

## Metoda zarządzania architekturą korporacyjną

T. GÓRSKI

e-mail: gorski@wat.edu.pl; tomasz.gorski@rightsolution.pl

Institute of Information Systems  
Cybernetics Faculty, Military University of Technology  
Kaliskiego Str. 2, 00-908 Warsaw

---

Artykuł opisuje sposób zarządzania architekturą korporacyjną w złożonej organizacji. Przedstawione zostanie podejście obejmujące metodykę wytwarzania oprogramowania, sposób opisu oraz wsparcie narzędziowe. Jako metodyka wytwarzania oprogramowania zaproponowana została Rational Unified Process (RUP). W artykule opisany zostanie sposób mapowania Rational Unified Process na siatkę Zachmana. Pokazany zostanie sposób opisu architektury korporacyjnej wykorzystujący artefakty RUP. Zaprezentowane zostanie także zastosowanie Unified Modeling Language (UML) 2.0 do opisu architektury korporacyjnej wraz z opisem zastosowanych profili języka UML. Opracowana metoda zilustrowana zostanie przykładem zarządzania architekturą korporacyjną na przykładzie usługi publicznej „Udział w zamówieniach publicznych”, opisanej we „Wrotach Polski”. W przykładzie zostanie pokazany sposób zastosowania narzędzi IBM Rational Software Architect i IBM Rational RequisitePro do zarządzania architekturą korporacyjną.

---

**Keywords:** architektura korporacyjna, proces projektowania systemu informatycznego

### 1. Architektura korporacyjna

Podstawowym celem tworzenia architektury korporacyjnej jest wsparcie celów i procesów biznesowych zachodzących w organizacji. Wynikają z tego potrzeby informacyjne, które należy zaspokoić przez budowę systemów informatycznych.

Przy tworzeniu architektury korporacyjnej należy kierować się następującymi zasadami:

- wszystkie dane i artefakty generowane przez organizację są zarządzane i utrzymywane przez tą organizację,
- możliwa jest techniczna realizacja architektury i może być ona zaimplementowana w ramach rozsądnych nakładów finansowych i w akceptowalnym czasie,
- architektura umożliwia śledzenie od wymagań biznesowych do technicznej implementacji,
- architektura dokumentuje się sama.

Przy opracowywaniu architektury korporacyjnej istotne jest sformułowanie odpowiedzi na następujące pytania: Dlaczego?, Co? i Jak? Wymienione pytania określają trzy podstawowe zakresy architektury korporacyjnej:

- biznesowy,
- logiczny,
- technologiczny.

Architektura korporacyjna ma bezpośrednie przełożenie na metodykę wytwarzania systemów

informatycznych zastosowaną w organizacji, która powinna obejmować w/w zakresy architektury korporacyjnej.

Opracowane są różne metodyki zarządzania architekturą korporacyjną np.: metodyka TRM (Technology Roadmapping Methodology), TOGAF (The Open Group Architecture Framework). Istotnym wydaje się zaproponowanie kompletnego rozwiązania do zarządzania architekturą korporacyjną. W ogólności w metodzie zarządzania złożonym systemem a w szczególności architekturą korporacyjną istotnym jest zapewnienie trzech elementów:

- procesu wytwórczego,
- metody opisu,
- wsparcia narzędziowego.

Przedstawiana metoda obejmuje w/w elementy. Do weryfikacji kompletności zastosowanej metody opisu architektury korporacyjnej zastosowano Siatkę Zachmana. W 1987 roku John Zachman opublikował w IBM SYSTEMS JOURNAL artykuł "A Framework for Information Systems Architecture"[8], w którym przedstawił model opisujący artefakty procesu wytwarzania oprogramowania. Zachman wniosł, że system informatyczny powinien wspierać cele biznesowe korporacji. Zidentyfikował on dwa różne typy reprezentacji, które użyte razem jasno opisują naturę i cel tych artefaktów w ramach organizacji.

Pierwsza reprezentacja stanowi perspektywę ludzką. Zachman zdefiniował sześć widoków architektonicznych potrzebnych do dokładnego opisu wszystkich etapów wytwarzania artefaktów systemu informatycznego w ramach korporacji. W ramach każdego z tych widoków poziom szczegółowości może być dostosowywany odpowiednio do potrzeb. Widoki:

- Scope (Ballpark) View - opisuje cel biznesowy i strategię, co definiuje pole gry (ballpark) dla innych widoków,
- Owner's View - opisuje organizację, w której system informatyczny musi funkcjonować (określa części korporacji do automatyzacji),
- Designer's View - pokazuje jak system spełni potrzeby informacyjne organizacji (aspekty specyficzne dla rozwiązania lub ograniczenia specyficzne dla wytwarzania),
- Builder's View – przedstawia jak system zostanie zaimplementowany (dostarcza określonych rozwiązań i technologii rozwiązujących ograniczenia wytwórcze),
- Out-of-Context View - szczegóły implementacyjne określonych elementów systemu,
- Operational View – przedstawia funkcjonujący system w jego środowisku operacyjnym.

	Data (What)	Function (How)	Network (Where)	People (Who)	Time (When)	Motivation (Why)
Scope View	List of things important to the enterprise	List of processes the enterprise performs	List of locations where the enterprise operates	List of organizational units	List of business events/cycles	List of business objectives
Owner's View	Entity relationship diagram	Business process model (physical data flow diagram)	Logistics network (nodes and links)	Organizational chart, with roles, skill sets, security issues	Business master schedule	Business rules
Designer's View	Data model (converged entities, fully normalized)	Essential data flow diagram, application architecture	Distributed system architecture	Human interface architecture (roles, data, access)	Dependency diagram, entity life history (process structure)	Business rule model
Builder's View	Data architecture (tables and columns), map to legacy data	System design: structure chart, pseudo-code	System architecture (hardware, software types)	User interface (how the system will behave), security design	"Control flow" diagram (control structure)	Business rule design
Detailed View	Data design (denormalized), physical storage design	Detailed program design	Network architecture	Screens, security architecture (who can see what?)	Timing definitions	Rule specification in program logic
Operational View	Converted data	Executable programs	Communications facilities	Trained people	Business events	Enforced rules

Rys. 1. Siatka Zachmana [1]

Druga reprezentacja przedstawia sześć punktów skupienia. Punkty te używane są do klasyfikacji odpowiednich własności i podkreślenia różnych aspektów systemu. Każdy punkt skupienia dostarcza innego, skupionego na produkcji widoku na system. Punkty skupienia:

- What – zawartość systemu (dane w przypadku systemu informatycznego),

- How – sposób użycia i funkcjonowanie systemu (procesy i przepływy sterowania),
- Where – elementy przestrzenne oraz ich relacje, włączając rozproszenie systemu,
- Who – ludzie i jednostki organizacyjne oraz ich interakcje z systemem informatycznym,
- When – sekwencyjność i zależności czasowe do procesów i przepływów (w widoku tym opisywana jest także współbieżność),
- Why – przyczyny tworzenia systemu i zasady ograniczania go.

Przez łączenie tych dwóch wymiarów Siatka Zachmana stanowi narzędzie do analizy artefaktów procesu wytwórczego. Unikalność użytych reprezentacji umożliwia dokładne opisanie natury i celu każdej komórki. Używając tego opisu organizacja może ocenić pokrycie jej procesu wytwarzania oprogramowania i wpływ nowych narzędzi i technik w kontekście strategii zarówno biznesowych jak i informatycznych.

## 2. Rational Unified Process

W ramach metody zarządzania architekturą korporacyjną zaproponowana została metodyka wytwarzania oprogramowania Rational Unified Process (RUP).

RUP implementuje sześć najlepszych praktyk inżynierii oprogramowania: wytwarzaj iteracyjnie, zarządzaj wymaganiami, używaj architektury komponentowej, modeluj wizualnie (UML), stale weryfikuj jakość, zarządzaj zmianami.

W ramach RUP występuje dziewięć dyscyplin począwszy od modelowania biznesowego a skończywszy na utrzymaniu środowiska uruchomieniowego (produkcyjnego). W ramach każdej z dyscyplin określone są czynności i role odpowiedzialne za wykonanie tych czynności, artefakty (produkty) potrzebne do realizacji czynności a także artefakty będące wynikiem realizacji czynności. W ramach dyscyplin określony jest także sposób realizacji czynności i szablony artefaktów.

Cały proces wytwarzania podzielony jest na fazy a w ramach faz występują iteracje. Metodyka ta szczególnie nacisk kładzie na podejście iteracyjne do wytwarzania oprogramowania. W ramach każdej iteracji wykonywane są czynności z każdej z dyscyplin występujących w RUP. W zależności od fazy projektu nacisk kładziony jest na inne dyscypliny. Szczegółowo RUP został opisany w [6], [10].

### 3. Zunifikowany język modelowania UML 2.0

Przy tworzeniu architektury korporacyjnej istotny jest dobór języka jej opisu. Rozwój i powszechność stosowania języka UML przemawia za jego wyborem. Unified Modeling Language jest językiem do: wizualizacji, specyfikacji, dokumentowania i budowania systemu informatycznego. Dzięki zastosowaniu UML można w sposób graficzny przedstawić różne spojrzenia na system: strukturalne, dynamiczne. Tworząc modele UML dokonujemy specyfikacji wymagań na system, szczególnie stosując model przypadków użycia i diagramy aktywności. Modelując system za pomocą diagramów UML jednocześnie dokumentujemy decyzje projektowe w jednoznaczny i zrozumiały dla całego zespołu sposób. Natomiast stosując narzędzia wspierające automatyzację modelowania systemu i generowania kodu z opracowywanych modeli UML mamy możliwość budowy systemu w zdeterminowany i zautomatyzowany sposób. W ramach UML 2.0 występują następujące typy diagramów zachowania (przypadków użycia, aktywności, sekwencji, przeglądu interakcji, komunikacji, stanów, czasowy) oraz statyczne (klas, obiektów, pakietów, komponentów, struktury połączeń, wdrożeniowy). Wersja UML 2.0 została opisana w precyzyjny i zwięzły sposób w [3].

Istotą rozbudowy języka UML jest zachowanie jego prostoty. W związku z tym w specyfikacji języka UML nie pojawiło się wiele nowych elementów. Natomiast wraz z pojawieniem się specyfikacji 2.0 UML następuje szybki rozwój i powstają nowe elementy pozwalające na rozszerzenie możliwości języka UML nazywane profilami. Profil UML stanowi zbiór stereotypów, ograniczeń, typów wyliczanych i wartości metkowanych (ang. tagged values), które zebrane razem, rozszerzają UML dla określonej platformy albo dziedziny modelowania. Dla przykładu modelując system projektowany przy wykorzystaniu J2EE potrzebne są elementy modelu typu komponent EJB. Należy, więc utworzyć profil zawierający wszystkie elementy potrzebne do zamodelowania projektowanego rozwiązania.

Istotą stosowania profili do modeli jest to, że dodajemy możliwość używania określonych znaczeń do istniejących elementów modelu. Do pakietu (modelu) można stosować wiele profili. Przy usuwaniu profilu z pakietu nie są usuwane elementy modelowe, do których profil został

zastosowany, lecz jedynie rozszerzenia z profilu. Usuwając profil, odbieramy elementom modelu znaczenia, które zostały im nadane przy wykorzystaniu usuniętego profilu, np. zastosowano profil *Business Modeling* i stereotyp <<entity>> do klasy *Rezerwacja*. Po usunięciu profilu klasa *Rezerwacja* pozostaje, lecz już bez stereotypu <<entity>>.

Narzędzie IBM Rational Software architekt (RSA) umożliwia tworzenie własnych profili i stosowanie ich do projektów i modeli już istniejących. Własny profil budowany jest, jako projekt RSA. Można utworzyć własne stereotypy, określić relacje z elementami specyfikacji UML 2.0, do których stereotypy możemy stosować oraz określić typy wyliczane. Przygotowane profile można stosować do projektów w RSA i używać w nich uprzednio utworzonych elementów profilu.

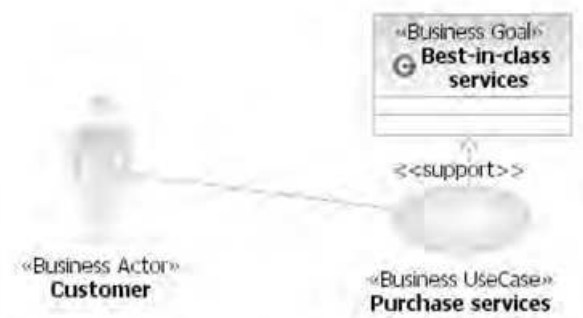
Przy opisie architektury korporacyjnej na poziomie modelowania biznesowego pomocnym jest zastosowanie profilu **IBM Rational UML Profile for Business Modeling** [5].

W ramach tego profilu występują dwa modele: **Business Use-Case Model** (model wymaganych procesów biznesowych) oraz **Business Object Model** (model obiektów opisujący realizację biznesowych przypadków użycia). Poniższy rysunek prezentuje przykładowy diagram celów pokazujący dekompozycję głównego celu biznesowego na szczegółowe cele biznesowe (Rys.2).



Rys. 2. Diagram celów

Cele biznesowe określone w organizacji wspierane są przez biznesowe przypadki użycia modelujące procesy w niej zachodzące (Rys. 3).



Rys. 3. Wspieranie celów biznesowych przez biznesowe przypadki użycia

Wykonanie projektu modelowania biznesowego jest sposobem na uzyskanie wymagań funkcjonalnych na potrzebne systemy informatyczne. Jeden proces modelowania biznesowego dostarcza pełen zestaw modeli biznesowych i może stanowić wejście dla kilku połączonych projektów wytwarzania oprogramowania. Występujące w ramach profilu modelowania biznesowego stereotypy przedstawiono na rysunku Rys. 4.

Metaklasa UML	Stereotypy
Actor	Business Actor
Class	Business Entity, Business Goal, Business Worker, Case Worker
Collaboration	Business Use Case Realization
Constraint	Business Rule
Dependency	owner, supports
Model	Business Use Case Model, Business Analysis Model
Package	Business System
Use Case	Business Use Case

Rys. 4. Stereotypy profile modelowania biznesowego

Po etapie modelowania biznesowego i wyspecyfikowaniu wymagań (realizowane w narzędziu IBM Rational RequisitePro) następuje etap analizy i projektowania systemu informatycznego. Na tym etapie istotna jest możliwość modelowania konstrukcji specyficznych dla przyjętej architektury logicznej i zastosowanych technologii.

Jednym z podejść architektonicznych, które jest ostatnio najszybciej rozwijane jest Architektura Zorientowana na Usługi (ang. Service Oriented Architecture (SOA)). Firma IBM opracowała i dostępny jest do stosowania w jej narzędziach profil umożliwiający modelowanie rozwiązań zgodnych z tą architekturą. Stosowanie **UML 2.0 Profile for Software Services** [4] umożliwia modelowanie rozwiązań ukierunkowanych na usługi. Profil ten zaimplementowany jest w IBM Rational Software Architect.

Poniższa tabela przedstawia stereotypy profilu UML 2.0 Profile for Software Services (Rys. 5).

Metaklasa UML 2.0	Stereotypy
Class	Message, Service Partition, Service Provider
Classifier	Service Consumer
Collaboration	Service Collaboration
Connector	Service Channel
Interface	Service Specification
Port	Service, Service Gateway
Property	Message Attachment

Rys. 5. Stereotypy profilu usług

Celem profilu jest dostarczenie wspólnego języka dla opisu usług. Opis ten zawiera potrzebne aktywności w całym cyklu wytwarzania oprogramowania oraz dostarcza widoków dla różnych udziałowców. Profil dostarcza możliwości szczegółowego planowania usług dla architektów we wczesnych fazach wytwarzania oprogramowania przez użycie logicznych podziałów do opisu usług w ramach całej korporacji.

#### 4. Mapowanie artefaktów procesu projektowania systemów informatycznych na siatkę Zachmana

Siatka Zachmana została zastosowana, jako narzędzie do oceny produktów opisujących różne aspekty opisu architektury korporacyjnej organizacji. W proponowanej metodzie mapowane zostały następujące typy artefaktów procesu projektowania systemów informatycznych: wymagania, dokumenty, modele, elementy modeli.

Na rysunku Rys. 6 przedstawiono mapowanie artefaktów do komórek Siatki Zachmana dla punktów skupienia: Motywacja, Funkcja oraz Sieć w ramach trzech pierwszych widoków architektonicznych, mających największe znaczenie architektoniczne.

	Motywacja	Funkcja	Sieć
Zakres (Scope)	Cel biznesowy, Wizja biznesowa	Biznesowy przypadek użycia	Lokalizacje biznesowe
Posiadacz (Owner)	Dokument Wizja i wymagania udziałowców	Specyfikacja biznesowego przypadku użycia	Pracownicy biznesowi i biznesowe diagramy komunikacji
Projektant (Designer)	Specyfikacja przypadku użycia i Wymagania dodatkowe	Model przypadków użycia	Systemy i podsystemy (diagram wdrożeniowy i komponentów)

Rys. 6. Mapowanie artefaktów procesu projektowania systemów informatycznych na Siatkę Zachmana

Na rysunku Rys. 7 przedstawiono mapowanie artefaktów do komórek Siatki Zachmana dla punktów skupienia: Ludzie, Czas, Dane w ramach trzech pierwszych widoków architektonicznych, mających największe znaczenie architektoniczne.

	Ludzie	Czas	Dane
<b>Zakres (Scope)</b>	Jednostki organizacyjne	Zdarzenia biznesowe	Słownik biznesowy
<b>Posiadacz (Owner)</b>	Realizacja biznesowego przypadku użycia (diagramy aktywności)	Realizacja biznesowego przypadku użycia (diagramy sekwencji)	Biznesowy model obiektów
<b>Projektant (Designer)</b>	Aktorzy i interfejs użytkownika	Model analityczny (diagramy interakcji)	Analityczny model klas

Rys. 7. Mapowanie artefaktów procesu projektowania systemów informatycznych na Siatkę Zachmana, cd.

W widoku **Zakres (Scope)** występują elementy dwóch modeli biznesowych: Business UseCase Model oraz Business Object Model oraz Wymagania. W widoku **Posiadacz (Owner)** występuje model biznesowy Business Object Model oraz Wymagania. W widoku **Projektant (Designer)** występują modele analityczne UseCase Model oraz Analysis Model.

W widoku **Zakres** wykorzystywane są diagramy przypadków użycia do przedstawienia Biznesowego modelu przypadków użycia i diagram klas do przedstawienia celów biznesowych. Dokumenty występujące w widoku **Zakres**:

- Słownik biznesowy,
- Wizja biznesowa.

W widoku **Posiadacz** wykorzystywane są diagramy przypadków użycia, aktywności, sekwencji i klas do przedstawienia realizacji biznesowych przypadków użycia oraz diagram komunikacji do przedstawienia współpracujących pracowników biznesowych. Dokumenty występujące w widoku **Posiadacz**:

- Specyfikacja biznesowego przypadku użycia,
- Reguły biznesowe,
- Wizja,
- Wymagania udziałowców.

W widoku **Projektant** wykorzystywane są diagramy przypadków użycia, aktywności, sekwencji i klas do przedstawienia realizacji przypadków użycia oraz diagram komponentów i wdrożeniowy do przedstawienia struktury systemu i środowiska uruchomieniowego (produkcyjnego) systemu. Dokumenty występujące w widoku **Projektant**:

- Wizja,
- Specyfikacja przypadku użycia,
- Specyfikacja dodatkowa,
- Słownik.

## 5. Przykład zastosowania metody

Zastosowanie metody zostanie pokazane na przykładzie usługi publicznej – udział w zamówieniu publicznym. Przykład ten został opracowany na potrzeby seminarium przeprowadzonego przez autora dla Ministerstwa Finansów RP.

Zarządzanie architekturą korporacyjną realizowane jest przy wykorzystaniu następujących narzędzi:

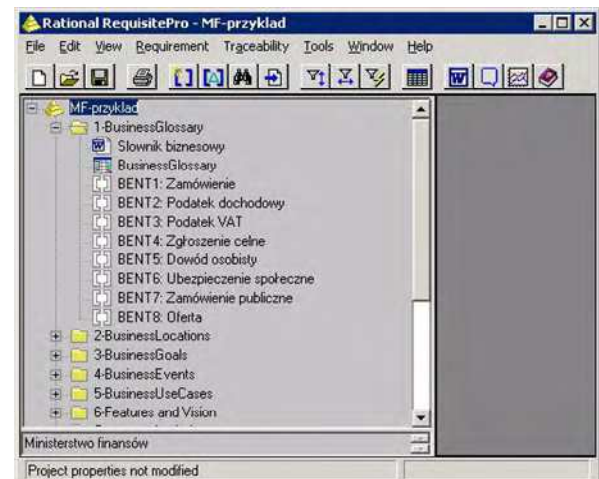
- IBM Rational RequisitePro – zarządzanie wymaganiami i dokumentami,
- IBM Rational Software Architect – zarządzanie modelami, diagramami i kodem aplikacji.

W zarządzaniu architekturą korporacyjną w widoku Zakres istotne jest opracowanie wymagań biznesowych i modelu biznesowych przypadków użycia. Narzędzie IBM Rational RequisitePro pozwala na wprowadzenie:

- własnych typów wymagań,
- własnych typów i szablonów dokumentów,
- określenie typów wymagań przechowywanych w określonych typach dokumentów,
- określenie struktury projektu,
- określenie śledzenie między typami wymagań.

W ramach rozpatrywanego przykładu określono następujące typy wymagań: pojęcie biznesowe, lokalizacja biznesowa, cel biznesowy, lokalizacja biznesowa, biznesowy przypadek użycia.

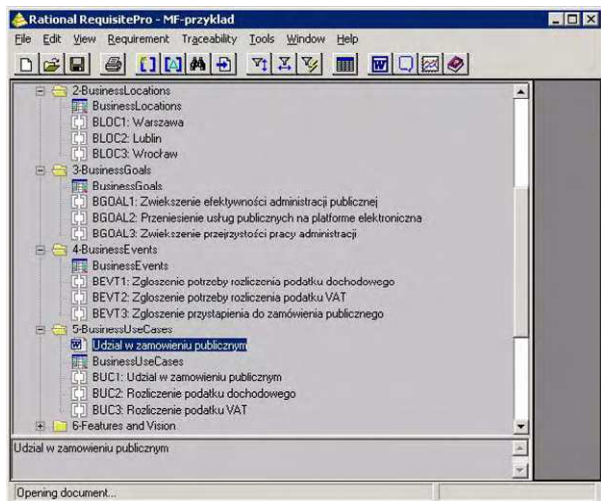
Wprowadzono także szablon dokumentu Słownik biznesowy (Rys. 8). Na jego podstawie utworzono dokument Słownik biznesowy i wprowadzono do niego pojęcia biznesowe.



Rys. 8. Widok dostosowanego projektu w RequisitePro

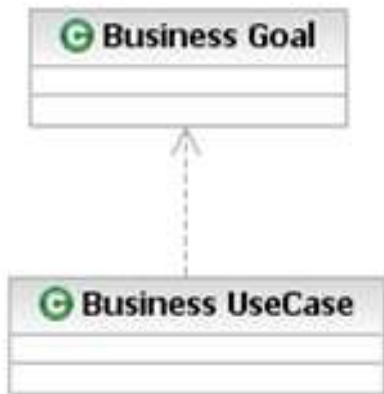


W ramach specyfikacji wymagań biznesowych wprowadzono lokalizacje biznesowe, cele biznesowe, zdarzenia biznesowe oraz biznesowe przypadki użycia. Biznesowy przypadek użycia „Udział w zamówieniu publicznym” został wyspecyfikowany w dokumencie Specyfikacja biznesowego przypadku użycia (Rys. 9).



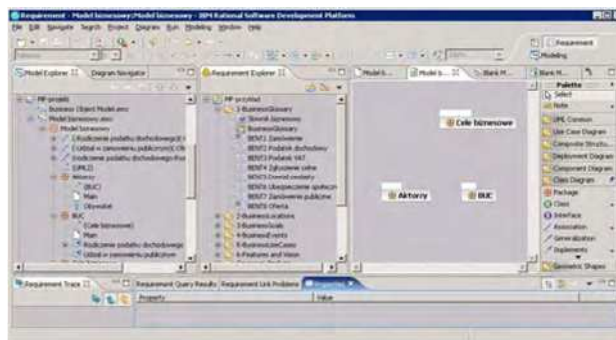
Rys. 9. Katalogi i wprowadzone wymagania określonych typów

Poniżej przedstawiono wybrane typy wymagań i śledzenie określone między nimi (Rys. 10).



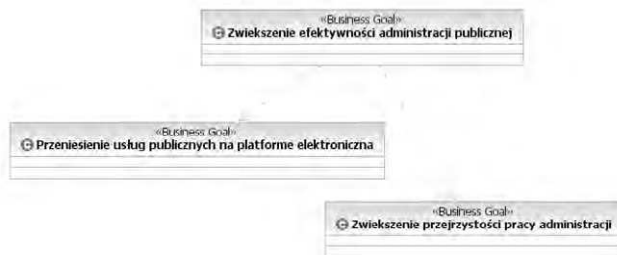
Rys. 10. Wybrane typy wymagań i śledzenie między nimi

Tak przygotowany projekt wymagań można zintegrować z pozostałą częścią architektury korporacyjnej zarządzaną w IBM Rational Software Architect (Rys. 11). Projekt wymagań widoczny jest w Requirement Explorer.



Rys. 11. IBM Rational Software Architect z widokiem na projekt wymagań

W RSA istnieje możliwość połączenia poszczególnych elementów modelu biznesowego (Model biznesowy.emx) z projektem wymagań (MF-przykład.rqs). Określenie śledzenie dotyczy zarówno wymagań funkcjonalnych jak i dodatkowych a także takich elementów modelu jak klasy czy pojęcia biznesowe w projekcie wymagań. W ramach opracowywania modelu biznesowego tworzony jest diagram celów (Rys. 12).



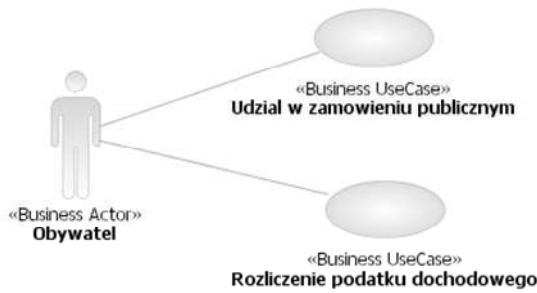
Rys. 12. Diagram celów biznesowych

Określone cele biznesowe wspierane są przez biznesowe przypadki użycia (Rys. 13).



Rys. 13. Wsparcie celów biznesowych przez biznesowe przypadki użycia.

Dodatkowo tworzony jest model biznesowych przypadków użycia prezentujący, aktorów biznesowych inicjujących, korzystających z procesów biznesowych (Rys. 14).



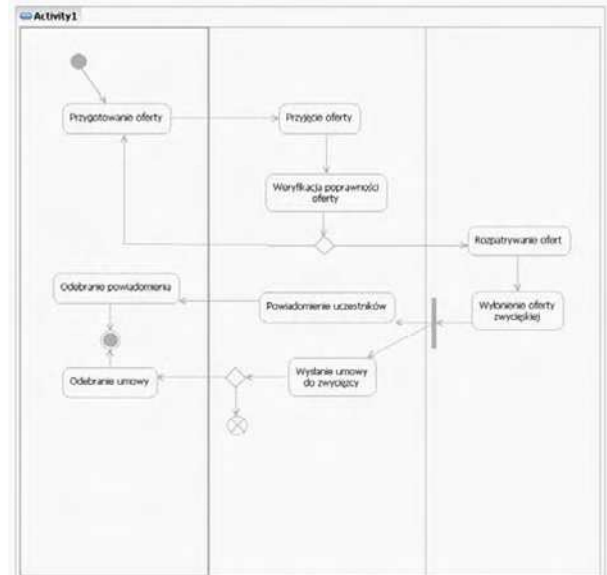
Rys. 14. Model biznesowych przypadków użycia

W zarządzaniu architekturą korporacyjną w widoku Posiadacz istotne jest opracowanie biznesowego modelu obiektowego. W biznesowym modelu obiektowym (Business Object Model.emx) tworzone są realizacje biznesowych przypadków użycia opisujące przebieg procesu biznesowego (Rys. 15).



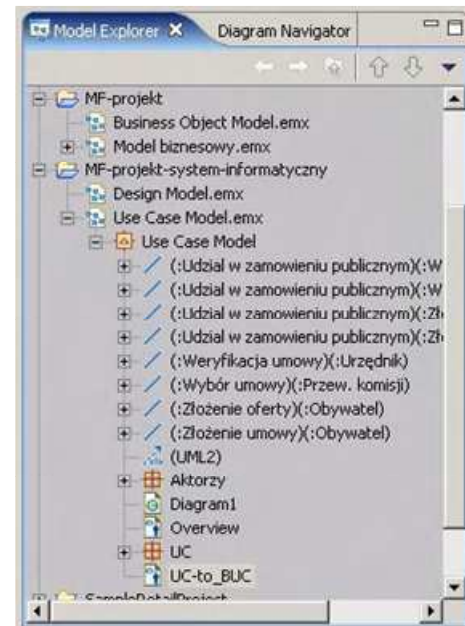
Rys. 15. Tworzenie relacji realizacji między biznesowymi przypadkami użycia a realizacjami tych przypadków

Na realizacje biznesowych przypadków użycia składają się diagramy aktywności, sekwencji i klas. Diagram klas przedstawia elementy biznesowe odpowiadające pojęciom biznesowym określonym w Słowniku biznesowym i wyspecyfikowanym w projekcie wymagań. Na rysunku Rys. 16 przedstawiono diagram aktywności prezentujący przebieg obsługi biznesowego przypadku użycia „Udział w zamówieniu publicznym”.



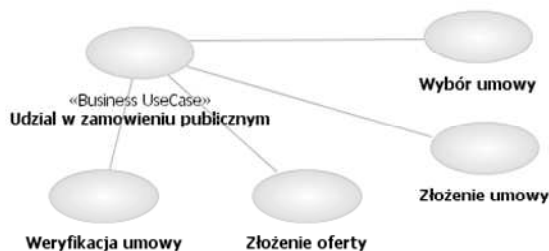
Rys. 16. Diagram aktywności prezentujący przebieg procesu udziału w zamówieniu publicznym.

W ramach zarządzania architekturą korporacyjną w widoku Projektant do pojedynczego projektu biznesowego w IBM Rational Software Architect może być podłączonych wiele projektów systemów informatycznych (Rys. 17).



Rys. 17. Możliwość zarządzania wieloma powiązаныmi projektami

Połączenie modeli następuje przez wskazanie przypadków użycia z modelu przypadków użycia projektu systemu informatycznego (MF-projekt-system-informatyczny.emx) realizujących biznesowy przypadek użycia z modelu biznesowego (Model biznesowy.emx) (Rys. 18).



Rys. 18. Połączenie projektu systemu informatycznego z projektem biznesowym

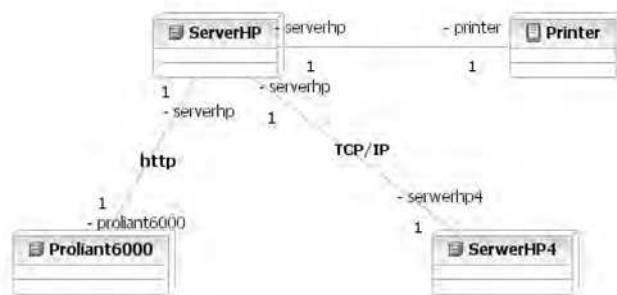
Następnie wykonywane są czynności w ramach dyscypliny analiza i projektowanie obiektowe dla poszczególnych przypadków użycia. Na tym etapie można zastosować elementy profilu UML 2.0 Profile for Software Services. Projektując rozwiązania zorientowane na usługi możemy zbudować zestaw usług i określić systemy informatyczne w korporacji, w których te usługi są wykorzystywane.

Poniżej przedstawiono diagram struktury połączeń prezentujący usługi świadczone w określonym systemie informatycznym organizacji (Rys. 19).



Rys. 19. Diagram struktury połączeń prezentujący usługi w ramach kolaboracji Obsługa zamówień publicznych.

Łatwiejsze staje się zarządzanie architekturą systemów informatycznych gdyż w jednym narzędziu możemy projektować wszystkie systemy informatyczne w organizacji. Kolejną zaletą jest możliwość określenia konfiguracji sprzętowej systemu bądź systemów informatycznych funkcjonujących w organizacji. W tym celu stosuje się diagram wdrożeniowy (Rys. 20).



Rys. 20. Diagram wdrożeniowy systemu informatycznego

## 6. Podsumowanie

Przedstawiona metoda pomaga zarządzać systemami informatycznymi w organizacji, umożliwiając utrzymanie powiązań między celami i procesami biznesowymi a wspierającymi je systemami informatycznymi. Metoda jest łatwo adaptowalna i można stosować ją do zarządzania architekturą korporacyjną, w szczególności w sektorze publicznym. Istotnym jest, że metodę można dostosować do potrzeb i tą dostosowaną postać opublikować w postaci stron html, zapewniając jej dostępność w ramach organizacji.

Zastosowane narzędzie IBM Rational RequisitePro pozwala na dostosowywanie typów wymagań, ich struktury oraz typów dokumentów. Natomiast IBM Rational Software Architect pozwala na zarządzanie wieloma projektami tak biznesowymi jak informatycznymi. Elastyczność narzędzia podnosi możliwość stosowania profili języka UML a także tworzenia własnych profili w ramach IBM Rational Software Architect. Istnieje też możliwość integracji z wymaganiami z RequisitePro na poziomie wymagań funkcjonalnych jak i dodatkowych.

Dla przykładu decydując się na architekturę Service Oriented Architecture dostępne jest wsparcie IBM Rational na poziomie:

- procesu - Rational Unified Process for SOA,
- języka modelowania - UML 2.0 Profile for Software Services,
- narzędzi - IBM Rational Software Architect, IBM Rational RequisitePro.

Zdaniem autora dzięki zastosowaniu opisanej metody oraz dostępnemu wsparciu IBM Rational na poziomie procesu, języka modelowania i narzędzi można będzie w sposób efektywny zarządzać architekturą korporacyjną organizacji i skupić się głównie na projektowaniu systemów informatycznych.



## 7. Literatura

- [1] Ambler S.W., *Enterprise Unified Process. Extending Rational Unified Process*, Pearson Education Inc., Crawfordsville, USA, 2005,
- [2] DJ the Villiers, *Using the Zachman Framework to Assess Rational Unified Process*, 2001, ([www.ibm.com](http://www.ibm.com))
- [3] Fowler M., *UML Distilled 3<sup>rd</sup> Edition*, Addison-Wesley, Kanada, 2005,
- [4] Johnston S., *UML 2.0 Profile for Software Services*, 2005, ([www.ibm.com](http://www.ibm.com)),
- [5] Johnston S. *Rational UML Profile for Business Modeling*, 2004, ([www.ibm.com](http://www.ibm.com)),
- [6] Kruchten P., *What is the Rational Unified Process?*, 2003, ([www.ibm.com](http://www.ibm.com)),
- [7] Rozanski N., *Software Systems Architecture*, Pearson Education Inc., Crawfordsville, USA, 2005,
- [8] Zachman J.A., *A framework for information systems architecture*, 1987.
- [9] Górski T., *Metoda zarządzania architekturą korporacyjną*, Software Developer's Journal Extra! nr 18, strony 58-64, 2005.
- [10] Kruchten P., *The Rational Unified Process, An Introduction*, Addison-Wesley, USA, 1999.