

29.03.2012. године
Београд,

УНИВЕРЗИТЕТ У БЕОГРАДУ
Веће научних области техничких
наука

Београд
Студентски трг бр 1

У прилогу дописа достављамо потребну документацију за давање сагласности на предлог теме докторске дисертације кандидата мр Илије Антовића.

Захтев се доставља у складу са чланом 54 Закона о Универзитету.

С поштовањем,

СЕКРЕТАР ФАКУЛТЕТА

Гордана Алексић Славински, дипл. Правник

Факултет организационих наука

УНИВЕРЗИТЕТ У БЕОГРАДУ

04-03-11/33
(Број захтева)

Веће научних области техничких наука
(Назив стручног већа коме се захтев упућује
сходночл 6. и члану 7. ст. 1. овог Правилника)

29.03.2012 .
(Датум)

ЗАХТЕВ
за давање сагласности на предлог теме докторске дисертације

Молимо да, сходо члану 57. ст. 3. Закона о универзитету («Службени гласник РС» бр. 21/02), дате сагласност на предлог теме докторске дисертације: (пун назив предложене теме докторске дисертације)
Аутоматско генерисање корисничког интерфејса апликације засновано на случајевима коришћења

НАУЧНА ОБЛАСТ: ОРГАНИЗАЦИОНЕ НАУКЕ

ПОДАЦИ О КАНДИДАТУ:

1. Име, име једног од родитеља и презиме кандидата:
мр Илија (Дарко) Антовић
2. Назив и седиште Факултета на коме је стекао високо образовање: ФОН
3. Година дипломирања: 2004
4. Назив магистарске тезе кандидата: Развој модела и алата за генерисање корисничког интерфејса на основу модела случајева коришћења и модела података
5. Назив факултета на коме је магистарска теза одбрањена: ФОН
6. Година одбране магистарске тезе: 2010. година
7. Назив факултета на коме је кандидат завршио докторске студије: /
Одсек, смер или група: /
Година завршетка докторских студија: -

Обавештавамо вас да је: НАСТАВНО НАУЧНО ВЕЋЕ ФОН-А
(назив надлежног тела факултета)

на седници одржаној: 28.03.2012. године размотрило предложену тему и закључило да је тема подобна за израду докторске дисертације.

ДЕКАН ФАКУЛТЕТА

Проф. др Милан Мартић

Прилог:

1. Предлог теме докторске дисертације са образложењем.
2. Акт надлежног тела факултета о подобности теме за израду докторске дисертације.

ПОДАЦИ О МЕНТОРУ

за кандидата Илију Антовића

Име и презиме ментора: Владан Девеџић

Звање: редован професор

Списак радова који квалификују ментора за вођење докторске дисертације:

1. V. Devedzic, S.R. Milenkovic, "Teaching Agile Software Development: A Case Study", **IEEE Transactions on Education**, Vol.54, No.2, 2011, pp. 273-278. DOI: 10.1109/TE.2010.2052104
ISSN: 0018-9359
IF 2010: 1.313
Листа: Science Citation Index, Science Citation Index Expanded
Област: Engineering, Computing & Technology
2. D. Đurić, V. Devedžić, "Incorporating the Ontology Paradigm Into Mainstream Programming Environment", **Informatica**, 2012. (forthcoming)
ISSN: 0868-4952
IF 2010: 1.186
Листа: Science Citation Index Expanded
Област: Engineering, Computing & Technology
3. D. Đurić, V. Devedžić, "Incorporating the Ontology Paradigm Into Software Engineering: Enhancing Domain-Driven Programming in Clojure/Java", **IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews**, Vol. 41, Issue 6, pp 1-13, 2011. DOI: 10.1109/TSMCC.2011.2140316
ISSN: 1094-6977
IF 2011: 2.132
Листа: Science Citation Index, Science Citation Index Expanded
Област: Engineering, Computing & Technology
4. D. Djuric, V. Devedzic, "Magic Potion: Incorporating New Development Paradigms through Metaprogramming", **IEEE Software**, Vol.27, No.5, 2010, pp. 38-44, DOI: 10.1109/MS.2010.90
ISSN: 0740-7459
IF 2010: 1.905
Листа: Science Citation Index, Science Citation Index Expanded
Област: Engineering, Computing & Technology
5. M. Devedzic, V. Devedzic, "Imagine: Using New Web Technologies in Demography", **Social Science Computer Review**, Vol.28, No.2, 2010, pp. 206-231, DOI: 10.1177/0894439309332661
ISSN: 0894-4393
IF 2010: 1.202
Листа: Science Citation Index, Science Citation Index Expanded
Област: Social & Behavioral Sciences

Заокружити одговарајућу опцију(А,Б,В или Г):

А) У случају менторства дисертације на докторским студијама у групацији техничко-технолошких, природно-математичких и медицинских наука ментор треба да има најмање три рада са SCI, SSCI, AHCI или SCIE листе.

Б) У случају менторства дисертације на докторским студијама у групацији друштвено-хуманистичких наука ментор треба да има најмање три рада са релевантне листе научних часописа (Релевантна листа научних часописа обухвата SCI, SSCI, AHCI и SCIE листе, листу часописа које је Министарство за науку класификовало као M24 и додатну листу часописа коју ће, на предлог универзитета, донети Национални савет за високо образовање. Посебно се вреднују и монографије које Министарство науке класификује као M11, M12, M13, M14, M41 и M51.)

В) У случају израде докторске дисертације према ранијим прописима за кандидате који су стекли академски назив магистра наука ментор треба да има пет радова (рефернци) које га, по оцени Већа научних области, квалификују за ментора односне дисертације.

Г) У случају да у ужој научној области нема квалификованих наставника, приложити одлуку Већа докторских студија о именовању редовног професора за ментора.

ДЕКАН ФАКУЛТЕТА

Датум 28.03.2012.

М.П.

Поводом предлога Већа студијског програма докторских академских студија, а на основу члана 56. Статута Факултета, на седници Наставно-научног већа одржаној 28.3.2012. године, једногласно је донета следећа

ОДЛУКА

Усваја се Извештај Комисије за оцену научне заснованости пријављене докторске дисертације кандидата мр Илије Антовића под насловом »АУТОМАТСКО ГЕНЕРИСАЊЕ КОРИСНИЧКОГ ИНТЕРФЕЈСА АПЛИКАЦИЈЕ ЗАСНОВАНО НА СЛУЧАЈЕВИМА КОРИШЋЕЊА«.

За ментора се предлаже др Владан Девеџић, ред. проф. ФОН-а. Извештај се доставља Већу научних области техничких наука Универзитета у Београду.

Председник Наставно научног већа

Проф. др Милан Мартић

Доставити:

- Служби за ПДС
- Секретару органа управљања и стручних органа
- Архиви

Наставно-научном већу ФОН-а

Одлуком Научно- наставног већа ФОН-а од 22.02.2012. године именовани смо у Комисију за оцену научне заснованости пријављене докторске дисертације кандидата мр Илије Антовића насловом: “Аутоматско генерисање корисничког интерфејса апликације засновано на случајевима коришћења” и на основу тога подносимо следећи

ИЗВЕШТАЈ

1. ПОДАЦИ О КАНДИДАТУ

Општи биографски подаци

Илија Антовић је рођен 8. 7. 1980. године у Котору, република Црна Гора.

Основну школу завршио је у Котору као и нижу музичку школу – инструмент клавир и гимназију – општи смер.

Године 1999. уписује Факултет организационих наука Универзитета у Београду на коме је дипломирао 2004. године. Након тога уписује постдипломске студије. Магистрирао је у области софтверског инжењерства 2010. године на Факултету организационих наука. Од 2000. године ангажован је на извођењу лабораторијских вежби на Факултету организационих наука, а од 2006. године запослен је у Лабораторији за софтверско инжењерство на ФОН-у. Ожењен је супругом Ивом са којом има сина Ива.

Подаци о магистратури

На Факултету организационих наука Илија Антовић је завршио магистарске студије са просечном оценом 10, и одбранио магистарски рад под називом: “Развој модела и алата за генерисање корисничког интерфејса на основу модела случајева коришћења и модела података”, 23. јуна 2010. године.

Списак објављених радова или других релевантних резултата и активности које
опредељују афинитет кандидата за рад на приложеној теми

Као сарадник Лабораторије за софтверско инжењерство учествовао је у извођењу значајних пројеката међу којима су:

- Идејни пројекат информационог система е-аукцијске јавне набавке (за потребе Министарства за телекомуникације и информатичко друштво Републике Србије),
- Пројекат KOSTMOD (Forsvarets forskningsinstitutт Министарства одбране Краљевине Норвешке и Министарства одбране Републике Србије).
- Модернизација информационог система за здравствене установе, IQ-net, Београд 2006.

Поред тога учествовао је у извођењу пројеката:

- Sodiumlight, Streetlight Management System (SMCS), a pilot project for the Qatar Public Works Authorities in Doha, 2010.
- IT strategy support to the modernization of the Ministry of finance and economy of Serbia, Европска Агенција за Реконструкцију, Београд 2006.
- Development of health information system for basic health and pharmaceutical services, republic of Serbia, Euro Health Group, Denmark, 2007.
- Пројектовање информационог система за ПР агенцију, PRA, Београд 2004.

Истакао се и у научно-истраживачком раду. Коаутор је књиге Пројектовање софтвера: Напредне Јава технологије, а аутор је више научних радова објављених на симпозијумима и у часописима. Октобра 2009. године одржао је гостујуће предавање на Friedrich-Schiller Univerzitetу у Јени – Немачка.

Списак објављених радова

Магистарски рад

- И. Антовић: “Развој модела и алата за генерисање корисничког интерфејса на основу модела случајева коришћења и модела података”, Област: Софтверско инжењерство, Ментор: др Синиша Влајић, ФОН, Београд, јун 2010.

Радови објављени у часописима

- Ј. Стојановић, Д. П. Милошевић, И. Антовић, Г. Секулић, Т. Бељић-Живковић, Утицај различитих режима инсулинске терапије на квалитет живота особа оболелих од DIABETES MELLITUS-а тип 1, Војносанитетски преглед, Београд, 2011. (потврда о прихватању за штампу)
- И. Антовић, С. Влајић, Примјена софтверских патерна у рјешавању проблема савременог менаџмента – пројектовање виртуалних организација –, часопис ИнфоМ, Волумен 18/2006, Београд
- Н. Ристин, С. Влајић, И. Антовић, М. Милић, В. Станојевић, Компаративна анализа Java и .Net WEB сервиса, часопис ИнфоМ, Волумен 40/2011, Београд

Радови објављени на конференцијама

- Д. Савић, С. Влајић, И. Антовић, В. Станојевић, М. Милић, Language for Use Case Specification, In: Proceedings of the 34th annual IEEE Software Engineering Workshop (SEW34). IEEE Computer Society, SEW-34 34th Annual IEEE Software Engineering Workshop Limerick, Ireland, 2011.
- С. Влајић, Д. Савић, И. Антовић, The Explanation of the Design Patterns by the Symmetry Concepts , The 14th IASTED International Conference on Artificial Intelligence and Soft Computing, (ASC 2011), Crete, Greece, 2011.
- И. Антовић, В. Станојевић, С. Влајић, Пројектовање виртуалних организација коришћењем софтверских патерна, XII конференција YU INFO, Копаоник, 2006.
- И. Антовић, С. Влајић, Софтверски патерни у функцији пројектовања виртуалних организација, SymOrg, X Међународни симпозијум, Златибор 2006.
- Г. М. Секулић, С. Влајић, И. Антовић, Метамодел валидационог механизма код STRUTS, SPRING MVC и JSF JAVA WEB оквира, XVII конференција YU INFO & ICIST, Копаоник, 2011.
- Г. М. Секулић, С. Влајић, И. Антовић, Компаративна анализа савремених Java web апликационих оквира, XVI конференција YU INFO & ICIST, Копаоник, 2010.
- А. Николић, И. Антовић, С. Влајић, М. Милић, Д. Савић, С. Д. Лазаревић, Компаративна анализа Hibernate и EJB технологије, Фестивал информатичких достигнућа – INFOFEST, Будва, 2011.
- И. Антовић, Д. Савић, В. Станојевић, М. Милић, С. Влајић, Алати и методе софтверског инжењерства по SWEBOOK пројекту, XIV конференција YU INFO, Копаоник, 2008.
- М. Милић, С. Влајић, И. Антовић, В. Станојевић, Д. Савић: Софтверске метрике као техника за евалуацију и побољшање квалитета софтвера, SymOrg, XI Међународни симпозијум, Београд 2008.

- В. Станојевић, И. Антоновић, С. Вајић, Квалитет софтвера по SWEBOOK пројекту, XII конференција YU INFO, Копаоник, 2006.
- Д. Савић, В. Станојевић, И. Антоновић, Развој информационог подсистема рендген службе у јава окружењу, SymOrg, X Међународни симпозијум, Златибор 2006.
- Б. Јовановић, Д. Савић, С. Влајић, И. Антоновић, Web портали за управљање одржавањем, Фестивал информатијских достигнућа – ИНФОФЕСТ, Будва, 2006.

Објављене књиге

- С. Влајић, Д. Савић, В. Станојевић, И. Антоновић, М. Милић, Пројектовање Софтвера – Напредне Јава технологије, Златни Пресек, Београд, 2008.

Оцена подобности кандидата за рад на предложеној теми

Кандидат Илија Антоновић већ неколико година ради на Факултету организационих наука при Лабораторији за софтверско инжењерство. Области интересовања кандидата су аутоматизација процеса развоја корисничког интерфејса, софтверски захтеви, софтверски патерни, савремене софтверске архитектуре. Као што се у приложеној биографији може видети био је ангажован на неколико значајних пројеката, публиковао је више научних радова у научним часописима и на конференцијама. Кандидату је прихваћен рад за штампу у часопису који је у 2010 години имао *Impact Factor* 0.199.

На основу свега наведеног, сматрамо да кандидат Илија Антоновић поседује потребно знање и искуство за израду докторске дисертације на тему **„Аутоматско генерисање корисничког интерфејса апликације засновано на случајевима коришћења“**.

2. ПРЕДМЕТ И ЦИЉ ИСТРАЖИВАЊА

Приказ проблема и предмета истраживања

Развој пословних апликација данас подразумева коришћење тронивојске архитектуре. Овај концепт предвиђа постојање три логичке целине – нивоа: ниво корисничког интерфејса, апликационе логике и складишта података. [Larman98]¹

1. Кориснички интерфејс представља улазно – излазну репрезентацију софтверског система.
2. Апликациона логика представља структуру и понашање софтверског система.
3. Складиште података одговорно је за чување стања атрибута софтверског система.

¹ [Larman98]C. Larman, *Applying UML and Patterns – An Introduction to Object-Oriented Analysis and Design and the Unified Process*, Second edition, Prentice Hall

Концепт корисничког интерфејса има више значења. Може да се односи на изглед екрана, распоред графичких компоненти, а може да буде посматран као један подсистем или ниво у архитектури софтверског система.

Са становишта корисника, разлика између нивоа корисничког интерфејса и других нивоа софтверског система није јасно уочљива, а најчешће је потпуно невидљива. За већину корисника софтверски систем представља средство помоћу којег се извршавају одређени задаци. За њих, кориснички интерфејс није само један део софтверског система, већ читав систем. [Vliet08]²

Са друге стране, са становишта пројектанта система, кориснички интерфејс се посматра као реализација улаза и/или излаза софтверског система. [Vlajić03]³ Постоји више различитих метода у развоју софтверских система, [Vliet08] али карактеристика која их повезује јесте раздвајање пројектовања функционалности система од пројектовања начина интеракције корисника са системом. Како животни циклус софтвера подразумева фазе прикупљања корисничких захтева, анализу, пројектовање, имплементацију и тестирање, пројектовање корисничког интерфејса започиње тек након завршетка фазе прикупљања корисничких захтева, или чак након пројектовања апликационе логике. Пројектовање корисничког интерфејса тада се своди на постављање графичких компоненти на екранске форме, прављење менија, величине компоненти, формата дијалога са порукама итд.

Насупрот томе, постоји и другачији приступ који предлаже пројектовање корисничког интерфејса паралелно са спецификацијом корисничких захтева. Овај приступ заснован је на корисничкој перспективи схватања софтверског система, која се може сумирати тезом да *кориснички интерфејс за корисника не представља само један део софтверског система већ је за њега то систем у целини*. [Vliet08] Разлог томе јесте што корисник не познаје детаље структуре и начина реализације система. Специфицирањем корисничких захтева даје се приказ начина на који корисник жели да изврши одређени задатак помоћу софтверског система. Извршење сваког од ових задатака подразумева интеракцију корисника и система која има одређену структуру. Кориснички интерфејс, по овом приступу, треба да прати ову структуру како би се достигао што виши ниво корисности (usability) софтверског система. Креирање корисничког интерфејса у раним фазама не подразумева потпуно функционалан интерфејс, већ прототип који првенствено има за циљ да на површину избаци питања и претпоставке које можда нису биле јасне применом других приступа у прикупљању и анализи захтева. [Pfleege06]⁴

Веома битан аспект код развоја софтверских система јесте цена извођења софтверског пројекта. Када се узме у обзир чињеница да израду софтверског производа чини низ интелектуалних и техничких активности у којима учествују високо школовани инжењери, јасно је да од времена и напора које је потребно уложити у производњу система у великој мјери зависи његова цена. Израда корисничког интерфејса софтверског система има значајан удео у читавом процесу. Када се погледају резултати

² [Vliet08] H. V. Vliet, Software engineering: principles and practice, John Wiley & Sons Ltd, 3rd edition, Chichester, West Sussex, England, 2008.

³ [Vlajić03] S. Vlajić, *Projektovanje programa (skripta)*, Dr Siniša Vlajić, Beograd, 2003.

⁴ [Pfleege06] S. L. Pfleege, J. M. Atlee, *Software Engineering Theory and Practice*, Third edition, Prentice Hall, 2006.

истраживања ове проблематике, интересантан је податак да 48% програмског кода софтверског система чини део везан за кориснички интерфејс, а чак 50% времена од укупног времена развоја одлази на његову имплементацију. [Myers]⁵ Слични резултати добијени су и у другим истраживањима у овој области [Kivisto00]⁶, без обзира на то што се у развоју користе алати који убрзавају и олакшавају овај процес.

Због чињенице да кориснички интерфејс представља директну везу корисника са системом, често се у току израде софтверског система дешава да корисник мења захтеве везане за његов изглед и структуру. Ове измене понекад изискују промене над имплементираним системом које неретко резултују одбацивањем до тада развијаног интерфејса и понављањем целокупног процеса. Чак и након тога, дешава се да корисник открије да ни ново решење не одговара његовим потребама, јер корисник често није у могућности да своје потребе довољно добро искаже кроз спецификацију захтева.

Све ове чињенице намећу потребу за аутоматизацијом овог процеса која треба да доведе до решења које ће убрзати процес креирања корисничког интерфејса и смањити његове трошкове. При томе, решење мора да у што већој мери максимизира атрибуте квалитета корисничког интерфејса. Добијени кориснички интерфејс мора бити заснован на корисничким захтевима. На тај начин, убрзао би се сам развој корисничког интерфејса, али и добио алат којим би се ефикасније извршио процес спецификације корисничких захтева. У том смислу корисник би брзо могао да уочи недостатке у спецификацији захтева и отклони их, како би добио систем који што ближе задовољава његове потребе. Аутоматизација овог процеса имала би утицај на све фазе животног циклуса софтвера: од прикупљања захтева до његове имплементације и тестирања.

Како је на самом почетку наведено, полазна тачка у креирању софтверског производа јесте фаза прикупљања корисничких захтева и резултати ове фазе представљају основу за све остале фазе у животној циклусу софтвера. Из тог разлога спецификација корисничких захтева треба да буде довољно детаљна да пружи потребне информације за пројектовање и имплементацију свих делова софтвера – од корисничког интерфејса, преко апликационе логике која учаурује доменску структуру и понашање система, до складишта података. Међутим, често спецификација корисничког захтева не даје довољно информација за пројектовање појединих делова система, па се препушта програмеру да на свој начин интерпретира захтеве и налази решења, а често добијено решење не задовољава потребе корисника. [Frost07]⁷ У том смислу, уколико се разматра развој корисничког интерфејса, из корисничког захтева ретко се могу добити информације о изгледу корисничког интерфејса, његовом садржају, врстама компоненти које треба да се користе и сл. Такође, многи модели, развијени за подршку пројектовању софтверских компоненти и њихових међусобних интеракција, не обрађују довољно организовање компоненти корисничког интерфејса као ни

⁵ [Myers] B. Myers, M. Rosson, *Survey on user interface programming*, ACM: Human Factors in Computing Systems, Proceedings SIGHI, 1992

⁶ [Kivisto00] K. Kivistö, *A third generation object-oriented process model, Roles and architectures in focus*, Academic Dissertation, Faculty of Science, University of Oulu, Oulu, 2000.

⁷ [Frost07] A. Frost, M. Campo, *Advancing Defect Containment to Quantitative Defect Management*, The Journal of Defense Software Engineering, Dec. 2007. Issue, 2007.

интеракцију корисника са овим компонентама. [Harmelen01]⁸ [DaSilva00]⁹ Зато је потребно одабрати технику којом ће се што целовитије специфицирати кориснички захтев, како би било могуће пројектовање различитих делова софтверског система. Данас, најраспрострањеније технике за прикупљање корисничких захтева јесу спецификација захтева помоћу случајева коришћења – *use case specification*, [Cockburn00]¹⁰ [Jacobson93]¹¹ и преко корисничких прича – *user stories*. [Cohn04]¹² Случајеви коришћења и корисничке приче имају доста додирних тачака (засновани су на сценаријима и усмерени ка циљевима корисника), али и битних разлика (у структурираности, опсегу који покривају, целовитости, трајности и детаљности). [Stellman09]¹³ И једна и друга техника имају за циљ да специфицирају намеру корисника тј. циљ који корисник жели да оствари кроз жељену функционалност.

Разматрајући наведене технике за прикупљање и спецификацију корисничких захтева може се уочити да се случајеви коришћења намећу као прихватљивије решење у односу на корисничке приче, нарочито ако се жели постићи аутоматизација креирања одређених делова софтвера. Разлог за то је њихова структурираност, целовитост као и могућност уључивања додатних детаља значајних за различите аспекте система. Ипак, укључивање додатних детаља у случајеве коришћења не препоручује се од стране неких аутора, или бар не у почетним фазама спецификације захтева. Међутим, многи аутори се слажу у оцјени да је, узимајући у обзир структуру спецификације корисничких захтева (SRS) као документа, најзгодније мјесто за приказивање захтева везаних за кориснички интерфејс, управо уз спецификацију случајева коришћења.

Најчешћи узрок неуспеха софтверских пројеката јесу недостаци и грешке у корисничким захтевима. [Frost07] Један од начина за њихово отклањање у раним фазама јесте прављење прототипова корисничког интерфејса. Корисник преко корисничког интерфејса ступа у интеракцију са системом и управља његовим стањима. У објектно оријентисаним апликацијама, стања система учаурују се у доменске објекте система који, заједно са везама и ограничењима, чине доменски модел софтверског система. Доменски модел описује статичку структуру софтверског система, тј. основне концепте из домена пословног проблема који се пројектује. Овај модел се још назива и концептуални модел, и прави се у раној фази (фаза анализе) животног циклуса софтвера. Доменски модел представља полазну тачку за даље пројектовање модела података који ће бити основа за пројектовање складишта података софтверског система. Како случајеви коришћења описују интеракцију између корисника и система, која се реализује преко корисничког интерфејса и има за циљ промену стања система,

⁸ [Harmelen01] V.M. Harmelen, *Object modeling and user interface design: designing interactive systems*. Addison Wesley Longman Publishing Co. Inc, Boston, 2001.

⁹ [DaSilva00] P. P. Da Silva, N. W. Paton, *User Interface Modeling With UML*, Proceedings of the 10th European-Japanese Conference on Information Modeling and Knowledge Representation, 2000.

¹⁰ [Cockburn00] A. Cockburn, *Writing Effective Use Cases*, Addison Wesley Longman Publishing Co. Inc, Boston, 2000.

¹¹ [Jacobson93] I. Jacobson, M. Christerson, *Object-Oriented Software Engineering*, Addison Wesley Longman Publishing Co. Inc, Boston, 1993.

¹² [Cohn04] M. Cohn, *User Stories Applied: For Agile Software Development*, Addison Wesley Longman Publishing Co. Inc, Boston, 2004.

¹³ [Stellman09] A. Stellman, *Requirements 101: User Stories vs. Use Cases*, <http://www.stellman-greene.com/2009/05/03/requirements-101-user-stories-vs-use-cases/>, Building Better Software: Jennifer Greene and Andrew Stellman weblog, postavljeno: 3. 05. 2009, preuzeto: 1. 07. 2009.

веза између ова три дела система морала би бити узета у обзир већ у фази прикупљања корисничких захтева. Када би се узела у обзир ова чињеница приликом креирања прототипа корисничког интерфејса, такав интерфејс би се могао користити за валидацију како случајева коришћења, тако и модела података. [Fitzgerald05¹⁴] Под валидацијом се подразумева провера усаглашености корисничких захтева са пројектованим доменским моделом.

Коришћење алата за генерисање корисничког интерфејса у великом делу би аутоматизовало процес пројектовања, имплементације и тестирања корисничког интерфејса што би као последицу довело до промена у спровођењу методе развоја софтвера. Поставља се питање да ли и даље треба користити постојећу методу развоја, или предложити измене у методи како би се време уштеђено аутоматизацијом искористило за пребацивање фокуса инжењера на остале аспекте система и њихово унапређивање.

У ширем смислу предмет овог истраживања јесте аутоматизација процеса имплементације корисничког интерфејса, као и веза између софтверских захтева и будућег корисничког интерфејса апликације узимајући у обзир карактеристике циљаних технологија и платформи за извршење. Разматрање овог предмета има за циљ утврђивање принципа и карактеристика које треба да буду саставни део будућег алата за аутоматизацију процеса имплементације корисничког интерфејса – генератора корисничког интерфејса, који ће имплементирати интерфејс у складу са корисничким захтевима, а у исто време у што већој мери убрзати процес развоја софтвера, смањити његове трошкове, и омогућити његово даље одржавање, а да при томе корисници алата имају слободу избора између различитих циљаних технологија и архитектура.

Како би утврдили који су то принципи и карактеристике које би омогућиле постизање ових задатака, потребно је извршити анализу постојећих алата и парадигми које су до сада коришћене, и утврдити предности и мане са циљем да се избегне понављање грешака које су за резултат имале чињеницу да ни један од постојећих алата није стекао значајнију популарност у софтверској индустрији. Зато ће се у тези дефинисати скуп критеријума који ће служити као средство за поређење постојећих алата, а касније и за упоређивање алата који ће бити резултат овог истраживања са постојећом праксом. У раду ће детаљно бити образложен разлог увођења сваког од критеријума, као и предности које се очекују испуњењем сваког од њих. Посебна пажња биће посвећена утврђивању веза између корисничких захтева специфицираних случајевима коришћења са будућим корисничким интерфејсом, и то анализом веза између структуре (доменски модел) и понашања (системске операције) система који су изведени из спецификације случајева коришћења са жељеним корисничким интерфејсом.

Због чињенице да се кориснички интерфејс за један исти кориснички захтев може реализовати на више различитих начина, биће разматрани кориснички интерфејси неколико различитих софтверских система како би се утврдиле заједничке карактеристике интерфејса, и идентификовали најчешће коришћени начини приказа корисничког интерфејса за исте или сличне корисничке захтеве. На овај начин, доћи ће се до скупа шаблона – темплејта корисничких интерфејса, па корисник алата неће бити

¹⁴ [Fitzgerald05] J. Fitzgerald, P. Larsen, P. Mukherjee, N. Plat, M. Verhoef, *Validated Designs for Object-oriented Systems*, Springer-Verlag, London, 2005.

ограничен на један приказ већ ће моћи да, на брз и једноставан начин, мења приказ и одабере онај који крајњем кориснику највише одговара. На овај начин постигла би се флексибилност у креирању корисничког интерфејса за исти кориснички захтев.

Како је раније утврђено, кориснички интерфејс представља реализацију улаза односно излаза софтверског система. Посматра се као засебан слој у софтверском систему, али чини његов саставни део. Данас постоји много различитих типова софтверских система - апликација, и најгрубље их можемо подијелити на web, desktop и мобилне апликације. За прављење сваке од ових апликација развијено је мноштво технологија и платформи за извршење. Свака од ових технологија пружа могућност прављења корисничког интерфејса. Алат за аутоматизацију процеса имплементације корисничког интерфејса треба да пружи могућност одабира између различитих циљних технологија. Како технологија има много, не треба имати амбицију прављења алата који би реализовао генерисање интерфејса за све технологије, али би архитектуру генератора требало пројектовати тако да буде проширива коришћењем софтверских патерна [GOF]¹⁵, како би се за сваку нову технологију могла независно имплементирати својеврсна „pluggable“ компонента, која се једноставно може интегрисати у алат. На тај начин, пружила би се флексибилност корисницима алата у избору циљане технологије за имплементацију корисничког интерфејса. Да би се ово остварило, претходно треба утврдити „компатибилност“ графичких компоненти и идентификованих шаблона корисничког интерфејса између различитих типова апликација. На пр. ако је један шаблон корисничког интерфејса одговарајући за десктоп апликацију, не мора да значи да ће да буде адекватан и за web или мобилну апликацију, најчешће због технолошких ограничења, па се мора извршити анализа карактеристика сваке врсте апликација и дефинисати јасна трансформација између основног скупа компоненти и шаблона и изведених шаблона који више одговарају одређеној врсти апликације.

Посебна пажња у истраживању биће посвећена структури генерисаног програмског кода. Биће разматрани различити софтверски патерни пројектовања како би се идентификовало који су патерни најпогоднији за програмски код генерисаног корисничког интерфејса како би се обезбедило његово лакше одржавање.

Поред свега наведеног, теза ће разматрати и карактеристике које нису у директној вези са корисничким интерфејсом, а које се најчешће имплементирају заједно са корисничким интерфејсом. Ове карактеристике се пре свега односе на валидацију улазних података и дефинисање корисничких улога и права приступа одређеним деловима софтверског система.

Како би оваква аутоматизација била могућа, потребно је дефинисати семантички богат модел корисничких захтева који ће бити обогаћен информацијама везаним за кориснички интерфејс. На тај начин омогућиће се генерисање извршивог кода корисничког интерфејса независно од циљане технологије или типа апликације (десктоп апликације у различитим технологијама, web апликације у различитим технологијама итд.). Идеја потиче од чињенице да кориснички захтев садржи информацију о корисниковим намјерама, функционалностима и циљевима које жели да постигне коришћењем система, без обзира на начин имплементације тог система. Алат

¹⁵ [GOF] E. Gamma, R. Helm, R. Jonson, J. Vilissides, *Design patterns*, Addison Wesley, 18th Printing, 1999.

који ће вршити аутоматизацију биће пројектован тако да омогући трансформацију модела у извршиви програмски код корисничког интерфејса за различите имплементационе технологије. Биће извршена анализа утицаја коришћења алата на методу развоја софтвера и биће предложене измене како би се на што бољи начин задовољили захтеви крајњих корисника.

Циљ истраживања

Циљ овог рада је да се развије модел за спецификацију корисничког захтева који ће бити довољно семантички богат како би на основу њега било могуће не само пројектовање и имплементација корисничког интерфејса, већ и аутоматизација овог процеса. На основу извршене анализе постојећих алата и парадигми које су до сада коришћене и идентификованих предности и мана сваког од њих у тези ће се дефинисати скуп критеријума који ће служити као средство за поређење постојећих алата, а касније и за упоређивање алата који ће бити резултат овог истраживања са постојећом праксом. У раду ће детаљно бити образложен разлог увођења сваког од критеријума, као и предности које се очекују испуњењем сваког од њих. Посебна пажња биће посвећена утврђивању веза између корисничких захтева специфицираних случајевима коришћења са будућим корисничким интерфејсом, и то анализом веза између структуре (доменски модел) и понашања (системске операције) система који су изведени из спецификације случајева коришћења са жељеним корисничким интерфејсом. Циљ је да модел на основу којег ће се вршити генерисање програмског кода корисничког интерфејса садржи у себи све потребне информације како би било могуће задовољење свих постављених критеријума. Како би се олакшало креирање овог модела биће развијен софтверски алат који има улогу да води корисника кроз процес спецификације корисничких захтева (у облику “wizard”-а) и да конструише семантички богат модел који је потом спреман за трансформацију у извршиви код.

Апликација развијена током овог истраживања - генератор - треба да покаже да је могућа аутоматизација процеса креирања корисничког интерфејса на основу предложеног модела у складу са дефинисаним критеријумима, како би се добио извршиви програмски код у различитим имплементационим технологијама.

Поред тога, на основу анализе начина интеракције корисника са системом, циљ је да се дође до скупа могућих шаблона корисничког интерфејса којим се различити кориснички захтеви могу приказати.

Крајњи циљ овог рада је да се у што већој мери смањи вријеме и напор, а тиме и трошкови процеса имплементације корисничког интерфејса апликације као најзахтевнијег дела процеса имплементације читавог софтверског система. Као доказ успешности овог истраживања биће спроведен експеримент који ће подразумевати учешће искусних програмера и мерење времена развоја и измене софтверског система са и без коришћења генератора, као и анкета у којој ће програмери моћи да дају своју оцену о предложеном приступу и вредности оваквог алата у свакодневном коришћењу за потребе привреде, као и евентуалне предлоге за његово побољшање.

Студијом случаја, биће презентован целокупан процес развоја корисничког интерфејса предложеним приступом.

Осврт на релевантне библиографске изворе и остварене резултате (цитирати) који су у вези са предметом истраживања

У претходним поглављима изложене су библиографске јединице везане за појединачне сегменте овог истраживања, а у наставку ће бити приказани извори и тренутно актуелни пројекти и приступи који се односе на предмет истраживања.

До данас је било више покушаја да се направи алат који би аутоматизовао процес имплементације дела или читавог корисничког интерфејса апликације, али нити један од ових алата није успео да се значајније наметне на тржишту и око себе окупи шири круг корисника. [Kennard10]¹⁶ Постоје бројни алати који аутоматизују процес имплементације корисничког интерфејса, али само је мали број њих успео да привуче интересовање академске и индустријске јавности. Међу њима најзначајнији су: Metawidget, Web Ratio, Apache Isis (Naked Objects), AlphaSimple, и они ће бити детаљно анализирани у раду. Поред набројаних постоји и већи број алата и приступа, који су још увек у фази истраживања, али који су веома значајни за предмет овог истраживања. То су пре свега приступи [Elkoutbi00]¹⁷, [DaCruz10]¹⁸, [Kennard10]. Анализом ових алата и приступа може се утврдити да међу њима постоје значајне разлике. Ове разлике су присутне како у самим полазним основама па све до реализације генерисања корисничког интерфејса. Као полазне основе у постојећим алатима могу се наћи различити модели међу којима су најзаступљенији: модел задатка (Task model), доменски модел (Domain model), кориснички модел (User model), модел дијалога (Dialogue model) и презентациони модел (Presentation model). [Schlungbaum96]¹⁹ [Constantine03]²⁰ Што се тиче начина реализације аутоматског генерисања корисничког интерфејса, поједини алати обезбеђују динамичко генерисање корисничког интерфејса током извршења апликације, док други предвиђају генерисање програмског кода корисничког интерфејса. Такође, поједини алати омогућавају измене над генерисаним корисничким интерфејсом, као и неке додатне функционалности као што су флексибилност у начину приказа компоненти, избору компоненти које ће се наћи на корисничком интерфејсу, избору између корисничког интерфејса за десктоп и web апликације, избор између различитих циљних архитектура, док је код других читав интерфејс једнообразан и потпуно дефинисан самим алатом.

Све ове разлике говоре о томе да још увијек нису успостављени стандарди нити јасна правила која би требала да важе приликом генерисања корисничког интерфејса. Један од важних задатака овог истраживања јесте да се испитају предности и недостаци

¹⁶ [Kennard10] R. Kennard, J. Leaney, *Towards a general purpose architecture for UI generation*, Journal of Systems and Software, Volume 83, Issue 10, 2010.

¹⁷ [Elkoutbi00] M. Elkoutbi, R. K. Keller, *User interface prototyping based on UML scenarios and high-level Petri nets*, ICATPN'00 Proceedings of the 21st international conference on Application and theory of petri nets Springer-Verlag Berlin, Heidelberg, 2000.

¹⁸ [DaCruz10] A. M. R. da Cruz, J. P. Faria, *Automatic Generation of User Interface Models and Prototypes from Domain and Use Case Models*, User Interfaces, InTech, Croatia, May, 2010.

¹⁹ [Schlungbaum96] E. Schlungbaum, *Model-based User Interface Software tools – Current state of declarative models*, Graphics, Visualization and Usability Center, Georgia Institute of Technology, Gvu Tech Report, Atlanta, 1996.

²⁰ [Constantine03] L. Constantine, L. Lockwood, *Structure and Style in Use Cases for User Interface Design*, 2nd International Conference on Usage-Centered Design, New Hampshire, 2003.

сваког од ових приступа, и да се на основу тога успостави скуп критеријума који треба да буду задовољени како би се почело са реализацијом генератора корисничког интерфејса који ће узети у обзир уочену постојећу добру праксу у овој области и објединити је. Отклањањем недостатака и усвајањем добрих карактеристика из постојећих приступа стварају се предуслови да нови алат на бољи начин реализује аутоматизацију процеса имплементације корисничког интерфејса, и тиме се наметне ширем кругу корисника.

3. ПОЛАЗНЕ ХИПОТЕЗЕ

На основу дефинисаних циљева истраживања, дефинисане су следеће хипотезе истраживања:

- Могуће је успоставити везу између модела случајева коришћења и жељеног корисничког интерфејса.
- На основу семантички богатог модела заснованог на моделу случајева коришћења и информацијама везаним за кориснички интерфејс могућа је аутоматизација процеса генерисања извршивог програмског кода корисничког интерфејса.
- Модел једног корисничког захтева може се аутоматски трансформисати у више шаблона корисничког интерфејса, задржавајући жељену функционалност.
- Модел једног корисничког захтева може се аутоматски трансформисати у програмски код корисничког интерфејса на различитим имплементационим технологијама за различите типове апликација

4. НАУЧНЕ МЕТОДЕ ИСТРАЖИВАЊА

Истраживање ће бити реализовано посебним научним методама и то:

- аналитичко-синтетичком методом – у делу разматрања постојећих техника за спецификацију корисничких захтева, и утврђивању постојања веза између модела случајева коришћења и жељеног корисничког интерфејса,
- метода апстракције и конкретизације – у делу који се бави трансформацијом семантички богатог модела корисничких захтева у извршиви програмски код корисничког интерфејса.

Од општих научних метода биће примењене метода моделовања у делу који се бави уочавањем карактеристика корисничког захтева значајних за дефинисање семантички богате спецификације способне за трансформацију у извршиви програмски код.

За доказивање успешности приступа развоју који ће бити изложен у овом раду биће спроведена анкета као техника методе научног испитивања, и биће спроведен експеримент на начин да се група искусних програмера подели на експерименталну и

контролну групу. Експериментална група би развијала и мењала софтверски систем коришћењем алата који је резултат овог истраживања, док би контролна група развијала и мењала систем на уобичајен начин.

Метода студија случаја, као техника за прикупљање података, биће коришћена како би се на реалном примеру показала успешност коришћења генератора у развоју и приказао целокупан процес развоја софтверског система предложеним приступом од фазе прикупљања и спецификације корисничког захтјева до имплементације софтверског система.

5. ОЧЕКИВАНИ НАУЧНИ ДОПРИНОС

- Преглед различитих техника спецификације корисничких захтева, техника за креирање модела података и значај прототиповања
- Преглед и упоредна анализа постојећих приступа и алата који аутоматизују процес имплементације корисничког интерфејса
- Идентификација и образложење критеријума које генератор корисничког интерфејса мора да задовољава како би могао бити прихваћен од стране ширег круга корисника у односу на постојеће генераторе
- Откривање веза између модела случајева коришћења и будућег корисничког интерфејса
- Откривање правила пресликавања између корисничког интерфејса развијеног за различите типове апликација (desktop, web и мобилне апликације)
- Дефинисање семантички богатог модела корисничких захтева заснованог на моделу случајева коришћења и информацијама везаним за кориснички интерфејс како би на основу њега било могуће не само пројектовање и имплементација корисничког интерфејса, већ и аутоматизација овог процеса.
- Аутоматизација процеса креирања корисничког интерфејса апликације у различитим имплементационим технологијама уз задовољење дефинисаних критеријума
- Идентификација скупа могућих шаблона корисничког интерфејса којим се различити кориснички захтеви могу приказати задржавајући жељену функционалност.
- Убрзавање процеса развоја корисничког интерфејса, а тиме и читавог софтверског система коришћењем резултата овог истраживања.

6. ПЛАН ИСТРАЖИВАЊА И СТРУКТУРА РАДА

1. Увод
2. Методологија
3. Значај проблематике рада и постојећи приступи који се баве проблемом аутоматизације процеса имплементације корисничког интерфејса
4. Дефинисање критеријума које генератор корисничког интерфејса апликације треба да задовољи
5. Упоредна анализа постојећих алата за генерисање корисничког интерфејса у односу на дефинисане критеријуме
6. Анализа различитих техника спецификације софтверских захтјева и значај прототиповања
7. Анализа елемената (структуре и садржаја) корисничког интерфејса различитих типова апликација
8. Анализа различитих нотација и техника за креирање и спецификацију модела података
9. Уочавање веза између модела случајева коришћења, модела података и будућег корисничког интерфејса
10. Уочавање шаблона корисничког интерфејса којим се различити софтверски захтјеви могу приказати
11. Дефинисање семантички богатог модела софтверских захтјева заснованог на моделу случајева коришћења, моделу података и информацијама везаним за кориснички интерфејс
12. Аутоматизација процеса трансформације предложеног модела у извршиви програмски код
13. Испитивање предности коришћења генератора у односу на традиционалне приступе пројектовању и имплементацији корисничког интерфејса
14. Студијски примјер
15. Приказ даљих праваца истраживања
16. Закључак

7. ЗАКЉУЧАК И ПРЕДЛОГ

Предложена тема докторске дисертације, под називом „Аутоматско генерисање корисничког интерфејса апликације засновано на случајевима коришћења”, по предмету истраживања, циљевима и методологији истраживања, има научну заснованост и одговара нивоу докторске дисертације. Предложена истраживања су значајна са теоријског и практичног становишта.

Кандидат мр Илија Антовић испуњава услове за подношење пријаве и израду ове докторске дисертације.

На основу изнетог, комисија предлаже Научно-наставном већу Факултета организационих наука да се кандидату мр Илији Антовићу одобри израда докторске дисертације под називом „Аутоматско генерисање корисничког интерфејса апликације засновано на случајевима коришћења” и да се за ментора именује др Владан Девеџић, редован професор.

8. ПОТПИСНИЦИ ИЗВЕШТАЈА

Београд, _____

др Владан Девеџић, редовни проф.,
Факултет организационих наука,
Универзитет у Београду

др Синиша Влајић, доцент,
Факултет организационих наука,
Универзитет у Београду

др Драган Бојић, доцент,
Електротехнички факултет,
Универзитет у Београду