasowski, A., Czarnecki, K., & Cowan, D. (2008, October). Efficient compilation techniques for large scale feature models. In *Proceedings of the 7th international conference on Generative programming and component engineering* (pp. 13-22).
 27. Batory, D. (2005, September). Feature models, grammars, and propositional formulas. In *International Conference on Software Product Lines* (pp. 7-20). Springer, Berlin, Heidelberg.
 28. Czarnecki, K., & Eisenecker, U. W. (2000). Generative programming.
 29. Acher, M. (2011). *Managing, multiple feature models: foundations, languages and applications* (Doctoral dissertation, Nice).
 30. Czarnecki, K., & Antkiewicz, M. (2005, September). Mapping features to models: A template approach based on superimposed variants. In *International conference on generative programming and component engineering* (pp. 422-437). Springer, Berlin, Heidelberg.
 31. Batory, D., Liu, J., & Sarvela, J. N. (2003, September). Refinements and multi-dimensional separation of concerns. In *Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering* (pp. 48-57).
 32. Acher, M., Collet, P., Lahire, P., & France, R. B. (2011, November). Slicing feature models. In *2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)* (pp. 424-427). IEEE.

33. Chen, L., Ali Babar, M., & Ali, N. (2009). Variability management in software product lines: a systematic review.
34. Czarnecki, K., & Pietroszek, K. (2006, October). Verifying feature-based model templates against well-formedness OCL constraints. In *Proceedings of the 5th international conference on Generative programming and component engineering* (pp. 211-220).
35. Jordan, H., Botterweck, G., Noll, J., Butterfield, A., & Collier, R. (2015). A feature model of actor, agent, functional, object, and procedural programming languages. *Science of Computer Programming*, 98, 120-139.
36. Piefel, M. (2006). A common metamodel for code generation.
37. Yang, Z., & Boyland, J. (2008). A Fundamental Permission Interpretation for Ownership Types. In *The 2nd IEEE Theoretical Aspects of Software Engineering Conference, Nanjing, China*.
38. Nierstrasz, O. (1989). A Survey of Object-Oriented Concepts.
39. Damiani, F., Poetzsch-Heffter, A., & Welsch, Y. (2012, March). A type system for checking specialization of packages in object-oriented programming. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 1737-1742).
40. Bugliesi, M., & Jamil, H. M. (1994, September). A logic for encapsulation in object oriented languages. In *International Symposium on Programming Language Implementation and Logic Programming* (pp. 215-229). Springer, Berlin, Heidelberg.
41. Cohen, A. T. (1984). Data abstraction, data encapsulation and object-oriented programming. *ACM SIGPLAN Notices*, 19(1), 31-35.
42. Appelbe, W. F., & Ravn, A. P. (1984). Encapsulation constructs in systems programming languages. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 6(2), 129-158.
43. Clarke, D., & Wrigstad, T. (2003, July). External uniqueness is unique enough. In *European Conference on Object-Oriented Programming* (pp. 176-200). Springer, Berlin, Heidelberg.
44. Potanin, A., Noble, J., Clarke, D., & Biddle, R. (2004). Generic ownership. In *ACM Conference on Object-Oriented Programming Languages, Applications, Languages, and Systems (OOPSLA)*.
45. Kaler, C. G., Brown, A. L., Atkinson, R. G., & Lucco, S. E. (2006). *U.S. Patent No.*

- 6,993,714. Washington, DC: U.S. Patent and Trademark Office.
46. Relyea, R. A., Bogdan, J. L., Ostertag, P. F., & Hillberg, M. J. (2009). *U.S. Patent No. 7,496,828*. Washington, DC: U.S. Patent and Trademark Office.
47. Cameron, N. R., Drossopoulou, S., Noble, J., & Smith, M. J. (2007). Multiple ownership. *ACM SIGPLAN Notices*, 42(10), 441-460.
48. Schärli, N., Black, A. P., & Ducasse, S. (2004, October). Object-oriented encapsulation for dynamically typed languages. In *Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications* (pp. 130-149).
49. Liskov, B. (2015). Perspectives on system languages and abstraction. In *SOSP History Day 2015* (pp. 1-25).
50. Hai, R., Quix, C., & Kensche, D. (2018, October). Nested schema mappings for integrating JSON. In *International Conference on Conceptual Modeling* (pp. 397-405). Springer, Cham.
51. Pingali, V. K., & Touch, J. D. (2008, December). Recursive temporal namespaces. In *Proceedings of the 2008 ACM CoNEXT Conference* (pp. 1-6).
52. Sato, Y., & Chiba, S. (2005, July). Loosely-separated “sister” namespaces in java. In *European Conference on Object-Oriented Programming* (pp. 49-70). Springer, Berlin, Heidelberg.
53. Zaid, L. A., Kleinermann, F., & De Troyer, O. (2010, November). Feature assembly: A new feature modeling technique. In *International Conference on Conceptual Modeling* (pp. 233-246). Springer, Berlin, Heidelberg.
54. Deissenboeck, F., & Pizka, M. (2006). Concise and consistent naming. *Software Quality Journal*, 14(3), 261-282.
55. Ensan, F., & Du, W. (2011). A knowledge encapsulation approach to ontology modularization. *Knowledge and information systems*, 26(2), 249-283.
56. Cheung, W. M., Bramall, D. G., Maropoulos, P. G., Gao, J. X., & Aziz, H. (2006). Organizational knowledge encapsulation and re-use in collaborative product development. *International Journal of Computer Integrated Manufacturing*, 19(7), 736-750.

57. Lu, Y., & Potter, J. (2006). Protecting representation with effect encapsulation. *ACM SIGPLAN Notices*, 41(1), 359-371.
58. De Meuter, W., Tanter, E., Mostinckx, S., Van Cutsem, T., & Dedecker, J. (2005, October). Flexible object encapsulation for ambient-oriented programming. In *Proceedings of the 2005 symposium on Dynamic languages* (pp. 11-21).
59. Schärli, N., Ducasse, S., Nierstrasz, O., & Wuyts, R. (2004, June). Composable encapsulation policies. In *European Conference on Object-Oriented Programming* (pp. 26-50). Springer, Berlin, Heidelberg.
60. Omg.org. 2020. *About The Unified Modeling Language Specification Version 2.5*. [online] Available at: <<https://www.omg.org/spec/UML/2.5/>> [Пристапљено: 11. октобар 2020].
61. Спецификација програмског језика *Java*
62. Спецификација програмског језика *C#*
63. Спецификација програмског језика *PHP*
64. Спецификација програмског језика *Javascript*
65. Спецификација програмског језика *Typescript*
66. Спецификација језика за моделовање пословних процеса *BPMN*
67. Спецификација језика за моделовање *UML*
68. Спецификација језика за означавање *XML*

7. Закључак и предлог

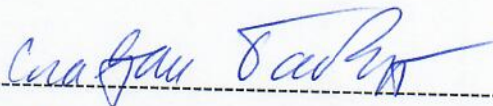
На основу изложеног може се закључити да кандидат Дејан Стојимировић испуњава све услове предвиђене Законом о високом образовању за одобрење израде докторске дисертације под насловом **Концепт простора имена као начин за савладавање сложености у развоју информационих система**.

Тема припада ужој научној области информациони системи. Остварени научни и стручни резултати могу имати утицај на развој и унапређење приступа и алата за аутоматизацију развоја софтвера заснованог на моделима.

На основу свега наведеног, Комисија предлаже Наставно-научном већу да прихвати предложену тему и одобри израду пријављене докторске дисертације. За ментора докторске дисертације се предлаже др Слађан Бабарогић, ванредни професор Факултета организационих наука, Универзитета у Београду.

У Београду, 12.10.2020. године

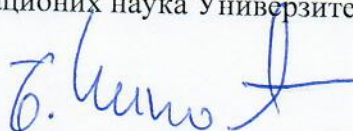
ЧЛАНОВИ КОМИСИЈЕ



др Слађан Бабарогић, ванредни професор Факултета
организационих наука Универзитета у Београду



др Дејан Симић, редовни професор Факултета
организационих наука Универзитета у Београду



др Бошко Николић, редовни професор
Електротехничког факултета Универзитета у Београду

