

STRESZCZENIA

Rozdział 1. Narzędzia wspomagające generowanie aplikacji typu „data intensive” dla środowiska webowego. Do podstawowych wyzwań inżynierii oprogramowania systemów biznesowych należą szybkość dostarczania oraz minimalizacja kosztów konserwacji. Osiągnięcie tych celów związane jest, między innymi, z minimalizacją złożoności aplikacji. W kontekście systemów, których głównym zadaniem jest przetwarzanie trwałych danych istotnym czynnikiem złożoności jest mechanizm trwałości. Jest on najczęściej realizowany poprzez zmapowanie modelu obiektowego aplikacji na model danych w relacyjnej bazie danych. Przyspieszenie procesu budowy aplikacji odbywa się poprzez wykorzystanie narzędzi automatyzujących proces mapowania oraz narzędzi generujących elementy interfejsu użytkownika umożliwiające operacje na tych danych. Artykuł przedstawia problemy jakie, pomimo rozwoju technologii służących mapowaniu obiektowo-relacyjnemu, pojawiają się na styku niedopasowanych modeli oraz proponuje idee alternatywnego podejścia do problematyki trwałości, jako interesujący temat badawczy, możliwy do realizacji przy współpracy nauki i biznesu.

Rozdział 2. Wzorce logiki aplikacji – reużywalne składniki wymagań. Wzorce (projektowe, architektoniczne i inne) są często stosowanym sposobem na redukcję czasu wykonania i poprawę jakości systemów oprogramowania. Także w zakresie sposobu funkcjonowania aplikacji można wyróżnić pewne powtarzalne rozwiązania, które obecnie nie są jednak powszechnie reużywane. Niniejszy rozdział uzupełnia tę lukę, przedstawiając bibliotekę zawierającą tzw. wzorce logiki aplikacji. Elementy biblioteki opisane są precyzyjnie zdefiniowanym poprzez metamodel językiem, który wykorzystuje konstrukcje z języka UML, ale uzupełnia ich semantykę oraz notację. Zawarte w bibliotece typowe czynności wykonywane przy wsparciu systemów informatycznych, prezentowane są w sposób pozwalający na elastyczne ich stosowanie. Szablonom interakcji użytkownik – system informatyczny towarzyszą statyczne, abstrakcyjne modele dziedziny związane z tymi wzorcami. Biblioteka wzorców jest wspierana przez otwarte narzędzie ReDSeeDS, umożliwiające łatwą aplikację wzorców w projektach rozwijanych zgodnie z zasadami modelowania oprogramowania.

Rozdział 3. Moduł wspomagający projektowanie i wytwarzanie złożonych systemów informatycznych opartych na Spring i OSGi. Opracowanie opisuje moduł programowy wspierający projektowanie i wytwarzanie złożonych systemów informatycznych, działających w oparciu o technologię Spring Framework i OSGi. Przedstawione rozwiązanie stanowi rozszerzenie funkcjonalności narzędzia CASE – Enterprise Architect. W pracy przedstawiono koncepcję architektury OSGi. Zaproponowano odpowiednie, nowe rozszerzenia języka UML, udostępnione w narzędziu Enterprise Architect, oparte na mechanizmie profili UML, uwzględniające specyfikę zastosowania Spring i OSGi. Pokazano przykład zastosowania omawianego rozszerzenia w ramach projektowania systemu, tzn. modelowania z użyciem diagramów klas, pakietów, komponentów. Przedstawiono rezultaty automatycznej generacji plików konfiguracyjnych, będące wynikiem działania zaproponowanego modułu programowego – wtyczki do narzędzia Enterprise Architect.

Rozdział 4. Reguły efektywnego projektowania semantycznych usług WWW. W rozdziale omówiono kluczowe zasady projektowania usług semantycznych w Internecie. Nawiązano do reguł formalizacji wiedzy za pomocą systemów ontologicznych, dla których implementacji opracowano języki programowania OWL i KIF. Odniesiono się także do sieci semantycznych, jako metody sztucznej inteligencji w kontekście założeń projektu Web 3.0. Omówiono zasady stosowania języków XML, XML Schema, RDF, RDF Schema, OWL, SPARQL, a także formatu reguł wymiany metadanych RIF.

Rozdział 5. Integracja systemów w architekturze zorientowanej na usługi. Celem niniejszego rozdziału jest analiza użyteczności architektury, zorientowanej na usługi dla integracji systemów informatycznych. W pierwszej części rozdziału zaprezentowano technologie integracyjne i stosowane rozwiązania. W szczególności omówiono architekturę zorientowaną na usługi w kontekście integracji systemów. Następnie przedstawiono technologię usług sieciowych, jako implementacja powyższej architektury. Na końcu przeprowadzono analizy i przedstawiono wnioski dotyczące zastosowania architektury zorientowanej na usługi, a w szczególności wykorzystania usług sieciowych dla integracji systemów informatycznych.

Rozdział 6. Kontrola spójności modeli UML za pomocą modelu przestrzennego DOD. Przedstawiono kontrolę spójności trzech podstawowych diagramów UML za pomocą modelu przestrzennego DOD (Diagram Obiegu Dokumentów). Model przestrzenny DOD umożliwia zaprojektowanie struktury aplikacji, jej zachowania i funkcjonalności. Taki sposób modelowania docelowego systemu umożliwia otrzymanie zestawu wybranych i spójnych diagramów UML, pozwalający na pełny i kompletny opis architektury w perspektywie logicznej. Wygenerowany diagram 3DOD pozwala zweryfikować spójność i kompletność poszczególnych elementów diagramów UML, z których został zbudowany.

Rozdział 7. Konsekwencje zastosowania modelowania w projektach informatycznych - badanie z udziałem praktyków. Rozdział zawiera najciekawsze wnioski z badania ankietowego, przeprowadzonego wśród przedstawicieli polskich firm informatycznych, którego celem było poznanie stanu zastosowania metod modelowania i narzędzi UML w projektach informatycznych oraz korzyści, wynikających z zastosowania tej technologii w projektach informatycznych, jak również występujących problemów, zarówno przy zastosowaniu modelowania, jak i przy braku zastosowania modelowania i narzędzi UML.

Rozdział 8. Model diofantyczny problemu osiągalności. Praktyka dnia codziennego obfituje w wiele różnorodnych zadań podejmowania decyzji, nawiązujących do problemów harmonogramowania cyklicznego. W wielu przypadkach, związanych z ograniczeniami powodowanymi występowaniem danych i zmiennych decyzyjnych należących do liczb całkowitych problemy te, z uwagi na swój diofantyczny charakter, są nierozstrzygalne. Problemy tego typu związane są najczęściej z harmonogramowaniem cyklicznym, występującym w zadaniach planowania rozkładów zajęć, dyżurów, rozkładów jazdy, itp. Oznacza to, że diofantyczny charakteru problemów harmonogramowania, winien znaleźć swoje odbicie w procesie programowania dedykowanych systemów wspomaganie decyzji. Uświadamiając w szczególności wynikające z tego z tego charakteru ograniczenie, sprowadzające się do konstatacji, iż struktura systemu determinuje a nie ogranicza jego zachowanie. Inaczej mówiąc, nie każde zachowanie jest dopuszczalne w danej strukturze systemu.

Rozdział 9. Zastosowanie wymagań jakościowych w procesie wytwarzania oprogramowania sterowanego modelami. Rosnąca złożoność systemów oprogramowania wymusza poszukiwanie metod umożliwiających skrócenie ścieżki prowadzącej od wymagań do projektu i implementacji. Koncepcja MDA i jej praktyczne realizacje, umożliwiają skrócenie tej ścieżki poprzez stosowanie transformacji modeli. O ile zastosowanie pewnych reguł transformacji do precyzyjnego modelu wymagań funkcjonalnych, pozwala na automatyczną generację modelu architektonicznego spełniającego te wymagania, o tyle proces ten nie zapewnia spełnienia wymagań jakościowych. Wymagania te, mając istotny wpływ na podejmowane decyzje architektoniczne, powinny stanowić czynnik decyzyjny w doborze zestawu transformacji. Rozdział ten przedstawia ideę transformacji precyzyjnej specyfikacji wymagań funkcjonalnych, sterowanej wymaganiami jakościowymi. Przedstawiona została koncepcja

rozszerzenia języka opisu wymagań RSL o język opisu wymagań jakościowych NFR Framework, połączona z ideą modularnej budowy transformacji.

Rozdział 10. Systemy MWING i IMES oraz ich wykorzystanie do badań wydajności sieci WWW.

Rozdział przedstawia dwa systemy MWING i IMES do pomiarów Internetu, które zostały zaprojektowane i wykonane pod kierunkiem autora w latach 2006-2009. Systemy zostały wykorzystane do pomiarów wydajności sieci WWW w kilku eksperymentach pomiarów aktywnych prowadzonych w stosunku do serwerów webowych zainstalowanych w publicznym Internecie oraz w środowisku PlanetLab. Zebrane dane pomiarowe posłużyły badaniom nad algorytmami predykcji czasu transferu plików pobieranych z serwisów WWW, a w tym m.in. przebadaniu z wykorzystaniem danych rzeczywistych nowego zaproponowanego przez autora podejścia *Web performance mining* do rozwiązania tego zadania predykcji, polegającego na przeprowadzeniu dwufazowej eksploracji danych (data miningu) i definiującego nowy kierunek badań w zakresie Web miningu.

Rozdział 11. Wielokryterialne podejmowanie decyzji z wykorzystaniem ocen rozmytych w środowisku OLAP.

W rozdziale sformułowano zadanie wielokryterialnego podejmowania decyzji w warunkach rozmytości sytuując problem wyboru w środowisku OLAP. Ponadto zaprezentowano sposoby rozwiązania tego zadania uwzględniające pewną relację porządku w przestrzeni zbiorów rozmytych, przy założeniu skończonej liczby decyzji dopuszczalnych. Rozmyty, wielokryterialny model decyzyjny OLAP, będący rozbudowaniem modelu rozmytego OLAP, umożliwia rozszerzenie klasycznego podejścia poprzez opis złożonych i nieprecyzyjnych zależności, łącząc standardowe podejście do problemów wspomagania decyzji z teorią zbiorów rozmytych w wielowymiarowym środowisku OLAP.

Rozdział 12. Łączenie zwinności metodyki Scrum z dojrzałością modelu CMMI.

W rozdziale przedstawiono podejście do łączenia zwinności metodyki Scrum oraz dojrzałości modelu CMMI. Na początku krótko omówiono metodykę Scrum oraz model dojrzałości CMMI. Następnie zaproponowano model odniesienia praktyk metodyki Scrum do praktyk modelu CMMI. Dla praktyk CMMI nierealizowanych częściowo bądź w pełni przez metodykę Scrum zaproponowano dodatkowe praktyki, które nie przytłaczając zwinności wprowadzają odpowiedni stopień dyscypliny. Dla wsparcia zastosowania proponowanego podejścia opracowano wspomagany kwestionariuszem i narzędziem programowym proces identyfikacji praktyk hybrydowego podejścia CMMI oraz Scrum. Zaproponowane podejście poddano walidacji w dwóch studiach przypadków z udziałem firm branży informatycznej, a także skonfrontowano z opinią ekspercką.

Rozdział 13. Zespołowe Wytwarzanie Oprogramowania. Praktyki wspierane w środowisku MS

TFS Wytwarzanie oprogramowania jest złożonym procesem a sukces przedsięwzięcia informatycznego zależy od wielu czynników. Aby uzyskać wyższą jakość końcowego produktu należy posłużyć się dyscypliną inżynierii oprogramowania. W rozdziale omówiono metodyki które mogą zagwarantować sukces projektu. Ograniczono się do metodyk zwinnych: Scrum i MSF. Następnie omówione zostało środowisko wytwarzania oprogramowania firmy Microsoft – Team Foundation Server. Wykazano w jaki sposób TFS realizuje definicje Forrester'a dotyczącą zarządzania cyklem rozwoju oprogramowania (ang. ALM) Przedstawione zostały praktyki wspierane przez środowisko. Dodatkowo przedstawione zostały doświadczenia firmy Volvo IT związane z wdrożeniem środowiska TFS w korporacji.

Rozdział 14. Zespołowa realizacja projektu Informatycznego: od pomysłu – do wdrożenia -

Prowadzenie pracy zespołowej z wykorzystaniem Internetu. Rozdział przedstawia problematykę edukację studentów informatyki w zakresie realizacji zespołowego projektu informatycznego w modelu nauczania zdalnego przez Internet. Przedstawiono w nim metody prowadzenia tego typu zajęć, zastosowane narzędzia oraz metodykę pracy ze studentami kierunku inżynieria oprogramowania.

Rozdział 15. Tworzenie systemu z wykorzystaniem współpracujących maszyn stanowych.

Budowa systemów realizujących złożoną logikę biznesową może być wspomagana przez rozwiązania inżynierii opartej na modelach. W prezentowanym podejściu działanie poszczególnych klas systemu zostało opisane przy użyciu maszyn stanowych UML. Wynikowy model integrujący różne części systemu został przetransformowany do kodu i uruchomiony razem z biblioteką implementującą wszystkie elementy specyfikacji maszyn stanowych. Do generacji kodu i realizacji maszyn stanowych wykorzystano

narzędzie FXU (ang. *Framework for eXecutable UML*), które przekształca modele klas i stanów do kodu C#. Modelowany system dotyczył realizacji mobilnych usług kontekstowych dla społeczności użytkowników sieci Internet. Główną częścią systemu był serwer obecności. Realizował on trzy podstawowe zadania: publikację statusu użytkownika, informowanie o jego zmianach oraz subskrypcję informacji o statusie innego użytkownika.

Rozdział 16. Praktyki w inżynierii oprogramowania – perspektywa pracy zespołowej. W rozdziale dokonano przeglądu praktyk zalecanych przez popularne metodyki wytwórcze oraz dokonano analizy ich użyteczności w kontekście zespołowego przedsięwzięcia programistycznego. Analiza została przeprowadzona w oparciu o wybrane metodyki z grupy metodyk zwinnych (XP, Scrum, OpenUp) oraz jedną z grupy metodyk sterowanych planami – RUP. Dodatkowo, podjęto próbę zdefiniowania znaczenia terminu praktyka (najlepsza praktyka), gdyż jego powszechne rozumienie jest nieprecyzyjne i często utożsamiane z samymi procesami wytwórczymi.

Rozdział 17. Grupowe wspomaganie decyzji w procesie zarządzania przedsięwzięciem informatycznym. Rozdział poświęcono wybranym sposobom grupowego wspomaganie decyzji (ang. group decision making, GDM) w zarządzaniu przedsięwzięciem informatycznym. Szczególną uwagę poświęcono przy tym przedsięwzięciom, związanym z korporacyjnymi systemami wspomaganie zarządzania. Uzasadniono celowość stosowania w zarządzaniu przedsięwzięciami i podano uwarunkowania grupowego wspomaganie decyzji. Przedstawiono także możliwości zastosowania do grupowego wspomaganie decyzji analizy hierarchicznej/sięciowej procesów (AHP/ANP). Ujęto przy tym różne podejścia do grupowego wspomaganie decyzji. Opis teoretyczny uzupełniono analizą, przedstawiającą praktyczne możliwości stosowania GDM w zarządzaniu przedsięwzięciami rozpatrywanego rodzaju. Wykorzystano w tym celu zagadnienie doboru odpowiedniego narzędzia zarządzania przedsięwzięciem. Uwzględniono przy tym poziomy atrybutów, opisujących przedsięwzięcie.

Rozdział 18. Integracja architektury procesowej przedsiębiorstwa i systemu wspomagającego zarządzanie klasy ERP. W rozdziale przedstawiono metody zapewnienia integracji architektury procesowej przedsiębiorstwa oraz systemu wspomagającego zarządzanie klasy ERP (ang. *Enterprise Resource Planning*). Przez „integrację” rozumiemy takie dopasowanie systemu klasy ERP do specyfiki przedsiębiorstwa, aby zagwarantować możliwie najpełniejsze wsparcie informatyczne dla procesów biznesowych zachodzących w przedsiębiorstwie, czyli wsparcie funkcjonowania przedsiębiorstwa na poziomie operacyjnym. Opisany wybór praktycznych metod i narzędzi opiera się na wieloletnim doświadczeniu własnym autorów z projektów przez nich realizowanych dla kilkudziesięciu polskich przedsiębiorstw. Prezentowane metody dotyczą: doboru najbardziej odpowiedniego dla przedsiębiorstwa gotowego systemu klasy ERP, zaprojektowania jego osadzenia w przedsiębiorstwie, a następnie jego konfiguracji i parametryzacji.

Rozdział 19. Modelowanie Usług Integracyjnych w Oparciu o Platformę OfficeObjects® Service Broker. W pracy przedstawiono platformę aplikacyjną powstałą jako wynik prac badawczych przeprowadzonych w ramach europejskiego projektu eGovBus. Ze względu na swoją elastyczną architekturę proponowana platforma umożliwia projektowanie oraz implementację generycznych usług w wielu obszarach zastosowań IT. W pracy przedstawiono architekturę platformy, omówiono główne narzędzia dostępne dla użytkownika oraz scharakteryzowano funkcjonalność proponowanego rozwiązania. Projektowanie usług integracyjnych z wykorzystaniem platformy OfficeObjects® Service Broker opiera się na dwóch podstawowych elementach – metodyce oraz zestawie wizualnych narzędzi umożliwiających zrealizowanie poszczególnych etapów tworzenia usługi. Wszystkie narzędzia użytkownika, zintegrowane są w ramach Środowiska Projektowania Usług, które jest dostępne z poziomu przeglądarki internetowej. W pracy przedstawiono także przykładową usługę integracyjną zaprojektowaną za pomocą proponowanej platformy.

Rozdział 20. Testowanie w przyrostowym i ewolucyjnym cyklu życia oprogramowania. Artykuł prezentuje doświadczenia dotyczące procesu testowania złożonego systemu internetowego rozwijanego w okresie ostatnich pięciu lat. System ten powstaje w cyklu przyrostowym i ewolucyjnym, przechodząc do kolejnych wydań. Rozróznilo wydania główne, które są poprzedzane pełnym zakresem testów regresji

oraz wydania rozszerzające, gdzie zakres testowania jest zawężony. Wyjaśniono miejsce procesu testowania w kontekście zarządzania zmianami oraz środowisko testowe i związane z nim wsparcie narzędziowe. Przedstawiono format dokumentowania przypadków testowych. Wyjaśniono zakres i problemy związane z automatyzacją testów. Przedstawiono dane prezentujące nakłady na testowanie oraz wskazujące korzyści wynikające z automatyzacji procesu testowania. Przedstawiono również plany dotyczące dalszego rozwoju zastosowanej metody testowania.

Rozdział 21. Norma PN ISO/IEC 12207 jako zabezpieczenie wdrożenia oprogramowania. W rozdziale omówiono praktyczne aspekty wykorzystania normy PN ISO/IEC 9001:2001 do zabezpieczenia interesów nabywcy dedykowanego oprogramowania. Norma może być stosowana dla umów dotyczących oprogramowania, w którym nie określono metody zarządzania projektem a wykonawca posiada certyfikat zarządzania systemem jakości zgodny z normą PN ISO/IEC 9001:2001. Rozdział zawiera opis wykorzystania normy do oceny wdrożenia, w szczególności do rozstrzygania sporów nt. przebiegu wdrożenia czy efektów wdrożenia. Norma PN ISO/IEC 12207 może również mieć zastosowanie do postępowań sądowych w celu analizy „zachowania należytej staranności” w sensie artykułu 527 kodeksu cywilnego. Potwierdzony wyrokiem sądowym brak „zachowania należytej staranności” oznacza nie tylko przegranie procesu, zwrot wydatków na wdrożenie systemu i wypłatę odszkodowania ale w przypadku projektu realizowanego jako zamówienie publiczne również wykluczenie na 3 lata z przetargów prowadzonych w oparciu o ustawę Prawo zamówień publicznych.

Rozdział 22. Opiniowanie „nowej, nowoczesnej, innowacyjnej” technologii. W rozdziale omówiono praktyczne aspekty - możliwości uzyskania dofinansowania czy też ulgi, jakie może przynieść odpowiednio inwestycja w „nową, innowacyjną” czy też „nowoczesną” technologię. Opinie dotyczące tych pojęć może wydawać m.in. samodzielne jednostki uczelni wyższej. W opiniach należy udowodnić zgodność przedmiotu opinii z odpowiednimi „legalnymi” definicjami. Przedstawiono korzyści z posiadania opinii i ideę oceny „nowej technologii” w sensie ustawy z dnia 25.02.1992r., o podatku dochodowym od osób prawnych. W rozdziale omówiono też pojęcie „innowacyjności technologii” w sensie opinii o innowacyjności, wymagana czasami przez Polską Agencję Rozwoju Przedsiębiorczości (PARP) i regionalne instytucje pośredniczące w niektórych konkursach na dofinansowania z funduszy regionalnych i Unii Europejskiej. Na zakończenie opisano, możliwość uzyskania przez zakład pracy chronionej odpisu z zakładowego funduszu rehabilitacji osób niepełnosprawnych „na wprowadzenie nowoczesnej technologii”.

Rozdział 23. Transformacja i porównanie modeli jako metoda weryfikacji zgodności implementacji z projektem architektury. W popularnej metodzie tworzenia oprogramowania, jaką jest MDE (ang. *Model Driven Engineering*) wykorzystuje się modele o różnorodnym stopniu szczegółowości. Kod wynikowy może być postrzegany jako końcowy, najbardziej szczegółowy obraz projektu. Utrzymywanie spójności między modelem projektu a kodem jest kluczowym czynnikiem powodzenia projektu programistycznego. Niestety, praktyka pokazuje, że ta spójność w czasie trwania projektu stopniowo się pogarsza. Zarządzający projektem powinni więc na bieżąco monitorować i ograniczać takie rozbieżności. W artykule przedstawiono implementację narzędzia wspierającego to zadanie. Na podstawie kodu wynikowego w Javie odtwarzany jest za pomocą inżynierii odwrotnej model, który porównywany jest z początkowym projektem systemu. W rozdziale zaproponowano także zestaw miar podobieństwa i eksperymentalnie oceniono ich efektywność.

Rozdział 24 Predykcja defektów na podstawie metryk oprogramowania – identyfikacja klas projektów. Konstrukcja modelu wskazującego elementy systemu informatycznego obarczone największą liczbą defektów, mimo iż nie jest skomplikowana, to nie jest czynnością często wykonywaną w przemysłowych procesach wytwarzania oprogramowania. Prawdopodobnie najważniejszą przyczyną tego stanu rzeczy jest znikoma ilość prac badających możliwość ponownego wykorzystania modelu w innym projekcie. Referat ten przedstawia badania w których zidentyfikowano klasy projektów, w ramach których można stosować ten sam model predykcji defektów. Postawiono hipotezę, że środowisko z którego projekt pochodzi determinuje rodzaj modelu jaki powinien zostać zastosowany, a następnie statystycznie zweryfikowano tę hipotezę, stosując modele predykcji defektów do różnych typów projektów i oceniając dokładność uzyskanej predykcji.

Rozdział 25. Analiza możliwości realizacji Modułu Rozliczalności Działań w Systemie Wspomagania Działań Operacyjno-Procesowych. Rozdział prezentuje analizę możliwości realizacji modułu odpowiedzialnego za rejestrację wszelkich działań użytkowników systemu. Przedstawiono możliwe architektury takiego modułu oraz zaprezentowano rozwiązanie zastosowane w prototypie, będącym częścią składową projektu rozwojowego, realizowanego przez Politechnikę Śląską wraz z firmą Wasko.

Rozdział 26. Weryfikacja procesów biznesowych metodą tablic semantycznych. Artykuł dotyczy zagadnień formalnej weryfikacji modeli procesów biznesowych z wykorzystaniem wnioskowania dedukcyjnego. Notacja BPMN jest szeroko wykorzystywana do opisu procesów biznesowych przedsiębiorstw, ale może także stanowić wizualizację dla języka BPEL, do którego może zostać odwzorowana, języka zgodnego z architekturą SOA. Do specyfikacji modelu biznesowego i żądanych własności wykorzystano logikę temporalną, jako dobrze opisującą kwestie żywotności i bezpieczeństwa systemu. Została zastosowana metoda wnioskowania metodą tablic semantycznych, która może przebiegać automatycznie i jest ciekawą alternatywą dla podejścia tradycyjnego, umożliwiając m.in. względnie łatwe wskazanie błędów w specyfikacji. Zaproponowano metody pozyskiwania formuł logiki temporalnej bezpośrednio z modeli BPMN utworzonych ze złożenia skończonej liczby znanych wzorców procesowych. **Słowa kluczowe:** weryfikacja, notacja BPMN, dedukcja, metoda tablic semantycznych, wzorce procesowe, ekstrakcja formuł.

Rozdział 27. Automatyczna weryfikacja modelu na etapie analizy wymagań. Artykuł dotyczy zagadnień formalnej weryfikacji przypadków użycia oraz scenariuszy przypadków użycia, zapisanych w postaci diagramów czynności z wykorzystaniem wnioskowania dedukcyjnego. Zaproponowana została metodologia, opisująca przejście od diagramów przypadków użycia do diagramów aktywności. Do specyfikacji systemu i żądanych jego własności została wykorzystana logika temporalna, natomiast wnioskowanie dedukcyjne odbywa się z wykorzystaniem metody tablic semantycznych. Zaproponowane zostały metody pozyskiwania formuł logiki temporalnej bezpośrednio ze scenariuszy przypadków użycia zapisanych w UML'owych diagramach czynności.

Rozdział 28. Indeksowanie w przetwarzaniu heterogenicznych zasobów. W artykule przedstawiono technikę ulotnego indeksowania, która dotyczy szczególnie trudnej dziedziny w kontekście optymalizacji, jaką jest przetwarzanie globalnych zapytań odnoszących się do rozproszonych heterogenicznych zasobów. W odróżnieniu od normalnych indeksów, które są strukturami nadmiarowymi przechowywanymi na serwerze, ulotny indeks jest materializowany tylko podczas wykonywania zapytania. Z tego powodu, przedstawiona technika jest skuteczna w przetwarzaniu złożonych zapytań, w których indeks jest wywoływany więcej niż jeden raz. Ulotne indeksowanie zostało zaimplementowane i przetestowane w prototypie obiektowej bazy danych ODRA.

Rozdział 29. Sterowanie modelem narzędzie do projektowania i symulacji Probabilistycznych Drzew Niezdatności z Zależnościami Czasowymi. Probabilistyczne Drzewa Niezdatności z Zależnościami Czasowymi (PDNZC) umożliwiają efektywną analizę niezawodności systemów technicznych, w których czas trwania zdarzeń awarii można opisać zmiennymi losowymi. PDNZC charakteryzują się nie tylko wysoką mocą ekspresji, ale także czytelnością i intuicją dla inżynierów, gdyż odwołują się do respektowanych standardów. Aby jednak PDNZC znalazły uznanie, należy zautomatyzować proces wyznaczania niezawodności poprzez stworzenie dedykowanego narzędzia ich tworzenia i analizy. Korzystając z transformacji międzymodelowych platformy Eclipse zbudowano komponentową aplikację sterowaną modelem do graficznego projektowania, walidacji składniowej oraz weryfikacji semantycznej drzew. Ponadto, zaprezentowano dwa podejścia do transformacji specyfikacji behawioralnej PDNZC: korzystając z języka Groovy oraz z silnika reguł biznesowych JBoss. Po weryfikacji wyników symulacji, przeanalizowano cechy nefunkcjonalne obydwu implementacji. Uzasadniono wybór podejścia dla dalszych badań nad rozszerzeniami PDNZC.

Rozdział 30. Porównanie dwóch metod symulacji zadań współbieżnych. W rozdziale omówiono i zanalizowano dwie metody realizacji symulacji zadań współbieżnych. Każde zadanie uczestniczące jest opisane przez układ równań różniczkowych oraz wyposażone w prawo sterowania, które umożliwia dążenie do celu z omijaniem zauważonych przeszkód. Każdy obiekt jest wykonywany cyklicznie z

wybrany okres próbki. Metoda pierwsza traktuje każde zadanie jako obiekt klasy Timer z pakietu Swing. W tej metodzie wszystkie zadania wykonywane są przez ten sam wątek, więc zastosowanie procesora wielordzeniowego nie przyspieszy działania narzędzia symulacyjnego. Metoda druga wykorzystuje specjalizowaną klasę z pakietu java.util.concurrent, którą można wyposażyć w zbiornik z określoną liczbą wątków i która przydziela wątkom określone zadania. W tej metodzie wirtualna maszyna języka Java (JVM) może przydzielić wykonanie wątków różnym rdzeniom podnosząc sprawność narzędzia symulacyjnego.

Rozdział 31. Pączkowanie – metoda rozwoju interoperacyjnych usług dla środowisk rozproszonych.

Przedstawiono 2 współczesne metody wytwarzania oprogramowania: iteracyjno-przyrostową oraz techniki zwinne, ich zalety i wady w kontekście budowy interoperacyjnych platform i środowisk rozproszonych. Zaprezentowano metodę rozwoju oprogramowania przez pączkowanie, jej założenia, zalety i wady. Przedstawiono technologie, na bazie których działa metodologia wytwarzania oprogramowania przez pączkowanie: Software Product Line, Enterprise Application Integration oraz Open-Source. Zaprezentowano ich mocne strony oraz przydatność dla omawianej metodologii. Przedstawiono proceduralnie metodę pączkowania oraz omówiono umiejscowienie kroków w przestrzeni wielowymiarowej. Zaprezentowano przykład wykorzystania metodologii podczas tworzenia interoperacyjnej platformy usług Politechniki Gdańskiej. Podano parametry związane z takim rozwojem. Porównano metodologię iteracyjno-przyrostową z metodologią pączkowania. Omówiono rezultaty wdrożenia metodologii na Politechnice Gdańskiej.

Rozdział 32. Integracja procesu konstruowania i doświadczalnej weryfikacji konstrukcji z wykorzystaniem sieci neuronowej.

Zaproponowane rozwiązanie w obszarze technologii informatycznych CAE jest na etapie eksperymentalnego wdrażania w ograniczonym zakresie. W pierwszej fazie korzystając z zebranego bogatego materiału doświadczalnego zbudowano system ekspertowy służący do predykcji stanów obciążeń zawieszenia zespołu napędowego śmigłowca. Pozwoliło to na wyeliminowanie w drodze symulacji tych obszarów parametrów lotu i obciążeń śmigłowca, które mogą być potencjalnie niebezpieczne dla istniejącej konstrukcji. System ekspertowy w zastosowaniu do modernizacji konstrukcji pozwolił także na określenie przewidywanych obciążeń konstrukcji dla pożądanych nowych obszarów parametrów eksploatacyjnych. Skraca to znacznie proces konstruowania, który poprzednio polegał na interaktywnym testowaniu kolejnych wersji konstrukcyjnych na stanowiskach badawczych lub w locie. Dotyczy to zwłaszcza obszarów, w których związki funkcjonalne między parametrami lotu a obciążeniami odpowiedzialnych elementów śmigłowca są silnie nieliniowe.

Rozdział 33. Standardy informacji medycznych jako forma integracji danych w szpitalnych systemach informatycznych.

Rozdział ten opisuje rozwój standardów w zakresie gromadzenia i przetwarzania informacji medycznych. Opisano dwa podstawowe standardy - HL7 i DICOM wraz z przedstawieniem wad i zalet związanych z ich zastosowaniem. Omówiono również Szpitalny System Informatyczny i jego składniki.

Rozdział 34. Badanie technologii informatycznych u dostawcy usług internetowych.

W artykule przedstawiono raport z badania technologii informatycznych dotyczącego analizy narzędzia, które firma wykorzystuje do zarządzania zawartością portalu (CMS). Badana firma uznała wykorzystywane narzędzia za niewystarczające w obliczu rozwoju potrzeb. Wobec tego postanowiono zweryfikować konieczność wymiany systemu na nowy oraz oszacować możliwość i opłacalność jego modyfikacji. W tym celu przeanalizowano funkcjonalności systemu CMS w podziale na logicznie wyodrębnione kategorie modułów oceniając każdą z nich. Jednocześnie dokonano audytu całej firmy w wykorzystaniem metodologii CobiT. Umożliwiło to określenie poziomu dojrzałości, na jakim znajduje się firma. Przeanalizowano poszczególne procesy biznesowe z czterech domen, które definiuje CobiT. Uzyskana ocena miała pozwolić odpowiedzieć kierownictwu firmy, czy wdrożenie nowego systemu bądź nowych modułów do obecnego systemu jest możliwe.

Rozdział 35. Modele integracji systemów informatycznych przedsiębiorstw. W artykule wskazano na potrzebę wsparcia organizacji informatycznych narzędziami informatycznymi służącymi do zarządzania zmianami i incydentami. Przedstawiono również charakterystykę systemu informatycznego do

kompleksowego zarządzania projektami, mogącego znaleźć zastosowanie do rozwiązywania kwestii zarządzania wersjami, wydania i incydentami.

Rozdział 36. Implementacja procesów zarządzania zmianami w przedsiębiorstwie. W artykule przedstawiono sposób implementacji procesów zarządzania zmianami w przedsiębiorstwie w oparciu o IBM Rational Team Concert. Celem artykułu jest wskazanie na przydatność wspierania procesów zarządzania zmianą przy pomocy IBM Rational Team Concert.

Rozdział 37. Zmniejszanie luk pomiędzy Przemysłem i Nauką z wykorzystaniem Inżynierii Wiedzy. Techniki inżynierii wiedzy (KE) stają się coraz bardziej użytecznymi i popularnymi elementami zintegrowanych systemów hybrydowych do rozwiązywania skomplikowanych problemów praktycznych w różnych dyscyplinach. Techniki inżynierii wiedzy cechują się takimi elementami jak: uczenie się z wcześniejszej wiedzy, obsługa nieustrukturyzowanych danych, pomoc w procesie podejmowania decyzji i prognozowania. Panel dyskusyjny pomoże określić, w jakim kierunku powinny zmierzać badania. Real "SMART" - technologie i nowatorskie rozwiązania wymagające wspólnej pracy pomiędzy przemysłem a nauką, mogące przyczynić się do rozwiązania wielu zagadnień. Obrady i analizy mogą pomóc społeczeństwu, naukowcom i przedsiębiorstwom w lepszym zrozumieniu problemów, ich zmotywowaniu i zaangażowaniu w celu opracowania lepszych pomysłów realizacji nowatorskich rozwiązań. Wielu naukowców i przemysł stale bada wyżej wymienione problemy, a w szczególności w ramach panelu ekspertów z różnych dziedzin, od przemysłu i nauki.

Rozdział 38. Koncepcja projektowania Inteligentnych Systemów Zarządzania. W niniejszej pracy przedstawiono pewną koncepcję projektowania i tworzenia Inteligentnych systemów wspomagania decyzji w zarządzaniu przedsiębiorstwem produkcyjnym. W przedstawianym podejściu systemy te proponuje się tworzyć jako systemy o otwartej architekturze i budowie modularnej, oparte na integracji współczesnych technologii sztucznej inteligencji (sieci neuronowe, algorytmy genetyczne, logika rozmyta i systemy ekspertowe) z dokładnymi metodami teorii podejmowania decyzji oraz symulacyjnego modelowania.

Rozdział 39. Platforma wymiany wiadomości oparta na kolejkach – wzorzec projektowy Rozdział przedstawia wzorzec projektowy (komponenty), który ma zastosowanie jako bazowa platforma komunikacyjna dla systemów klasy enterprise. Poprzez wprowadzenie komponentów jak ConfigurationService, MessageService i ClientGateway, jesteśmy w stanie zbudować wydajny system wymiany wiadomości. System ten charakteryzuje się wysoką niezawodnością, bezpieczeństwem i skalowalnością.

Rozdział 40. Wybór częściowego śladu w debuggerach dwukierunkowych. Debugger dwukierunkowy to narzędzie programistyczne, które rejestruje historię działania badanego programu i pozwala przeglądać jego wykonanie w swobodny sposób, przy pomocy interfejsu użytkownika podobnego do zwykłych debuggerów. Zarejestrowaną historię nazywamy śladem. Pełny ślad wykonania programu może być niewygodnie wielki. Rozważamy narzędzie, które działa podobnie jak debuggery dwukierunkowe, ale zbiera tylko pewną część śladu. Narzędzie takie nazywamy debuggerem *częściowego śladu*. Proponujemy dwa podejścia do strategii wyboru częściowego śladu: oparte na pokryciu kodu oraz oparte na algorytmie analizy skupień.

Rozdział 41. Testowanie regresyjne sterowane wymaganiami. Umiejętna selekcja testów regresyjnych jest bardzo istotnym aspektem zapewnienia jakości systemów oprogramowania wpływającym jednocześnie na poziom kosztów. Niniejszy rozdział przedstawia technikę automatycznej selekcji scenariuszy testowych spośród całego zestawu testów, opartą o ślady wymagań powstałe w wyniku transformacji zgodnych z koncepcją MDA. W wyniku transformacji, model wymagań, stworzony z wykorzystaniem języka RSL, przekształcany jest w modele projektowe, a następnie w kod. Podążając wygenerowanymi śladami znajdujemy wszystkie elementy odpowiedzialne za realizację wybranego wymagania (bądź grupy wymagań). Możliwe jest również śledzenie zależności tak, aby odnaleźć wszystkie wymagania, na których implementację mogła wpłynąć wprowadzona zmiana. Prezentowana, kompleksowa metoda testów regresyjnych została zrealizowana w narzędziu ReDSeeDS i może być zastosowana do budowy systemów o wysokim stopniu interakcji użytkowników w systemem.

Rozdział 42. Półautomatyczna klasteryzacja dokumentów na podstawie zadanego tematu. Zwykle algorytmy klasteryzacji uczą się bez nadzoru, proponowana metoda pozwala użytkownikowi na narzucenie sposobu grupowania pod kątem interesującego go tematu. Opisywana metoda składa się z następujących etapów: wprowadzenie pożądanego tematu, tagowanie danych na podstawie tematu i związków semantycznych, zastosowanie optymalizującego algorytmu hierarchicznej klasteryzacji danych wykorzystującego funkcję kryterialną do określenia najlepszej metody podziału.

Rozdział 43. Zarządzanie wiedzą - podejmowanie decyzji w systemie klasy workflow. Przykład implementacji węzłów decyzyjny w systemie klasy workflow z uwypukleniem szczególnych cech tej platformy. Jako proces biznesowy posłużył uproszczony proces uzyskiwania decyzji kredytowej, a jako mechanizmy wspierania decyzji zastosowano logikę, regułową oraz wnioskowanie oparte o podobieństwo spraw. Dodatkowo zwrócono uwagę, jakie elementy powinny zostać uwzględnione, aby inżynier wiedzy mógł skutecznie generować wartość dodaną w oparciu o taką platformę.

Rozdział 44. Optymalizacja witryny internetowej Banku Zachodniego WBK S.A. na podstawie testów użyteczności. Artykuł przedstawia praktyczne zastosowanie testów użyteczności w przemysłowym studium przypadku. Testy ukierunkowane były na podniesienie jakości użytkowej strony głównej serwisu internetowego ogólnopolskiego banku. W ramach wprowadzenia dokonano przeglądu znanych metod i narzędzi przeznaczonych do oceny i poprawy użyteczności. Wybrane metody i narzędzia zostały wykorzystane w badaniach użyteczności, których opis i wyniki stanowią główną część tekstu artykułu. Badania te zostały wykonane w firmie zajmującej się utrzymaniem witryny i obejmowały analizę heurystyczną, oparte na scenariuszach testy użyteczności z udziałem kilkunastoosobowej grupy ochotników oraz śledzenie działań rzeczywistych użytkowników serwisu (tzw. clicktracking). Końcowa część artykułu przedstawia zmiany zaproponowane na podstawie wyników badań, proces ich zatwierdzania i implementacji oraz podsumowanie.

Rozdział 45. Integracja modeli ontologicznych metodyk zarządzania projektami IT na przykładzie zarządzania zmianą. W środowisku przedsiębiorstw branży IT, których podstawową formą działalności jest realizacja projektów, poszukuje się rozwiązań poprawiających efektywność i konkurencyjność polegających na wdrożeniu sprawdzonych metodyk zarządzania projektami informatycznymi obejmujących podejście klasyczne i zwinne. Bardzo często stosowane są oba typy metodyk dla równoległe prowadzonych projektów, dokonuje się także ich adaptacji i prób integracji. W rozdziale proponuje się wykorzystanie formalnych modeli ontologicznych do opisu metodyk zarządzania projektami oraz przeprowadzenie ich integracji poprzez integrację modeli ontologicznych. W rozdziale opisano podstawowe elementy dwóch ontologii modelujących klasyczną metodykę PMBOK oraz najbardziej popularną zwinną metodykę Scrum: artefakty, role, procesy i zdarzenia. Pojęcia te razem ze zbiorem specyficznych relacji są wykorzystywane by formalnie zdefiniować zarządzanie zmianą w obu metodykach.

Rozdział 46. Aktywna metoda testowania bezpieczeństwa aplikacji webowych HttpValider oraz ocena jej skuteczności. W rozdziale omówiono zagadnienia oceny bezpieczeństwa aplikacji webowych, w szczególności podejście pasywnego i aktywnego testowania bezpieczeństwa. Przedstawiono metodę aktywnego testowania HttpValider obejmującą generowanie żądań HTTP pozwalające na identyfikację 13 luk bezpieczeństwa. Dla opracowanej metody zaprojektowano i zaimplementowano narzędzie wspierające automatyczne przeprowadzenie testów dla scenariusza użycia testowanej aplikacji zarejestrowane w pliku śladu przez testera. Skuteczność metody została potwierdzona podczas walidacji wykonanej dla testowych aplikacji zawierających oraz nie zawierających poszczególne luki bezpieczeństwa. Wykonano również testy porównawcze HttpValidera z dwoma innymi narzędziami testów bezpieczeństwa dla dwóch aplikacji webowych. Przedstawiono wyniki testów potwierdzające skuteczność metody HttpValider w wykrywaniu zbioru 13 luk bezpieczeństwa.

Rozdział 47. Wspomaganie zarządzania oparte na inżynierii wiedzy. Omawia się problemy związane z zachowaniem treści oraz (w ogólności) wiedzy pomiędzy poszczególnymi etapami PLM. Analizuje się też zagadnienie wpływu tych technik na realizację skutecznego zarządzania. Odnosimy się również do problemów implementacyjnych związanych z integracją systemów informatycznych oraz aplikacji webowych, biorąc pod uwagę bezpieczeństwo i niezawodność przedmiotowej wiedzy. Pytania oraz odpowiedzi wynikają także z perspektywy przemysłowo-biznesowej, ponieważ wiele z prezentowanych technik ma w rzeczywistości zastosowanie w aplikacjach czasu rzeczywistego działających w zarządzanych środowiskach w różnego rodzaju branżach przemysłowych. Wnioski z wiedzy zawartej w obiektach inżynierskich dotyczą nie tylko strony technicznej, ale również w niedługim czasie rozwinię się dyskusja nad etyką tego zagadnienia, ponieważ wiedza i doświadczenie będzie przenoszone do inteligentnych obiektów stanowiących fundamenty paradygmatu internetowego.