

Politechnika Warszawska
Wydział Elektroniki i Technik Informatycznych

Warszawa, 30 maja 2018 r.

D z i e k a n a t

Uprzejmie informuję, że na Wydziale Elektroniki i Technik Informatycznych Politechniki Warszawskiej odbędzie się w dniu 19 czerwca 2018 r. publiczna obrona rozprawy doktorskiej

mgr inż. Stanisława Niepostyna

temat: „Automatyzacja projektowania systemów informatycznych w środowisku BPM z zastosowaniem modelu FBS UML”

promotor – prof. dr hab. inż. Janusz Sosnowski z Politechniki Warszawskiej

recenzenci:

dr hab. inż. Aneta Poniszewska-Marańda z Politechniki Łódzkiej

prof. dr hab. Stanisława Wrycza z Uniwersytetu Gdańskiego

Obrona odbędzie się w dniu 19 czerwca 2018 r. w sali 116 na Wydziale Elektroniki i Technik Informatycznych – Gmach im. Janusza Groszkowskiego, Warszawa, ul. Nowowiejska 15/19; początek godz. 11.00.

Po adresem: www.elka.pw.edu.pl/Wydzial/Rada-Wydzialu/Harmonogram-obron-doktorskich-streszczenia-i-recenzje zapewniony jest na stronie Wydziału dostęp do tekstów streszczenia rozprawy i recenzji, jak również do tekstu rozprawy umieszczonej w Bazie Wiedzy Politechniki Warszawskiej.

Dziekan



prof. dr hab. inż. Krzysztof Zaremba

POLITECHNIKA WARSZAWSKA

**Wydział Elektroniki i Technik
Informacyjnych**

ROZPRAWA DOKTORSKA

mgr inż. Stanisław Jerzy Niepostyn

**Automatyzacja projektowania systemów informatycznych
w środowisku BPM z zastosowaniem modelu FBS UML**

Promotor
Profesor dr hab. inż. Janusz Sosnowski

Streszczenie

Wzrastające znaczenie i rola informatyzacji w niemal każdej dziedzinie życia powoduje potrzebę budowy coraz większych i bardziej złożonych systemów informatycznych. Jednocześnie szybko zmieniające się wymagania na systemy IT prowadzą do coraz krótszych terminów ich realizacji. W procesie projektowania złożonych systemów IT bardzo istotnym zagadnieniem jest opracowanie odpowiedniej architektury oprogramowania.

Tematem rozprawy jest metoda udoskonalania projektowania złożonych systemów informatycznych, implementowanych w środowisku oprogramowania zorientowanego obiektowo (ang. *object-oriented programs*), w szczególności w środowisku BPM (ang. *Business Process Management*), z wykorzystaniem metodyki Enhanced CMDA (w skrócie nazwanej e-CMDA, gdzie CMDA oznacza Consistent Model Driven Architecture), umożliwiającej automatyzację budowy architektury oprogramowania z zastosowaniem zdefiniowanego zbioru reguł spójności (ang. *consistency rules*) dla szeregu uporządkowanych diagramów w notacji UML. Metodyka e-CMDA wykorzystuje wprowadzoną metrykę architektury oprogramowania FBS (ang. *Functional-Behaviour-Structure*), umożliwiającą obliczenie jej entropii (tzn. miary odwrotności zawartości informacyjnej).

Opracowana metodyka e-CMDA, prowadzi szybciej i efektywniej do budowy architektury oprogramowania złożonych systemów informatycznych od diagramu opisującego kontekst biznesowy systemu, aż do modelu implementowalnego w środowisku oprogramowania zorientowanego obiektowo w taki sposób, by na każdym etapie tworzenia architektury oprogramowania zachowana była spójność, kompletność i transformowalność opisu jej kluczowych elementów poprzez zobiektywizowane obliczenie miary FBS. Systemy informatyczne budowane zgodnie z architekturą oprogramowania opracowaną według e-CMDA budowane są szybciej, wydajniej, z mniejszą liczbą błędów, z możliwością rozbudowy architektury oprogramowania w trakcie budowy i wdrożenia systemów IT. Opracowana metodyka została zweryfikowana na szeregu projektów komercyjnych.

Słowa kluczowe: *procesy biznesowe, architektura oprogramowania, spójność, kompletność i transformowalność diagramów UML, entropia diagramów UML, wymiary architektury oprogramowania, FBS, UML, BPM.*

**KWESTIONARIUSZ – RECENZJA ROZPRAWY DOKTORSKIEJ DLA RADY
WYDZIAŁU ELEKTRONIKI I TECHNIK INFORMACYJNYCH
POLITECHNIKI WARSZAWSKIEJ**

Tytuł rozprawy: Automatyzacja projektowania systemów informatycznych w środowisku BPM z zastosowaniem modelu FBS UML

Autor rozprawy: mgr inż. Stanisław Jerzy Niepostyn

Promotor rozprawy: prof. dr hab. inż. Janusz Sosnowski

1. Jakie zagadnienie naukowe jest rozpatrzone w pracy /teza rozprawy/ i czy zostało ono dostatecznie jasno sformułowane przez autora? Jaki charakter ma rozprawa (teoretyczny, doświadczalny, inny)?

Tematyka rozprawy dotyczy zagadnień, związanych z procesem projektowania systemów informatycznych, w szczególności problemu specyfikacji architektury oprogramowania oraz próby automatyzacji tego procesu poprzez zastosowanie autorskiego rozwiązania opartego na standardzie UML.

Proces budowy dużych i złożonych systemów informatycznych, pociągający za sobą realizację rozbudowanych, a często i skomplikowanych projektów informatycznych wymaga wprowadzenia i stosowania systematycznych i jasno sformułowanych metod projektowania tychże systemów. Istotnym elementem tych metod jest zagadnienie projektowanie architektury systemu. Problem ten jest dostrzegany w literaturze, jednakże nadal brakuje systematycznego podejścia do projektowania architektury złożonych i rozbudowanych systemów informatycznych.

Ponadto, istotnym problemem pojawiającym się w procesie budowy złożonych systemów informatycznych jest konieczność zapewnienia kompletności tworzonej architektury oprogramowania oraz spójności jej poszczególnych elementów. Jest to spowodowane powszechnie przyjętym sposobem projektowania oprogramowania przy użyciu wielu modeli, prezentujących oddzielne aspekty, zagadnienia lub punkty widzenia, perspektywy tworzonego oprogramowania. Z jednej strony takie podejście przyspiesza i znacznie ułatwia proces budowy systemu, ale z drugiej strony generuje brak kompletności i spójności poszczególnych modeli, widoków, perspektyw projektowanego rozwiązania.

W związku z powyższym autor rozprawy podjął próbę opracowania metody projektowania złożonych systemów informatycznych, w szczególności architektury tychże systemów, implementowanych w środowisku zorientowanym obiektowo, wykorzystując modele standardu UML.

Przedłożona do recenzji rozprawa ma 167 stron, została podzielona na 7 rozdziałów, w tym Wstęp i Podsumowanie. Wykaz bibliografii został przedstawiony na 11 stronach (niestety poszczególne pozycje bibliograficzne nie zostały ponumerowane przez Autora), w tym Doktorant jest samodzielnym autorem 1 pozycji, a współautorem 15 pozycji.

Rozdział 1 rozprawy jest wstępem do pracy, prezentuje poruszaną w niej tematykę, formułuje problem badawczy, uzasadnia celowość podjęcia rozpatrywanej tematyki, przedstawia tezę rozprawy oraz opisuje jej strukturę. Rozdział 2 prezentuje aktualny stan wiedzy w zakresie metod budowy oprogramowania z wykorzystaniem modeli opisujących architekturę oprogramowania oraz w zakresie reguł spójności i kompletności wybranych diagramów UML, a także zastosowania metryk złożoności oprogramowania. W rozdziale 3, Autor przedstawia notację użytą do opisu reguł spójności i kompletności modeli UML. Rozdział 4 zawiera prezentację autorskich definicji spójności i kompletności, dotyczących architektury tworzonego oprogramowania oraz metodę obliczania autorskiej metryki architektury oprogramowania FBS (*Functional Behaviour Structure*). Algorytm budowy zaawansowanej architektury oprogramowania e-CMDA (*Enhanced Consistency Model Driven Architecture*), sterowany oceną spójności i kompletności z wykorzystaniem metryki FBS został zaprezentowany w rozdziale 5. Rozdział 6 to krótki opis projektów systemów informatycznych, zrealizowanych zgodnie z zaproponowaną metodą e-CMDA. Posumowanie pracy, wnioski oraz perspektywy rozwoju poruszanego tematu przedstawiono w rozdziale 7.

Teza rozprawy została sformułowana następująco: „*metodyka e-CMDA (Enhanced Consistency Model Driven Architecture) pozwala systematycznie i efektywnie oceniać oraz budować spójną i kompletną architekturę oprogramowania systemów informatycznych przez:*

- *Zastosowanie opracowanego zestawu oryginalnych reguł spójności do transformacji kluczowych elementów szeregu uporządkowanych diagramów UML.*
- *Generowanie (transformacji) szeregu uporządkowanych diagramów projektowych UML, który opisuje architekturę oprogramowania systemu informatycznego od modelu opisującego kontekst biznesowy systemu, aż do modelu implementowalnego w środowisku oprogramowania zorientowanego obiektowo.*
- *Analizę spójności, kompletności i transformowalności zarówno diagramów, jak i reguł spójności przy wykorzystaniu oryginalnej metryki FBS.”*

Uważam, że teza rozprawy oraz obszar naukowy rozprawy zostały określone jasno i precyzyjnie. Praca ma charakter teoretyczno-implementacyjny. Część teoretyczna obejmuje:

- przeprowadzenie analizy dostępnych źródeł literaturowych na temat poruszanych w rozprawie zagadnień, dotyczących procesu projektowania systemów informatycznych, w szczególności problemu specyfikacji architektury oprogramowania, w tym spójności i kompletności elementów architektury,
- opracowanie autorskich definicji spójności i kompletności, dotyczących architektury tworzonego oprogramowania,
- opracowanie autorskiej metryki architektury oprogramowania FBS oraz
- opracowanie algorytmu budowy zaawansowanej architektury oprogramowania e-CMDA, opartego na zbiorze diagramów UML, opisujących architekturę oprogramowania w perspektywie kontekstowej, biznesowej, systemowej i implementacyjnej.

Część implementacyjna obejmuje stworzenie prototypu narzędzia „e-CMDA Tool”, służącego do generowania architektury oprogramowania prostych przykładów systemów informatycznych, a dzięki temu zweryfikowanie poprawności podejścia e-CMDA, składającego się przede wszystkim z algorytmu e-CMDA i metryki FBS.

Tematyka rozprawy sytuuje ją w obszarze badawczym, związanym z poszukiwaniem efektywnych metod, dotyczących projektowania systemów informatycznych, w szczególności specyfikacji architektury oprogramowania złożonych systemów informatycznych z wykorzystaniem standardu UML.

Uważam, że podjęcie tematu pracy doktorskiej w rozpatrywanym obszarze i zakresie jest celowe i w pełni uzasadnione potrzebą zapewnienia i zwiększania jakości współczesnie tworzonego oprogramowania, a w szczególności dużych, złożonych oraz dynamicznych w swoim działaniu systemów informatycznych. Jest to w pełni uzasadnione ze względów teoretycznych, poznawczych i praktycznych na tle obecnego stanu wiedzy w rozpatrywanym obszarze.

2. Czy w rozprawie przeprowadzono w sposób właściwy analizę źródeł / w tym literatury światowej, stanu wiedzy i zastosowań w przemyśle / świadcząco o dostatecznej wiedzy autora. Czy wnioski z przeglądu źródeł sformułowano w sposób jasny i przekonujący?

Autor przedstawił analizę źródeł w rozdziale 2 i częściowo w rozdziale 3 rozprawy. Obejmuje ona tematykę modelowania architektury oprogramowania z wykorzystaniem odpowiednich modeli architektonicznych, w tym modelu koncepcyjnego architektury oprogramowania, prezentując podstawowe pojęcia, dotyczące procesu tworzenia i rozwoju oprogramowania. Autor dyskutuje problem tworzenia oprogramowania zarówno w klasycznych, ciężkich metodach, jak i w metodach/metodykach lekkich. Omawiane są problemy, dotyczące zapewnienia spójności architektury oprogramowania, między innymi poprzez podanie przykładowych, istniejących w literaturze definicji niespójności, niespójności modeli, niespójności diagramów UML oraz definicje spójności układu diagramów. Dyskutowane są rodzaje niespójności oraz ich wymiary. Następnie podana została definicja niespójności według Spanoudakisa i Zismanna, która stanowi punkt wyjścia do dalszych rozważań w prezentowanym temacie.

Autor omawia również reguły spójności, które można zastosować dla diagramów standardu UML, prezentując i zestawiając w formie tabeli istniejące w literaturze wybrane reguły spójności dla diagramów UML. Według autora istniejące reguły spójności służą jedynie do weryfikacji, a nie do budowy diagramów, nie mają powiązania z metodyką wytwarzania oprogramowania; nie tworzą razem spójnego zbioru reguł dla konkretnej architektury oraz są przedstawiane w różnorodny sposób bez zastosowania jednoznacznej i jasnej symboliki.

Kolejnym poruszonym zagadnieniem jest kompletność architektury oprogramowania, ze szczególnym uwzględnieniem kompletności danych prezentowanych na poszczególnych diagramach UML. Definicje kompletności architektury są niejednoznaczne i subiektywnie oceniane przez ekspertów.

Na koniec omawiane jest zagadnienie metryk architektury oprogramowania, które według Autora służą wyłącznie do porównywania złożoności diagramów UML, bez możliwości wyznaczania atrybutów jakościowych. Ponadto, nie istnieją metryki złożoności oprogramowania, umożliwiające ocenę wyników na wyższym poziomie abstrakcji.

W rozdziale 3 Autor prezentuje podstawowe pojęcia, dotyczące pseudokodu, wyrażeń regularnych. Przedstawia wykaz znaczeń wybranych diagramów UML oraz symbole i skróty diagramów projektowych, a także oznaczenia wybranych elementów diagramów UML, które są następnie wykorzystywane w opisie zaproponowanego w pracy rozwiązania. Ponadto, zaprezentowano notację reguł spójności oraz zestawienie wybranych z literatury reguł spójności diagramów UML.

Przeprowadzona analiza obecnego stanu wiedzy w poruszanej tematyce oraz istniejących rozwiązań wskazuje, że autor poprawnie rozumie problemy, związane z tematem rozprawy. Posiada wiedzę na temat rozwiązań krajowych i światowych w poruszonym zakresie. Co prawda część istotnych zagadnień oraz problemów została jedynie zasygnalizowana, jednakże taka forma pozwala mieć nadzieję, że Autor rozprawy posiada również wiedzę szczegółową na ich temat.

3. Czy autor rozwiązał postawione zagadnienie, czy użył właściwej do tego metody i czy przyjęte założenia są uzasadnione?

W moim przekonaniu Autor w wystarczającym stopniu przeanalizował aktualny stan wiedzy w rozpatrywanym temacie, jasno sformułował problemy, a następnie w zadawalającym stopniu rozwiązał je w swojej pracy.

Zaproponowane podejście e-CMDA, bazując na opracowanych regułach spójności, oferuje pełny zestaw diagramów projektowych w czterech perspektywach: kontekstowej, biznesowej, systemowej, implementacyjnej. Stanowi to zatem podstawę do stworzenia pełnego i kompletnego projektu systemu informatycznego, stworzonego w użyciem standardu UML. Ponadto, zamiast ręcznego konstruowania diagramów oraz zmusznej ich weryfikacji na podstawie reguł spójności, opracowana metoda umożliwia, na podstawie wyboru autorskich reguł spójności, automatyczne generowanie spójnych i kompletnych diagramów projektowych. Dodatkowo, w sytuacji modyfikacji elementów tych diagramów zapewniam, poprzez wyliczenie metryki FBS tych diagramów, ich spójność, kompletność i transformowalność.

Proponowane rozwiązanie przedstawiono na poziomie teoretycznym oraz wykonano prototyp narzędzia, umożliwiający praktyczne zastosowanie opracowanej metody, co pozwala ocenić jej efektywności oraz porównać z istniejącymi rozwiązaniami.

Ponadto, do rozprawy dołączono płytę CD, na której zamieszczono „dodatki”, opisujące zrealizowane prace autora:

- Wykaz reguł spójności proponowanych w literaturze oraz zestawienie autorskich reguł spójności w algorytmie e-CMDA.
 - Zasady dobierania wag elementów diagramów UML i weryfikacji metryki FBS.
 - Przykład realizacji algorytmu e-CMDA dla przykładowego systemu informatycznego Office.
 - Opis diagramów architektury oprogramowania wybranych zrealizowanych projektów informatycznych.
 - Opis prototypowego narzędzia e-CMDA Tool, użytego do praktycznego zastosowania zaproponowanego podejścia e-CMDA.
- Dodatki te stawiają bardzo istotny wkład autora w rozpatrywanym temacie.

4. Na czym polega oryginalność rozprawy, co stanowi samodzielny i oryginalny dorobek autora, jaka jest pozycja rozprawy w stosunku do stanu wiedzy czy poziomu techniki reprezentowanych przez literaturę światową?

Oryginalnym rezultatem rozprawy, a tym samym samodzielnym i oryginalnym dorobkiem Autora według mojej oceny jest opracowanie metody wspomagającej projektowanie systemów informatycznych, która umożliwia automatyzację procesu budowy architektury oprogramowania z zastosowaniem zdefiniowanego zbioru reguł spójności dla zestawu uporządkowanych diagramów projektowych w notacji UML. W szczególności do oryginalnych wyników uzyskanych przez Autora zaliczam:

- Opracowanie algorytmu e-CMDA, umożliwiającego budowę architektury złożonych systemów informatycznych, dzięki zastosowaniu opracowanego zestawu reguł spójności, dotyczących diagramów standardu UML.
- Sformułowanie zestawu reguł spójności dla elementów wybranych diagramów UML, które stanowią podstawę dla projektowania zbioru powiązanych diagramów UML, opisujących architekturę oprogramowania w perspektywie kontekstowej, biznesowej, systemowej i implementacyjnej oraz opracowanie notacji dla prezentowania reguł spójności w postaci wyrażen regularnych.

- Zbudowanie metryki złożoności architektury oprogramowania FBS, która pozwala określić miarę kompletności, spójności i transformowalności diagramów UML.
- Wyznaczenie wag ponad 30 elementów UML dla trzech wymiarów architektury oprogramowania (funkcje systemu, zachowanie i struktura) w wybranych diagramach UML.

Dorobek autora powiększa implementacja zaproponowanych rozwiązań poprzez stworzenie prototypu narzędzia wykorzystującego algorytm e-CMDA do konstruowania architektury oprogramowania w sposób zautomatyzowany. Metoda e-CMDA została ponadto zweryfikowana poprzez zastosowanie jej w rzeczywiste realizowanych systemach informatycznych ze świata biznesu.

pozytywnie oceniam wszystkie wymienione elementy pracy.

5. Czy autor wykazał umiejętność poprawnego i przekonującego przedstawienia uzyskanych przez siebie wyników / zwięzłość, jasność, poprawność redakcyjna rozprawy/?

Sposób prezentacji podejmowanych w pracy zagadnień jest na ogół jasny i zrozumiały, chociaż Autor stosuje bardzo dużo różnorodnych symboli na oznaczenie poszczególnych diagramów oraz ich elementów. Wprowadzone symbole są bardzo ogólne, często nieintuicyjne, ich nazwy są z reguły skrótami od pełnych nazw, ale w niektórych kolejność poszczególnych słów jest inna niż kolejność liter w symbolu.

Ponadto, część zagadnień, a także wyników pracy Autora poruszona jest w rozprawie bardzo pobieżnie. Szczegółowy opis wyników pracy jest zawarty w załącznikach, które znajdują się tylko na płycie CD dołączonej do pracy.

Tekst rozprawy został przygotowany starannie. Strona redakcyjna nie budzi większych zastrzeżeń – moje uwagi zostały przedstawione w kolejnym punkcie recenzji.

6. Jakie są słabe strony rozprawy i jej główne wady?

Analiza rozprawy nasunęła mi kilka uwag krytycznych:

- Niejasne, niejednoznaczne użycie słowa/terminu „funkcjonalne”, „funkcjonalności” (np. str. 31, 122); w tym stosowanie sformułowania „wymiar funkcjonalności” – np. str. 31, 34, 99, 101, 145; sformułowania „charakter funkcjonalny” str. 69.
- Forma niektórych zdań nasuwa wątpliwości stylistyczne, np. str. 32.
- Niepoprawne użycie słowa „funkcjonalność” – np. str. 24, 34.
- Autor używa języka potocznego oraz żargonu, np. „wymagania na system IT” str. 12, „wymagania na system” str. 33, 39.
- Zbyt częste i niepoprawne z punktu widzenia zasad interpunkcji, a przez to zbyt rażące, użycie słowa „czy”.
- Niejasne sformułowanie: „reguła dotyczy jednego (każdego) przypadku użycia”, str. 59.
- Zbyt ogólne i przez to niezrozumiałe nazwy niektórych rozdziałów i podrozdziałów oraz sekcji pracy.
- Brak spisu stosowanych w pracy symboli z objaśnieniem ich znaczenia – Autor stosuje bardzo dużo różnorodnych symboli na oznaczenie poszczególnych diagramów oraz ich elementów – symbole są bardzo ogólne, często nieintuicyjne (często są to skróty od nazw poszczególnych diagramów, podanych albo w języku angielskim, albo w języku polskim, ale niekoniecznie słowa są w odpowiedniej kolejności). Ponadto, część elementów jest określona poprzez kolejne litery alfabetu,

co dodatkowo nie ułatwia zapamiętanie tych symboli w trakcie zapoznawania się z kolejnymi częściami rozprawy.

- Tabela 2 – przyjęte symbole wybranych elementów diagramów UML nie są intuicyjne, w żaden sposób nie są związane z nazwami elementów – powoduje to trudności w zapamiętaniu i poprawnym odwoływaniu się do tych symboli w trakcie lektury rozprawy.
 - Numeracja rysunków i tabel niezgodna z ogólnie przyjętymi zasadami dla rozpraw.
 - Brak mocnego uzasadnienia zastosowania entropii dla opisu reguł spójności architektury oprogramowania.
 - Na str. 80 podano wagi dla elementów diagramu maszyny stanowej – nie jest jasne, na jakiej podstawie wyliczono takie wartości wag. Wyjaśnienie znajduje się w Dodatku B, zamieszczonym na płycie CD, ale czy nie powinno znaleźć się w treści rozprawy choć w skróconej wersji?
 - W tabeli 4 oraz na stronach 72 i 73 podano wartości wag FBS wybranych elementów UML – nie jest jasne na podstawie jakich zasad lub reguł wyliczane są takie wartości wag. Jaka jest dopuszczalna skala tych wartości?
 - Stosowana przez autora „metoda Abtran” powinna być krótko opisana w pracy.
 - Tabela 7 zawiera spis projektów IT wraz z informacjami na temat czasu realizacji i liczby stworzonych poszczególnych diagramów UML – w treści samej rozprawy nie jest niestety wyjaśnione, co to są za projekty, kto je realizował, jaka jest ich wiarygodność oraz ich danych, a także jaka jest ich poprawność.
 - Str. 84, trzeci akapit – nieprecyzyjność w opisie wartości.
 - Nieprecyzyjne stosowanie określeń „rozdział/podrozdział/sekcja”, dotyczących poszczególnych części struktury rozprawy – np. strony 29, 32, 35, 39, 47, 49, 50, 58, 60, 62, 66, 70, 73, 83, 91, 99. Wprowadza to niepotrzebny chaos do pracy.
 - Brak spisu rysunków i spisu tabel na końcu pracy. Brak indeksu stosowanych w pracy symboli, których jest wiele.
- Przedstawione uwagi nie obniżają, jednakże mojej pozytywnej oceny pracy.

7. Jaka jest przydatność rozprawy dla nauk technicznych?

Opracowana metoda e-CMDA umożliwia automatyzację procesu budowy architektury oprogramowania złożonych systemów informatycznych, począwszy do diagramu opisującego kontekst biznesowy systemu, a skończywszy na diagramie sekwencji, diagramie komponentów i diagramie wdrożeń, reprezentujących perspektywę implementacyjną. W zaproponowanym podejściu określono ściśle tworzone diagramy, które powiązane ze sobą tzw. regułami spójności (łączyącymi elementy z różnych diagramów), jakie powinny zostać zrealizowane w celu sprawnej i wydajnej budowy oprogramowania dowolną techniką (MDA, BPM, metodyki klasyczne, metody zwinne). Jednocześnie zaproponowana metoda umożliwia systematyczne tworzenie zbioru diagramów UML, które stanowią kompletną, spójną i transformalną architekturę oprogramowania (na każdym etapie jej tworzenia), od diagramów perspektyw kontekstowej do diagramów perspektyw implementacyjnej.

Stworzona metoda została wykorzystana przez autora w praktyce do opracowania architektury komercyjnych systemów informatycznych. Według zamieszczonej przez Autora informacji, przyspieszyła ona proces opracowywania architektury oprogramowania.

Wyniki rozprawy mają zatem znaczenie dla dalszego rozwoju nauki w dziedzinie inżynierii oprogramowania oraz posiadają wartość aplikacyjną dla rzeczywiste realizowanych projektów informatycznych.

8. Do której z następujących kategorii Recenzent zalicza rozprawę:

- a. niespełniająca wymagań stawianych rozprawom doktorskim przez obowiązujące przepisy
- b. wymagająca wprowadzenia poprawek i ponownego recenzowania
- c. spełniająca wymagania
- d. spełniająca wymagania z wyraźnym nadmiarem
- e. wybitnie dobra, zasługująca na wyróżnienie

Przedstawioną mi do recenzji rozprawę zaliczam do kategorii „c” – rozprawa spełnia wymagania określone przez obowiązujące przepisy. Tym samym wnoszę o dopuszczenie Pana mgr inż. Stanisława Niepostyna od publicznej obrony.



Recenzja

pracy doktorskiej mgr. inż. Stanisława Jerzego Niepostyna
pt. „Automatyzacja projektowania systemów informatycznych w
środowisku BPM z zastosowaniem modelu FBS UML”,
napisana pod kierunkiem
promotora prof. dr hab. inż. Janusza Sosnowskiego
wydanej na Wydziale Elektroniki i Technik Informatycznych,
Politechniki Warszawskiej w 2017 r

1. UWAGI OGÓLNE

Recenzowana rozprawa liczy 167 stron i składa się ze Wstępu, czterech rozdziałów, przykładów wykorzystanie metodyki e—CMDA, Literatury , Podsumowania, a także 5 Dodatków, czyli załączników - w postaci elektronicznej na CD-ROMie. Praca ma niestandardową numerację poszczególnych części, mianowicie Wstępowi przypisano numer 1, Przykładowi numer 6, a Podsumowaniu numer 7. Wstępom i zakończeniom, podsumowaniom prac doktorskich nie powinno nadawać się numerów, natomiast przykłady z wykorzystania metodyki e-CMDA powinny być załącznikami.

Pracę uważam za kreatywną w związku ze zdefiniowanym problemem badawczym i zastosowanymi oryginalnymi metodami, natomiast znacząco opóźnioną w czasie w wymiarze współczesnej inżynierii oprogramowania. Autor koncentruje się wokół reguły spójności w języku UML. Język ten był dominantą modelowania systemów informatycznych przez pierwszą dekadę XXI wieku. Obecnie raczej rzadko jest używany w analizie i projektowaniu systemów informatycznych, poza dwoma rodzajami diagramów, tj. przypadków użycia a także klas w projektowaniu baz danych. UML w znacznej części został zastąpiony przez metodyki adaptacyjne (Agile) zwane „zwinnymi” i „żwawymi”, których niezwykle pobieżny opis znajduje się na stronie 21 recenzowanej pracy. Wiem o tym jako autor książek i artykułów i referatów na temat języka UML w językach polskim i angielskim. Stąd praca stanowi niewątpliwie interesującą koncepcję naukową, jednak bez większego znaczenia praktycznego obecnie.

Informatycy tworzyli, we wspomnianym okresie, projekty systemów wykorzystując 14 rodzajów diagramów UML i opracowaną w tym celu metodykę RUP. Podstawowe i niezbędne w procesie tworzenia systemów informatycznych reguły spójności i kompletności pomiędzy diagramami a także ich elementami, zawarte są we wspomagających w praktyce te procesy pakietach CASE (computer-aided systems engineering), jak np. Enterprise Architect, w których spójność i kompletność są automatyczne monitorowanie, a przypadki niespójności i braku kompletności modelu UML – na bieżąco sygnalizowane. Stąd proponowane przez Autora sformalizowane reguły definiowania spójności w dużej mierze nie były, a obecnie wątpliwe czy będą – składnikiem modelowania procesów i prac projektowych. Natomiast, mogłyby zostać wbudowane we współczesne narzędzie testowania oprogramowania. W tym celu należy stworzyć jednak narzędzie typu e-CMDA, po znaczącej jego modyfikacji, w zasadzie od podstaw.

Krytycznie należy ocenić aspekty metodyczne pracy. Niezbyt precyzyjny jest sam tytuł pracy. Dominantami recenzowanego doktoratu są bowiem język UML oraz architektura oprogramowania. To drugie określenie nie występuje w tytule. Natomiast występuje tam akronim BPM, czyli modelowanie procesów biznesowych, a na ten temat nie ma w pracy nawet podrozdziału. Autor nie formułuje celu pracy a to należy do standardów prac naukowych. Za to jest obszerna (ok 0,5 strony), stąd mało precyzyjna - teza badawcza, w której dla odmiany występuje określenie architektury oprogramowania. Zupełnie pominięto omówienie podstaw metodologicznych pracy w szerszym kontekście, poza prezentacją szczegółowych metod własnych.

2. OCENA MERYTORYCZNA

Praca składa się z czterech rozdziałów, numerowanych od 2 do 5. Rozdział pierwszy poświęcony jest problemom modelowania architektury oprogramowania. Odwoływanie się tam do wzorców architektury oprogramowania do pozycji literatury z lat nawet 60-tych, nie jest szczęśliwym pomysłem dla współczesnej pracy informatycznej. Z kolei, oryginalnym pomysłem jest analiza, prezentacja i definicja problemów niespójności oprogramowania. Znaczące pod tym względem jest zestawienie, analiza porównawcza reguł spójności pomiędzy diagramami UML. Rys. 2 i jego opis są niespójne. Część wyrażen na rys.2 nie została przetłumaczona.

W rozdziale drugim, zatytułowanym „Podstawowe pojęcia i notacje”, Autor wykazał się profesjonalną znajomością języka UML, wprowadzając własny wykaz oznaczeń wybranych diagramów UML. Rozwiązania te jednak podporządkowują proponowaną architekturę oprogramowania językowi UML, czyniąc ją mniej uniwersalną. Uznanie budzi podstawa heurystyczna autorskiej metodyki e-CDM w postaci analizy i

opracowania ponad 2000 diagramów UML w 20 projektach. Była to wyjątkowa szansa, ze względu na znacząco malejące zainteresowanie językiem UML obecnie.

Bardzo sformalizowany jest rozdział trzeci zatytułowany "Metryka FBS, definicje spójności, kompletności, transformowalności". Można zastanawiać się nad celowością badania entropii diagramu UML, zwłaszcza rozbudowanymi formalizmami matematycznymi. Oryginalnym i najistotniejszym osiągnięciem Autora jest metryka FBS, zaprojektowana i zweryfikowana metodą Abran, w tym zestawienie wag FBS wybranych elementów diagramów UML. Rozdziału dopełniają definicje spójności, kompletności i transformowalności architektury oprogramowania

Ostatni z rozdziałów pt. „Zaawansowana struktura oprogramowania sterowana spójnością” zawiera poszerzenie tematyki rozdziału trzeciego, w aspekcie algorytmu e-CMDA. Szczególny nacisk położono tu na proces użytkowania algorytmu e-CMDA w tworzeniu architektury oprogramowania w trzech perspektywach: kontekstowej, biznesowej, systemowej i implementacyjnej. Każda z nich ujęta jest w sekwencje od 2 do 3 szczegółowych kroków – raczej chyba etapów czy faz.

Zaskakuje odwoływanie się Autora do publikacji odległych czasowo (zdarzają się prace z lat sześćdziesiątych – sic!). W zasadzie bibliografia współczesnej, informatycznej pracy doktorskiej powinna zawierać publikacje z czasopism z Impact Factorem oraz znaczących międzynarodowych konferencji informatycznych, przede wszystkim w języku angielskim, głównie z ostatnich kilku lat. Niedostosowanie się do tej zasady jest bardzo niekorzystne dla pracy dr, czyniąc ją mniej lub mało merytorycznie aktualną.

OCENA FORMALNA TEKSTU

Tekst recenzowanej pracy nie należy do starannych, występują potknięcia językowe i stylistyczne. Autor nie przestrzega wielu reguł poprawnego edytowania tekstu. Należy w tym kontekście podnieść zwłaszcza kwestie wyliczeń w tekście i prezentacji rysunków. Autor stosuje zbyt często, zbyt długie, kilkustronicowe wyliczenia, np. na stronach 17 – 20, 40-42, 49 – 50, 52 – 55, 150 – 151 i inne. Takie podejście nadaje tej pracy charakter instruktażowy. Praca jest napisana w j. polskim toteż niezrozumiałe jest częste stosowanie konwencji mieszanej – polskiej i angielskiej, , np. na rysunkach 2, 12, 16. Autor stosuje również autocytowania; dobre praktyki publikacyjne – zalecają ich unikania. W literaturze występuje aż 11 publikacji autorskich Doktoranta. W niektórych przypadkach są to referaty na krajowych lub regionalnych konferencjach. W recenzjach artykułów do czasopism IF wręcz zwalcza się taką praktykę, bowiem powstają pytania: w jakim stopniu poprzednie publikacje autorskie zostały ujęte w samym doktoracie? A nie powinny być. Inne ważne pytanie, to: jaki w tej sytuacji jest nowy, innowacyjny

wkład teoretyczny Doktoranta w recenzowanym doktoracie. Co innego jeśli podstawą ubiegania się o stopień doktora jest zestaw publikacji Doktoranta.

Ponieważ specjalizuję się w naukach ekonomicznych (Informator Nauki Polskiej), w dziedzinie analizy i projektowania systemów informatycznych zarządzania, trudno mi wydać opinię, potwierdzającą poprawność wywodu matematycznego, zwłaszcza na stronach 67-70 oraz w dodatkach 1-3 na CD-ROMie. Poprawność formalną reguł i wywodu matematycznych powinny być sprawdzone przez samodzielnego pracownika naukowego z dziedziny nauk matematycznych. Takie ryzyko, ale i oczekiwanie związane jest z interdyscyplinarnymi pracami naukowymi.

4. Konkluzja

W częściach 1-3 niniejszej recenzji pracy doktorskiej Pana mgr. inż. Stanisława Jerzego Niepostyna pt. „Automatyzacja projektowania systemów informatycznych w środowisku BPM z zastosowaniem modelu FBS UML”, podniosłem szereg istotnych uwag, wskazałem braki, niedociągnięcia i istotne wątpliwości. Jednakże, tak jak zaznaczyłem na początku recenzji – praca ma cechy kreatywności i oryginalności. Autor podjął się trudnego zadania i wyzwania, stworzył własne rozwiązania i koncepcje w postaci algorytmu e-CDMA, metryki złożoności architektury FBS, narzędzia wykorzystującego algorytm e-CMDA. Mgr inż. Stanisław Jerzy Niepostyn powinien otrzymać szansę obrony proponowanych przez siebie koncepcji i rozwiązań naukowych. W nawiązaniu do Ustawy z dn. 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz stopniach i tytule w zakresie sztuki, wnioskuję o przyjęcie i dopuszczenie pracy doktorskiej autorstwa Pana mgr. inż. Stanisława Jerzego Niepostyna pt. „Automatyzacja projektowania systemów informatycznych w środowisku BPM z zastosowaniem modelu FBS UML” - do publicznej obrony.

