

Realizacja zadań na egzaminie praktyczny z inżynierii oprogramowania

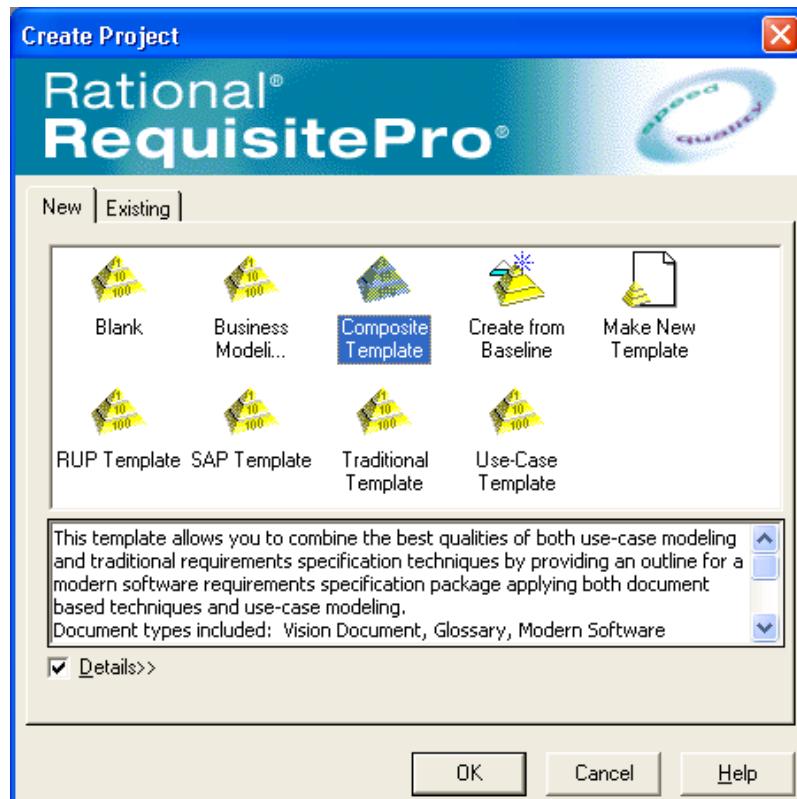
opracowała Monika Michrowska

Spis treści

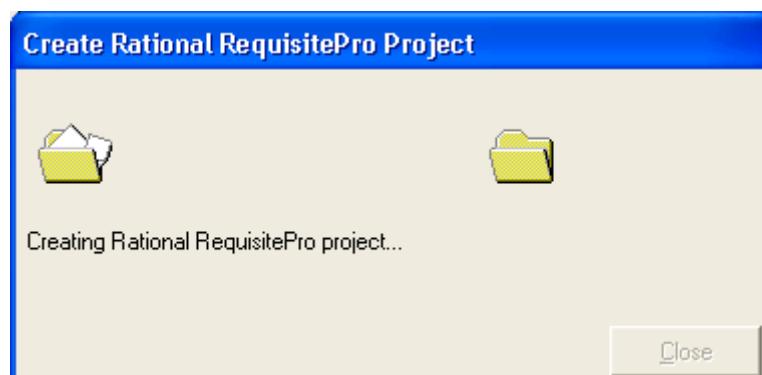
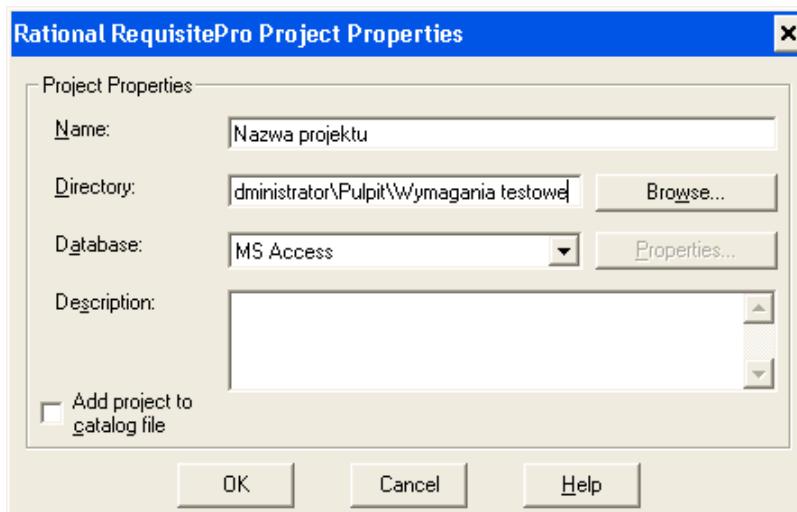
1.	Utwórz repozytorium wymagań werbalnych (na podstawie wzorca „Composite”); repozytoria modeli opisowych i wizualnych.....	2
2.	Tworzenie wymagań w Rational Requisite PRO	3
3.	widok żądań udziałowców	4
4.	Tworzenie pojęć w notacji BNF	6
5.	widok wymagań	7
6.	widok śledzenia wymagań	9
7.	Dokonaj strukturalizacji wymagań typu UC	11
8.	Dokonaj eksportu tych widoków do pliku RaportWymagań.doc	14
9.	Repozytorium wymagań wizualnych (model wymagań – na podstawie wzorca „pusty”)	15
10.	Tworzenie Worspace.....	15
11.	Wybór perspektywy	17
12.	Tworzenie połączenia z repozytorium.....	19
13.	Zarządzanie obszarami projektu.	21
14.	Tworzenie nowego workspace w repozytorium.....	22
15.	Pending Changes	28
16.	Odbieranie zmian przychodzących.....	30
17.	Utwórz model przypadków użycia poprzez asocjację repozytoriów	31
18.	Utwórz model przypadków użycia	37
19.	specyfikacja aktorów.....	37
20.	Model powiązań aktorów.....	38
21.	model analizy systemu (jako element repozytorium RSA)	39
22.	Utwórz model Klas	40
23.	Stereotypy	43
24.	Realizacja przypadków użycia- do diagramów interakcji i VOPC.....	44
25.	Tworzenie diagramów VOPC i sekwencji.....	46
26.	Specyfikacja klas analitycznych	48
27.	Dodawanie klasy granicznej	48
28.	Dodawanie klasy sterującej	49
29.	Dodawanie klasy danych	50
30.	Utworzenie diagramu sekwencji; fragment w modelu sekwencji	51
31.	Przekształć model sekwencji w model interakcji.....	59
32.	Tworzenie diagramu VOPC.....	61
33.	Diagram komponentów	63
34.	Komponent bazy danych	70
35.	Mapowanie klas i wygenerowanie skryptu SQL	79
36.	Dodatki	91
	Tworzenie Work Item	91
	Utwórz model aktywności dla wybranej usługi.....	91

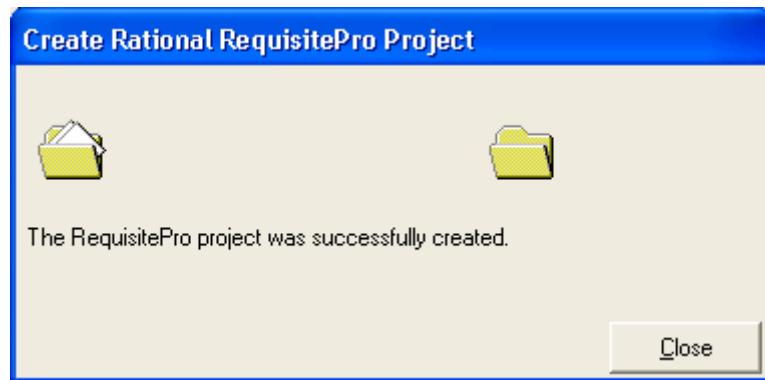
1. Utwórz repozytorium wymagań werbalnych (na podstawie wzorca „Composite”); repozytoria modeli opisowych i wizualnych

- ✓ Utwórz folder na pulpicie, który posłuży jako przestrzeń dla projektu wymagań.



- ✓ Wprowadź nazwę projektu i podaj directory gdzie projektu ma się znajdować:





Zamknij to okno.

- ✓ Repozytorium zostało utworzone.

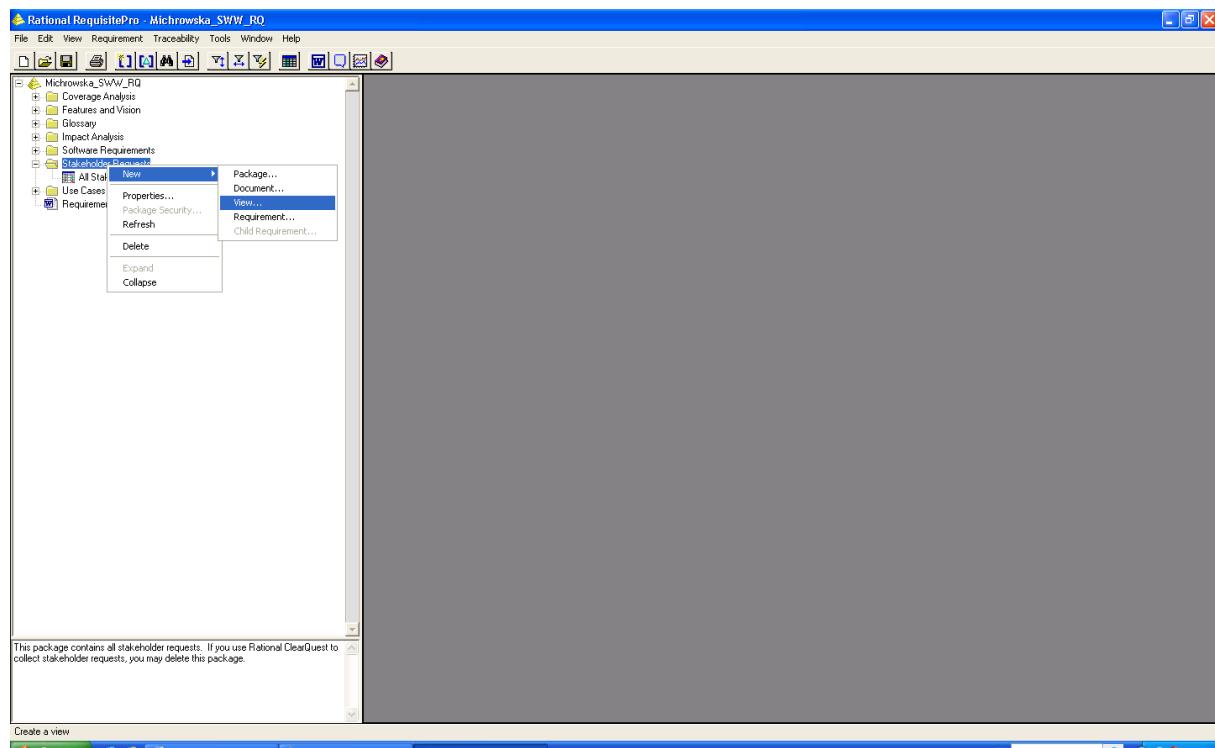
2. Tworzenie wymagań w Rational Requisite PRO

Utwórz przynajmniej:

- a. 2 wymagania typu STRQ;
- b. 2 wymagania typu TERM;
- c. 4 wymagania typu UC;

Utwórz widoki atrybutów dla opracowanych typów wymagań;

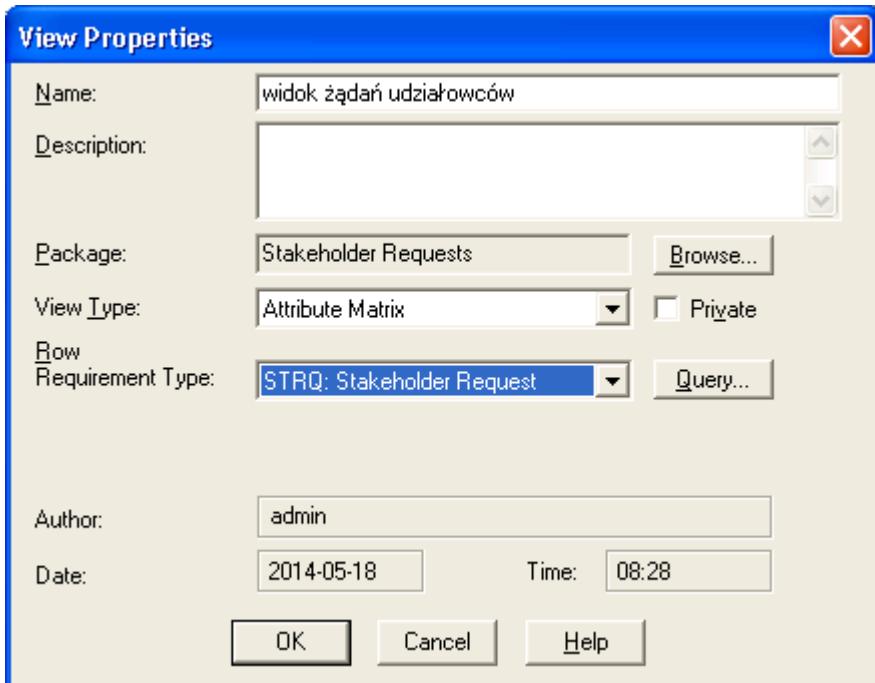
- ✓ Zaczni od utworzenia widoku wybranych wymagań, co usprawni wprowadzanie wymagań.
- ✓ Wybierz New>> view...



- ✓ Nadaj nazwę widoku.
- ✓ View Type : Attribute Matrix
- ✓ Row Requirement Type: Wybierz typ wymagania dla którego chcesz tworzyć widok.

3. widok żądań udziałowców

- ✓ Wybierz STRQ: Stakeholder Request



- ✓ Utworzono widok żądań udziałowców.
- ✓ Dodawaj wymagania w utworzonym widoku.

Pamiętaj! Wymagania należy wpisywać w polu Text, a nie NAME.

Przykład: STRQ: Poprawa informowania studentów i pracowników o terminach zajęć.

Screenshot of Rational RequisitePro interface showing the Requirements view.

The left sidebar shows project navigation with nodes like Michrowska_SWV_RQ, Features and Vision, Impact Analysis, Stakeholder Requests, Software Requirements, Use Cases, and Requirements Management Plan.

The main pane displays a table titled "Requirements" with columns: Stakeholder P, Origin, Unique ID, Location, Package, Author, Revision, Date, Reason, Traced-from, and Traced-to.

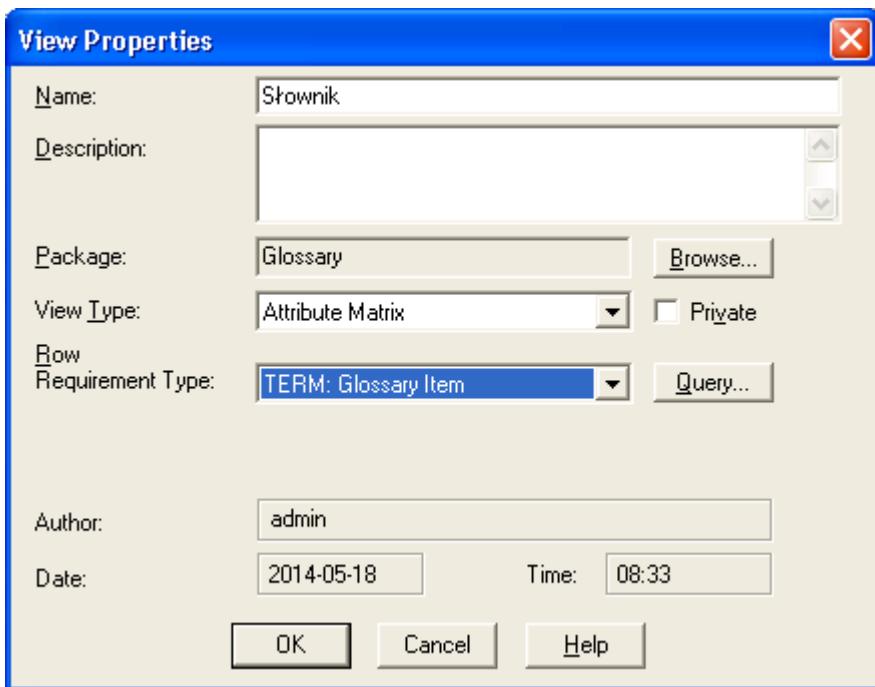
A single requirement is listed:

Requirement	Stakeholder P	Origin	Unique ID	Location	Package	Author	Revision	Date	Reason	Traced-from	Traced-to
STR01: Poprawa informowania studentów i pracowników o terminach zajęć		1	empty	Database	Stakeholder R	admin	1.0000	2014-05-18 08:08	Requirement c		

The status bar at the bottom indicates "Requirement created" and "1 requirement".

- ✓ Utwórz widok słownika.

Widok słownika:



4. Tworzenie pojęć w notacji BNF

The screenshot shows the Rational RequisitePro interface. On the left is a tree view of project components: Michrowska_SWV_RQ, Coverage Analysis, Features and Vision, Glossary, Impact Analysis, Software Requirements, Stakeholder Requests, Use Cases, and Requirements Management Plan. The 'Glossary' node is expanded, showing 'All Glossary Terms' and 'Stownik'. A requirement titled 'TERMPending' is selected. The requirement details are as follows:

Unique ID	Location	Package	Author	Revision	Date	Reason	Traced-from	Traced-to	RootTag#
empty	Database	None	admin	1.0000	2014-05-18 08				pending

Text:

```
Student to osoba posiadająca uprawnienia do uczestczania i posiadająca indeks.
```

Student= nazwisko + (ime) + nr_indeksu

This screenshot shows the same Rational RequisitePro interface as the previous one, but with a different requirement selected: 'TERMPending'. The requirement details are as follows:

Unique ID	Location	Package	Author	Revision	Date	Reason	Traced-from	Traced-to	RootTag#
3	Database	Glossary	admin	1.0000	2014-05-18 08	Requirement c		1	
empty	Database	None	admin	1.0000	2014-05-18 08				pending

Text:

```
Plan zajęć to dokument określający:  
1. W jakim terminie  
2. W jakim miejscu  
3. Odbędą się zajęcia z określonego przedmiotu.
```

Student to osoba posiadająca uprawnienia do uczestczania i posiadająca indeks.

Student= nazwisko + (ime) + nr_indeksu

Przykład: *Plan zajęć to dokument określający:*

1. *W jakim terminie*

2. *W jakim miejscu*

3. *Odbędą się zajęcia z określonego przedmiotu.*

termin = data + godzina

miejsce = nr_sali + nr_budynku

przedmiot= nazwa_przedmiotu + forma zajęc

wykaładowca = tytuł naukowy + dane pracownika

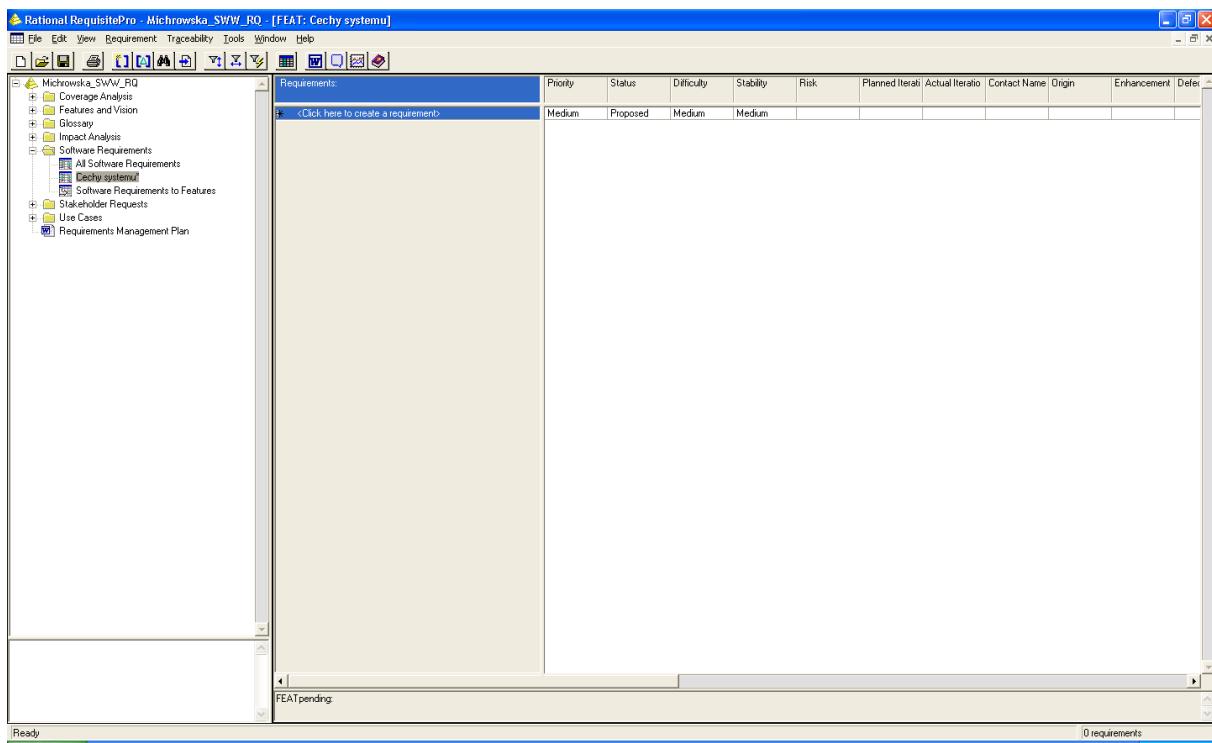
GrupaStudentow= nazwa grupy + rok studiow

planZajec = {termin + miejsce + przedmiot + wykaładowca + grupa_studentow}

Mało istotne terminy umieszczamy w definicji pojęć bardziej złożonych. W procesie modelowania wymagań postrzegamy wycinek rzeczywistości z pozycji zewnętrznej.

5. widok wymagań

- ✓ Utwórz widok wymagań typu cecha FEAT:



Screenshot of Rational RequisitePro showing a requirement named "FEATPending".

The requirement details:

Name	Priority	Status	Difficulty	Stability	Risk	Planned Iterati	Actual Iteratio	Contact Name	Origin	Enhancement	Defe
FEATPending	Medium	Proposed	Medium	Medium							

Description: System udogłowi wygenerowanie planu zajęć. Odpowiednim urządzeniem zgodnie z ich uprawnieniami. Plan zajęć może być wydrukowany oraz przekazany w formie pliku CSV.

Bottom status bar: Ready | 0 requirements

Pamiętaj! Należy akcentować fakt, że wymaganie cecha może składać się z wielu zdań.

Screenshot of Rational RequisitePro showing two requirements: FEAT1 and FEAT2.

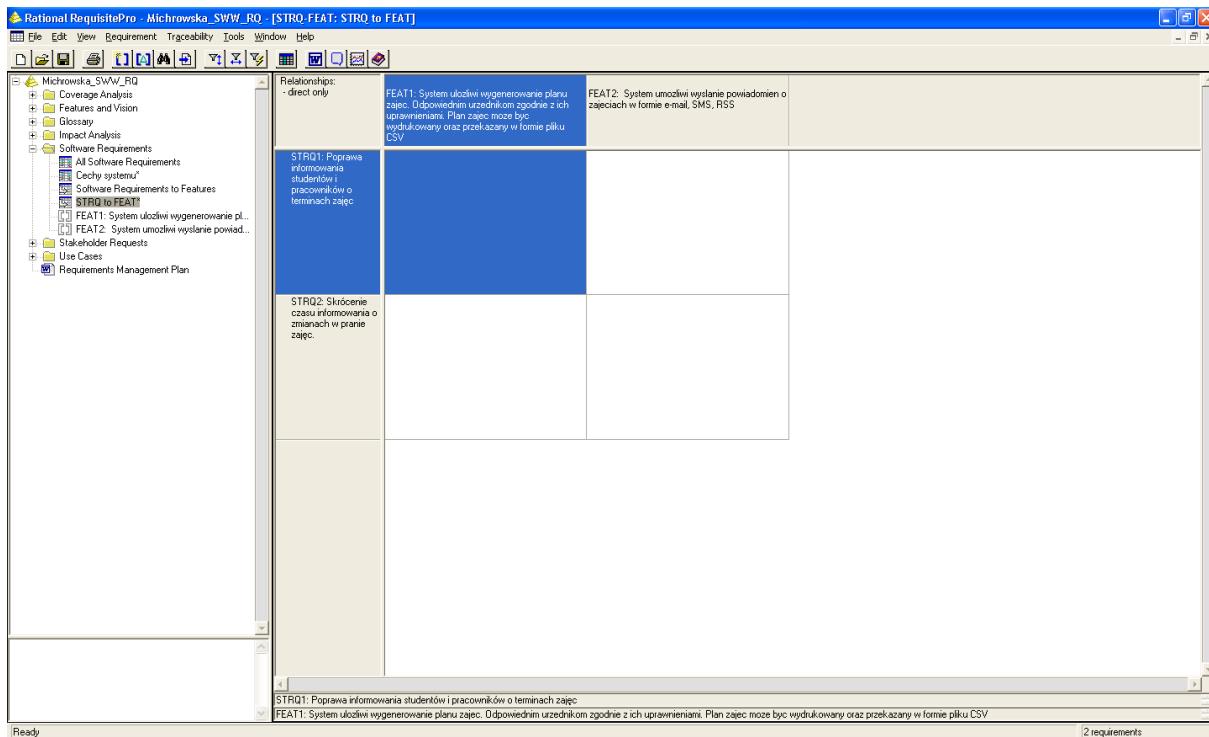
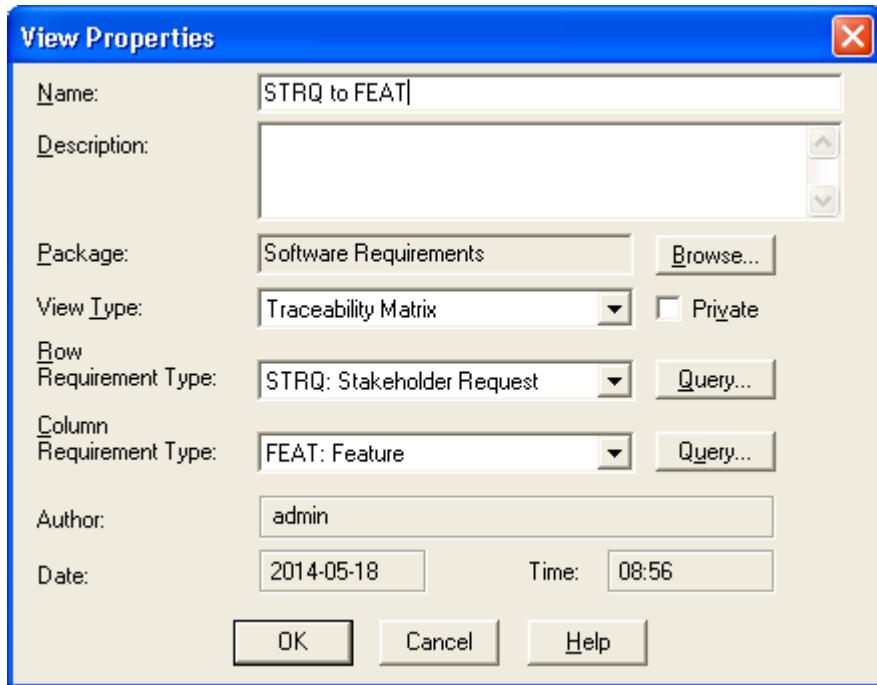
The requirement details:

Name	Priority	Status	Difficulty	Stability	Risk	Planned Iterati	Actual Iteratio	Contact Name	Origin	Enhancement	Defe
FEAT1: System udogłowi wygenerowanie planu zajęc...	Medium	Proposed	Medium	Medium							
FEAT2: System umożliwi wysłanie powiadomień o zajęciach...	Medium	Proposed	Medium	Medium							
* <Click here to create a requirement>	Medium	Proposed	Medium	Medium							

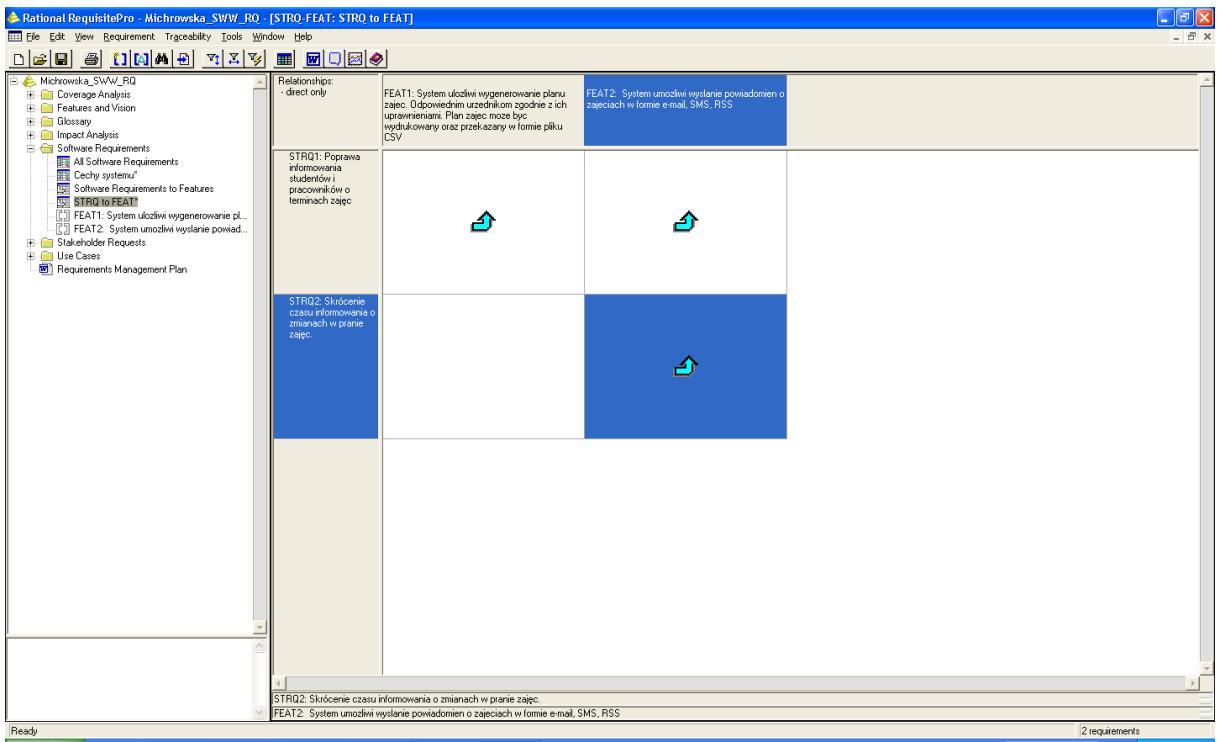
Bottom status bar: Requirement created | 2 requirements

6. widok śledzenia wymagań

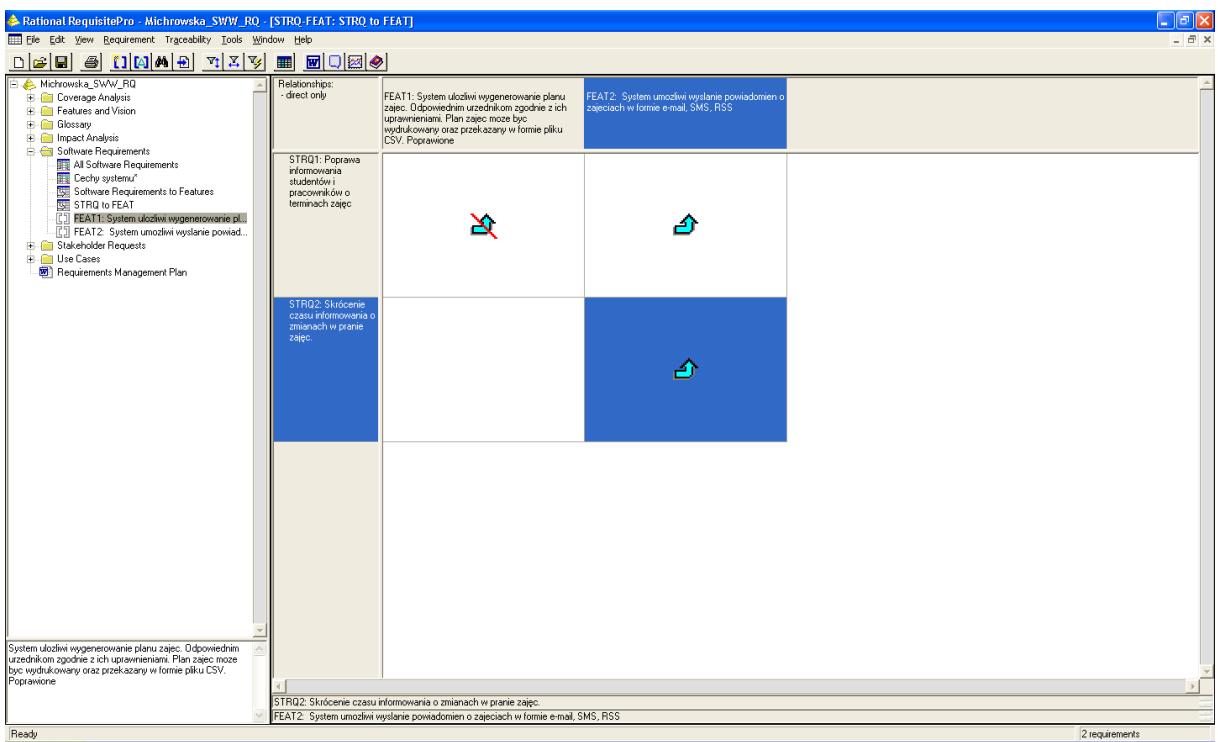
- ✓ Utwórz jeden widok śledzenia zmian STRQ to FEAT
- ✓ View_Type wybierz jako Traceability Matrix



- ✓ Wybierając prawym przyciskiem myszy na przecięciach wymagań oznacz odpowiednie wymagania jako traced to.



Pamiętaj! Należy zmienić treść co najmniej jednego wymagania, aby na macierzy był przekreślony odnośnik:



✓ Twórz widok UC.

Michałowa_SWV_RQ

- Coverage Analysis
- Features and Vision
- Glossary
- Impact Analysis
- Software Requirements
 - All Software Requirements
 - Cechy systemu*
 - Software Requirements to Features
 - STRQ to FEAT
 - FEAT1 - System udościwy generowanie pl...
 - FEAT2 - System umożliwia wyświetlanie powiad...
- Stakeholder Requests
- Use Cases
 - Przykłady użycia*
 - Use Case Survey
 - Use Cases Traced to Features
 - UC1: Wyświetl plan zajęć
- Requirements Management Plan

Wyszczególnij plan zajęć

Property	Priority	Status	Difficulty	Stability	Risk	Contact Name	Affects Archite	Planned Iterati	Actual Iterati	Enha	
UC1: Wyświetl plan zajęć	Medium	Proposed	Medium	Medium	False	Name	Medium	Proposed	Medium	Medium	False
*	<Click here to create a requirement>										

Requirement created

1 requirement

Pamiętaj, aby nazwy UC tworzyć bez polskich znaków!

7. Dokonaj strukturalizacji wymagań typu UC

Stwórz relacje pomiędzy wymaganiami, przez ustawienie rodziców, dla wymagań potomnych. Strukturalizacji wymagań można dokonywać we wszystkich typach wymagań, nie tylko w UC. W trakcie egzaminu, jest wymagane utworzenie struktury dla UC.

- ✓ Opcja change Parent.. np. wybierając prawym na wymaganiu potomnym.

Michałowa_SWV_RQ

- Coverage Analysis
- Features and Vision
- Glossary
- Impact Analysis
- Software Requirements
 - All Software Requirements
 - Cechy systemu*
 - Software Requirements to Features
 - STRQ to FEAT
 - FEAT1 - System udościwy generowanie pl...
 - FEAT2 - System umożliwia wyświetlanie powiad...
- Stakeholder Requests
- Use Cases
 - Przykłady użycia*
 - Use Case Survey
 - Use Cases Traced to Features
 - UC1: Wyświetl plan zajęć
 - UC1: Wyszczególnij plan zajęć (we wskazanym terminie)
 - UC3 Udoskonalenie
- Requirements Management Plan

Wyszczególnij plan zajęć (we wskazanym terminie)

Property	Priority	Status	Difficulty	Stability	Risk	Contact Name	Affects Archite	Planned Iterati	Actual Iterati	Enha	
UC1: Wyświetl plan zajęć	Medium	Proposed	Medium	Medium	False	Name	Medium	Proposed	Medium	Medium	False
UC1: Wyszczególnij plan zajęć (we wskazanym terminie)	Medium	Proposed	Medium	Medium	False	Name	Medium	Proposed	Medium	Medium	False
*	<Click here to create a requirement>										

Multiple requirements selected.

Change the parent of the selected requirements

Start Zadania na egzaminie... Tools Rational RequisitePro... PL Wyszukaj na pulpicie 09:14

The screenshot shows the Rational RequisitePro interface with the title bar "Rational RequisitePro - Michrowska_SW_RQ - [UC: Przykadki użycia]". The menu bar includes File, Edit, View, Requirement, Traceability, Tools, Window, Help. The toolbar has icons for New, Open, Save, Print, and others. The left sidebar displays a project tree:

- Michrowska_SW_RQ
 - Coverage Analysis
 - Features and Vision
 - Glossary
 - Impact Analysis
 - Software Requirements
 - All Software Requirements
 - Cechy systemu*
 - Software Requirements to Features
 - STRQ to FEAT
 - FEAT1: System udostępnianie wygenerowanej pl...
 - FEAT2: System umożliwi wysłanie powiad...
 - Stakeholder Requests
 - Use Cases
 - Przykadki użycia*
 - Use Case Survey
 - Use Case Template

A red box highlights the "Use Cases" node and its children. The main workspace shows a "Requirements" table:

Property	Priority	Status	Difficulty	Stability	Risk	Contact Name	Affects Archite	Planned Iterati	Actual Iteratio	Enhanc
Name	Medium	Proposed	Medium	Medium						
Name	Medium	Proposed	Medium	Medium						
Name	Medium	Proposed	Medium	Medium						
Name	Medium	Proposed	Medium	Medium						
* <Click here to create a requirement>										

The table contains five rows of requirements, all with "Name" as the property, "Medium" as priority, "Proposed" as status, "Medium" as difficulty, and "Medium" as stability. The risk column is empty. The contact name and affects architecture columns are also empty. The planned iteration and actual iteration columns are empty. The enhancement column is empty.

The status bar at the bottom shows "Multiple requirements selected." and "3 requirements".

The screenshot shows the Rational RequisitePro interface. On the left is a tree view of project components: Michrowska_SWW_RQ, Coverage Analysis, Features and Vision, Glossary, Impact Analysis, Software Requirements (with All Software Requirements and Czech systems' children), Software Requirements to Features (with STRG to FEAT and FEAT1: System udogwiający generowanie plików XML and FEAT2: System umożliwiający wysłanie powiadomień e-mail), Stakeholder Requests, Use Cases (with Przykłady użycia*), Use Case Survey, Use Cases Traced to Features, UCT1: Wyświetl plan zajęć, and Requirements Management Plan. A red box highlights the 'UCT1: Wyświetl szczegółowy plan zajęć (we wskazanym terminie)' node under Use Cases.

A context menu is open over this node, with 'Requirement...' highlighted. The menu also includes 'New...', 'Change Parent...', 'Change Type...', 'Properties...', 'Go To', 'Discussions...', 'Delete', 'Expand', and 'Collapse' options.

The main workspace displays a table titled 'Requirements' with columns: Property, Priority, Status, Difficulty, Stability, Risk, Contact Name, Affects Archite, Planned Iteratio, Actual Iteratio, and Erkha. It lists requirements like 'UCT1: Wyświetl plan zajęć' and its sub-items 'UCT1.1: Wyświetl szczegółowy plan zajęć (we wskazanym terminie)', 'UCT1.1.1: Udostępnij dane o miesiąc wskazanych zajęć', and 'UCT1.1.2: Udostępnik dane o grupie studentów...'. Below the table is a note: 'Multiple requirements selected.'

At the bottom left, a status bar says 'Create a child requirement of the selected requirement.'

Rational RequisitePro - Michrowska_SWV_RQ - [UC: Przykładki użycia]

File Edit View Requirement Traceability Tools Window Help

Michrowska_SWV_RQ

- Coverage Analysis
- Features and Vision
- Glossary
- Impact Analysis
- Software Requirements
 - All Software Requirements
 - Software Requirements to Features
 - STRQ to FEAT
 - FEAT1: System udogodniwy generowanie pl...
 - FEAT2: System umożliwia wysłanie powiad...
- Use Cases
 - Przykładki użycia*
 - Use Case Survey
 - Use Cases Traced to Features
 - UC1: Wyświetl plan zajęć
 - UC1.1: Wyświetl szczegółowy plan zajęć (we wskazanym terminie)
 - UC1.1.1: Udostępnij dane o miejscu wskazanych zajęć
 - UC1.1.2: Udostępnij dane o grupie studentów uczestniczących w zajęciach
 - UC1.2: Udostępnij dane o grupie studentów uczestniczącej w zajęciach
- Requirements Management Plan

Udostępnik dane o grupie studentów uczestniczącej w zajęciach

Multiple requirements selected.

Requirement created

3 requirements

Requirements	Name	Priority	Status	Difficulty	Stability	Risk	Contact Name	Affects Archive	Planned Iteration	Actual Iteration	Entered
UC1: Wyświetl plan zajęć	Name	Medium	Proposed	Medium	Medium						False
UC1.1: Wyświetl szczegółowy plan zajęć (we wskazanym terminie)	Name	Medium	Proposed	Medium	Medium						False
UC1.1.1: Udostępnij dane o miejscu wskazanych zajęć	Name	Medium	Proposed	Medium	Medium						False
UC1.1.2: Udostępnij dane o grupie studentów uczestniczących w zajęciach	Name	Medium	Proposed	Medium	Medium						False

Zadaniem na egzaminie będzie utworzenie 4 wymagań typu UC.

Rational RequisitePro - Michrowska_SWV_RQ - [UC: Przykładki użycia]

File Edit View Requirement Traceability Tools Window Help

Michrowska_SWV_RQ

- Coverage Analysis
- Features and Vision
 - Vision Document
 - All Features
 - Cechy systemu
 - Features to Stakeholder Requests
 - FEAT1: System udogodniwy generowanie planu zajęć. Dostępny...
 - FEAT2: System umożliwia wysłanie powiadomień o zakończeniu zajęć...
 - FEAT3: System umożliwia załatwianie procesu planowania...
- Glossary
- Impact Analysis
- Software Requirements
 - All Software Requirements
 - Software Requirements to Features
 - STRQ to FEAT
 - SR1: Dostępność planu 365/12/24 na poziomie 99.5 %
- Stakeholder Requests
 - All Stakeholder Requests
 - widok zgód i udziałowców
 - STRQ1: Poprawa informowania studentów i pracowników o tematach...
 - STRQ2: Skrócenie czasu informowania o zmianach w planie zajęć
 - STRQ3: Zmniejszenie ilości błędów popełnianych przy planowaniu...
 - STRQ4: Skrócenie nakładu pracy potrzebnej na ułożenie konkret...
- Use Cases
 - Przykładki użycia*
 - Use Case Survey
 - Use Cases Traced to Features
 - UC1: Wyświetl plan zajęć
 - UC1.1: Wyświetl szczegółowy plan zajęć (we wskazanym terminie)
 - UC1.1.1: Udostępnij dane o miejscu wskazanych zajęć
 - UC1.1.2: Udostępnij dane o grupie studentów uczestniczących w zajęciach
 - UC1.2: Udostępnij dane o grupie studentów uczestniczącej w zajęciach
 - UC4: Wyświetl sugerowany plan zajęć
 - UC4.1: Wprowadź żądania do sugerowanego planu zajęć
 - UC4.2: Wprowadź zmiany w sugerowanym planie zajęć
 - UC4.2.1: Wyświetl komunikat informujący o zajętości sali
 - UC4.2.2: Wyświetl komunikat o zajętości wykładowcy
 - UC4.2.3: Wyświetl komunikat o zajętości grupy
- Requirements Management Plan

Wyswietl komunikat o zajętości grupy

Multiple requirements selected.

Ready

10 requirements

Requirements	Name	Priority	Status	Difficulty	Stability	Risk	Contact Name	Affects Archive	Planned Iteration	Actual Iteration	Entered
UC1: Wyświetl plan zajęć	Name	Medium	Proposed	Medium	Medium						False
UC1.1: Wyświetl szczegółowy plan zajęć (we wskazanym terminie)	Name	Medium	Proposed	Medium	Medium						False
UC1.1.1: Udostępnij dane o miejscu wskazanych zajęć	Name	Medium	Proposed	Medium	Medium						False
UC1.1.2: Udostępnij dane o grupie studentów uczestniczących w zajęciach	Name	Medium	Proposed	Medium	Medium						False
UC1.2: Udostępnij dane o grupie studentów uczestniczącej w zajęciach	Name	Medium	Proposed	Medium	Medium						False
UC4: Wyświetl sugerowany plan zajęć	Name	Medium	Proposed	Medium	Medium						False
UC4.1: Wprowadź żądania do sugerowanego planu zajęć	Name	Medium	Proposed	Medium	Medium						False
UC4.2: Wprowadź zmiany w sugerowanym planie zajęć	Name	Medium	Proposed	Medium	Medium						False
UC4.2.1: Wyświetl komunikat informujący o zajętości sali	Name	Medium	Proposed	Medium	Medium						False
UC4.2.2: Wyświetl komunikat o zajętości wykładowcy	Name	Medium	Proposed	Medium	Medium						False
UC4.2.3: Wyświetl komunikat o zajętości grupy	Name	Medium	Proposed	Medium	Medium						False

- ✓ Utwórz widok śledzenia STRQ to FEAT

8. Dokonaj eksportu tych widoków do pliku RaportWymagań.doc

- ✓ Mając otwarty widok, który chcesz wyeksportować wybierz **File>>Export>>Export to Word**

- ✓ Zapisz plik w wybranym miejscu.
 ✓ Export, został wykonany.
 ✓ Zostało zakończone modelowanie werbalne, należy przejść do elementu modelowania wizualnego w Rational Software Architect.

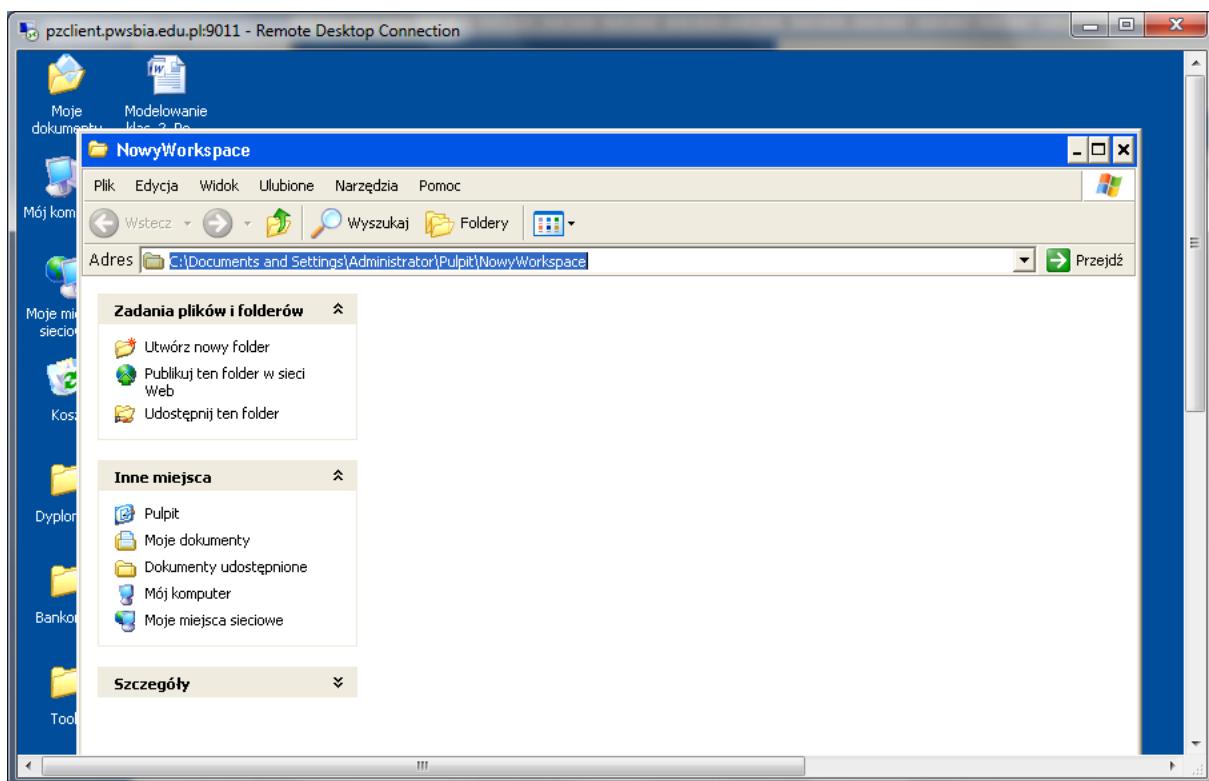
9. Repozytorium wymagań wizualnych (model wymagań - na podstawie wzorca „pusty”)

Przed przejściem do realizacji tego zadania, musisz mieć skonfigurowany workspace i środowisko, tak aby tworzyć poprawny model. Poniższe punkty odnoszą się do konfiguracji środowiska i przestrzeni roboczej (piaskownicy-sandbox)

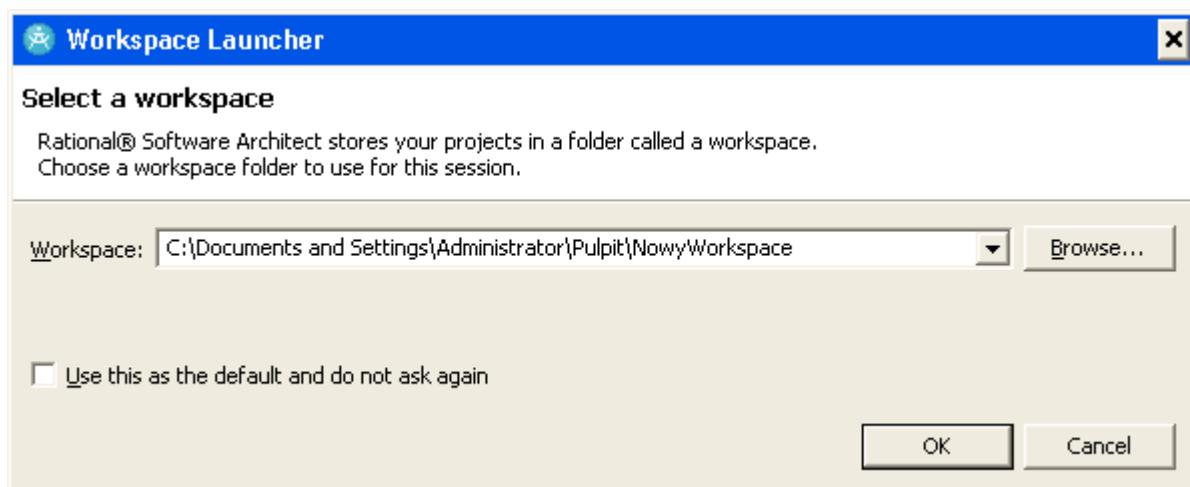
10. Tworzenie Worspace

- ✓ Na pulpicie utwórz nowy folder, który posłuży jako workspace do wykonania zadania. Lepiej, żeby nie był to ten sam folder, w którym tworzyłeś model w Rational Requisite Pro, powoduje to problem przy późniejszej asocjacji repozytoriów.
- ✓ Skopiuj ścieżkę do folderu- trzeba będzie podać przy otwieraniu RSA

C:\Documents and Settings\Administrator\Pulpit\NowyWorkspace

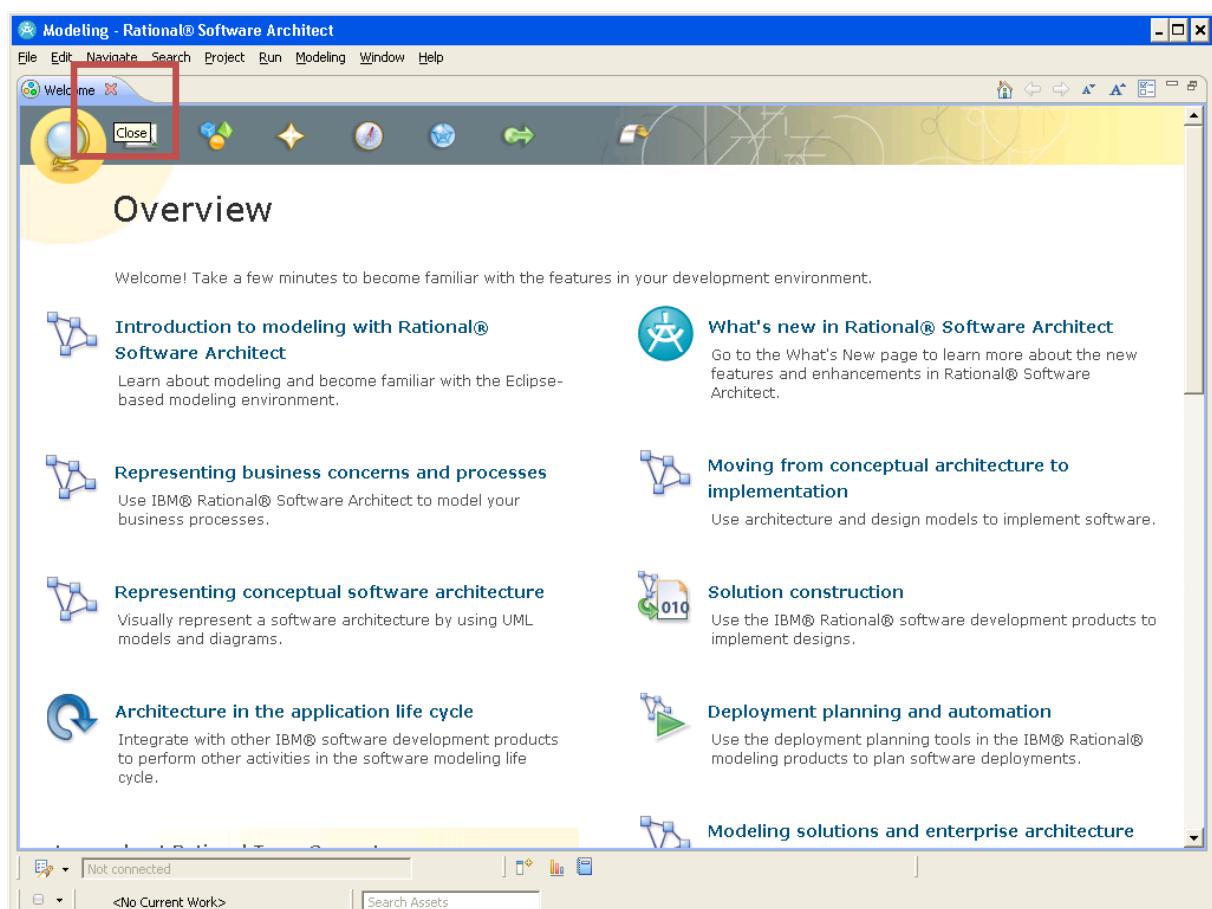
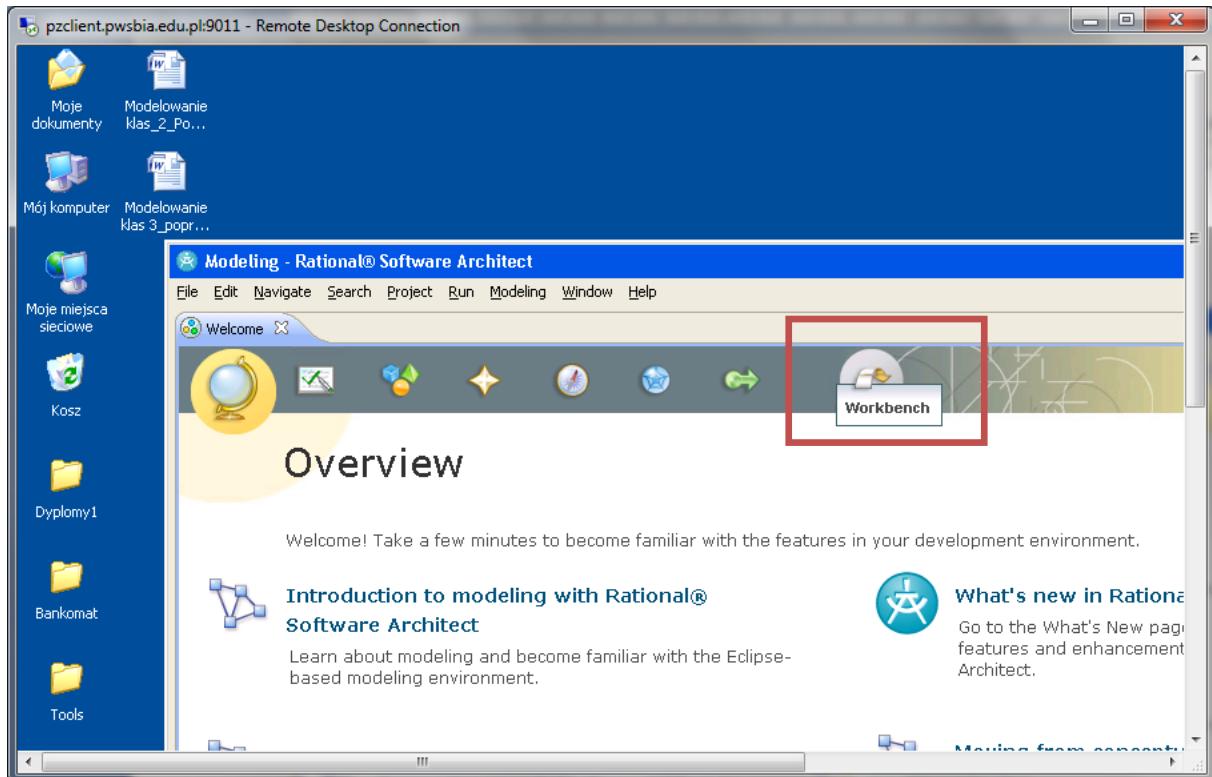


- ✓ Otwórz RSA podając skopowaną ścieżkę:



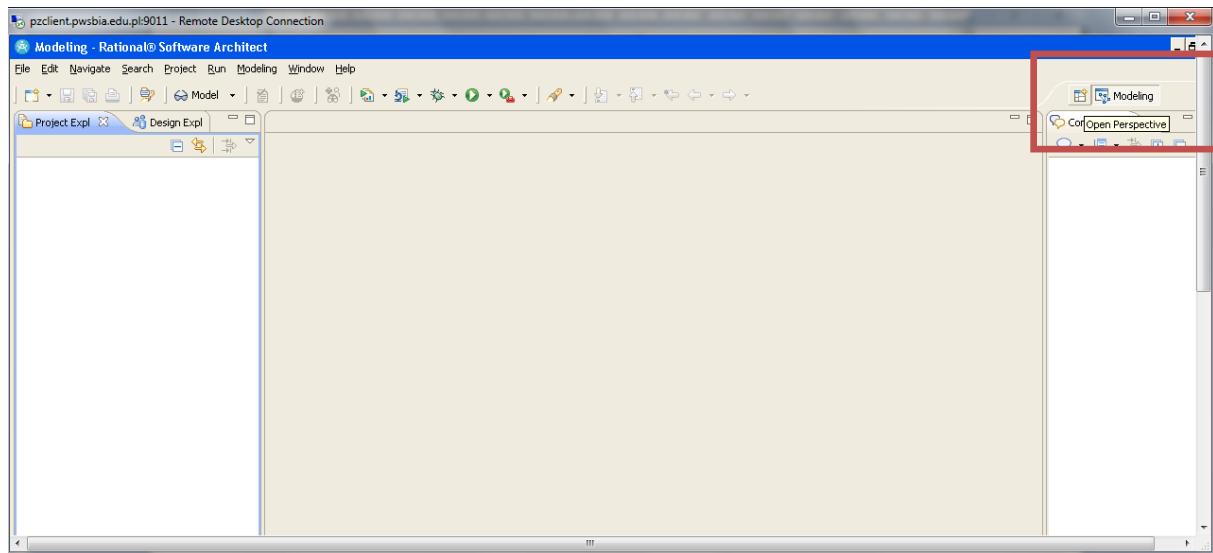
- ✓ Zatwierdź OK. i zaczekaj na otwarcie aplikacji.

- ✓ Wybierz Workbench lub zamknij to okno:

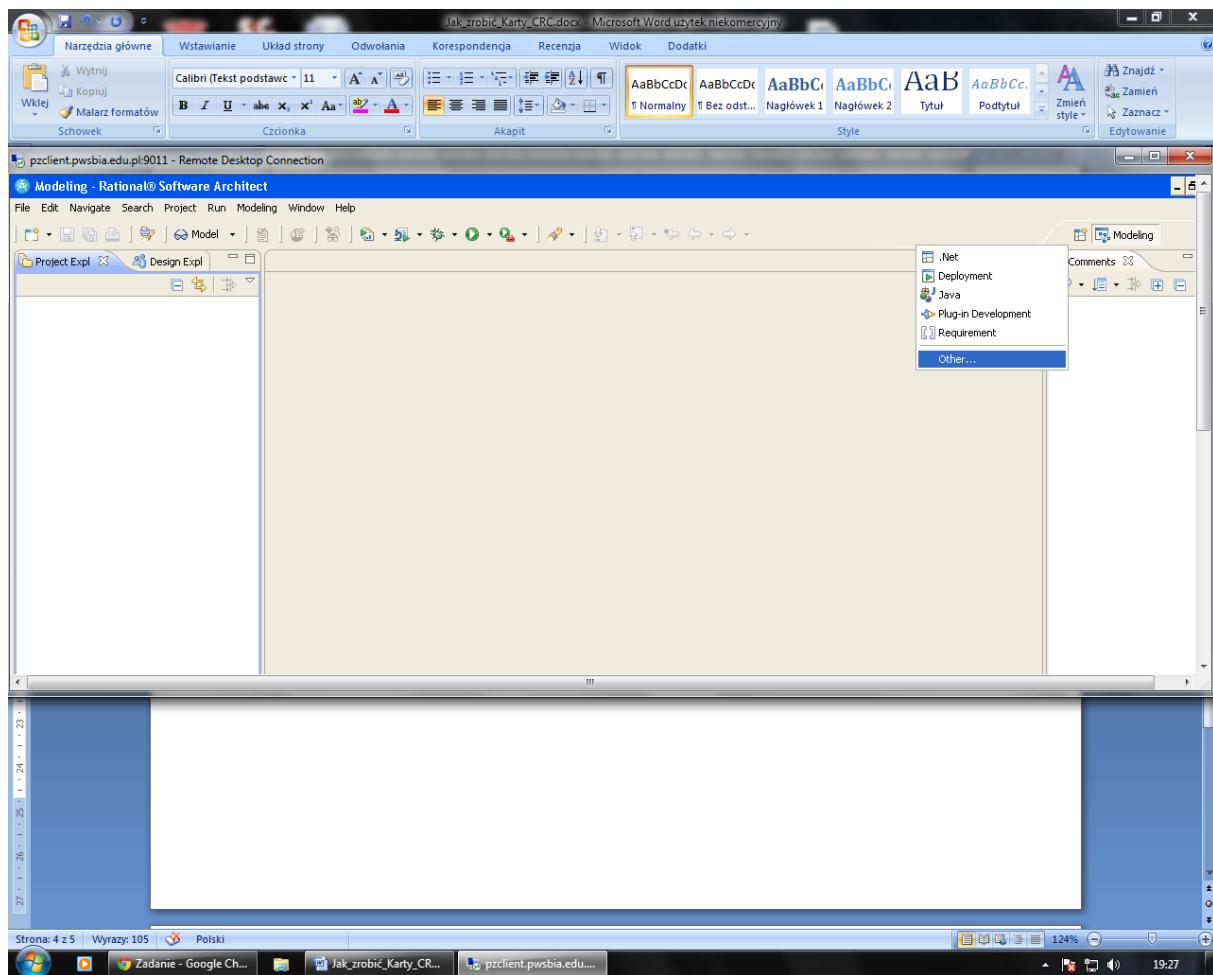


11. Wybór perspektywy

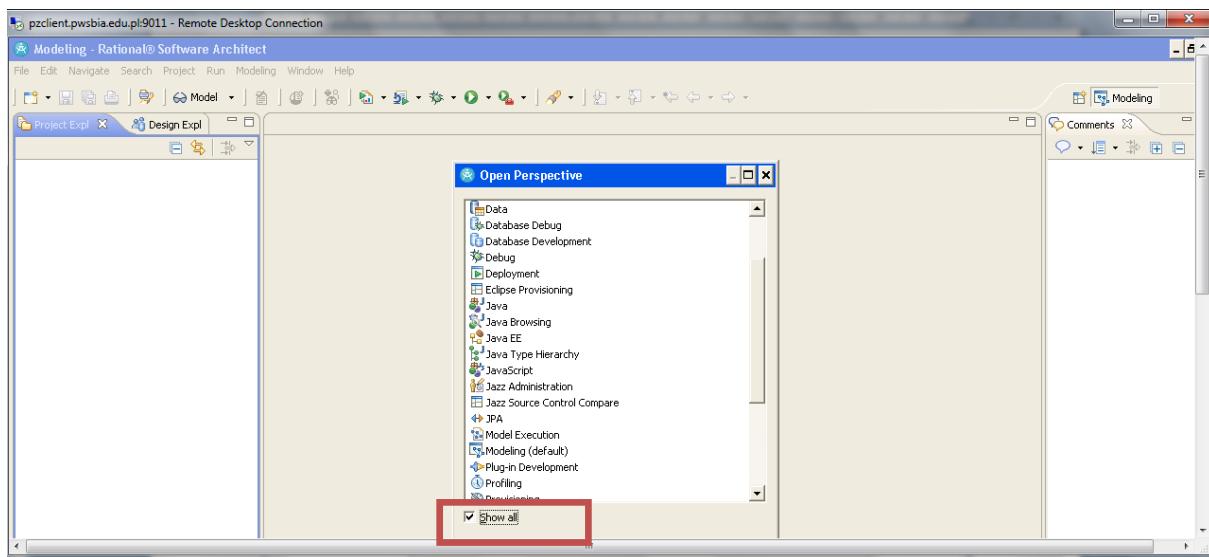
Wybierz perspektywę Jazz Administration (w prawym górnym rogu) lub z menu Window.



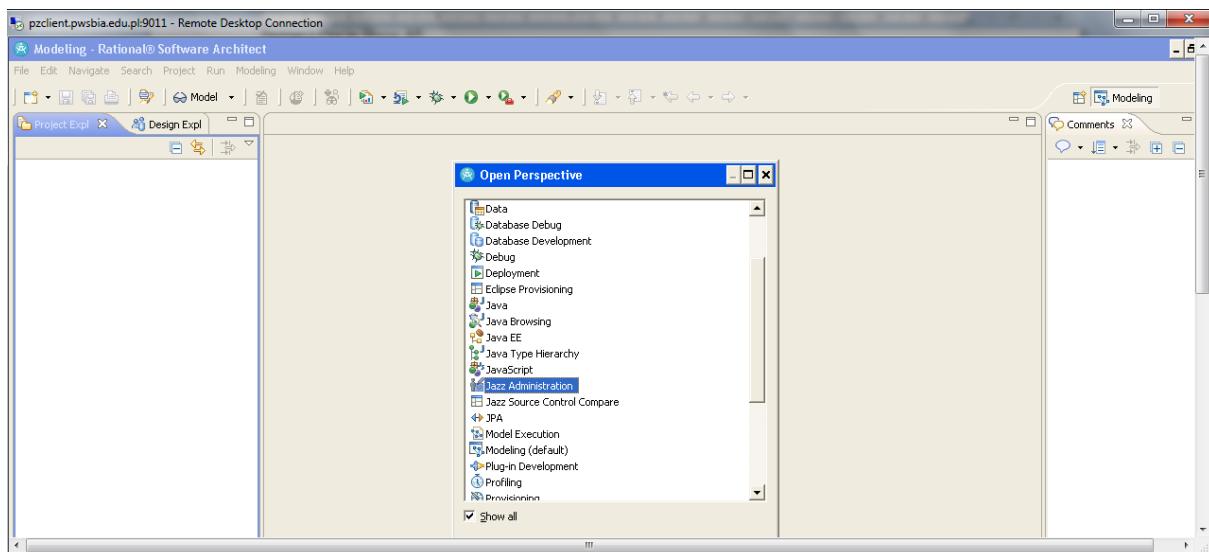
- ✓ Wybierz Other, w celu uzyskania widoku wszystkich perspektyw:



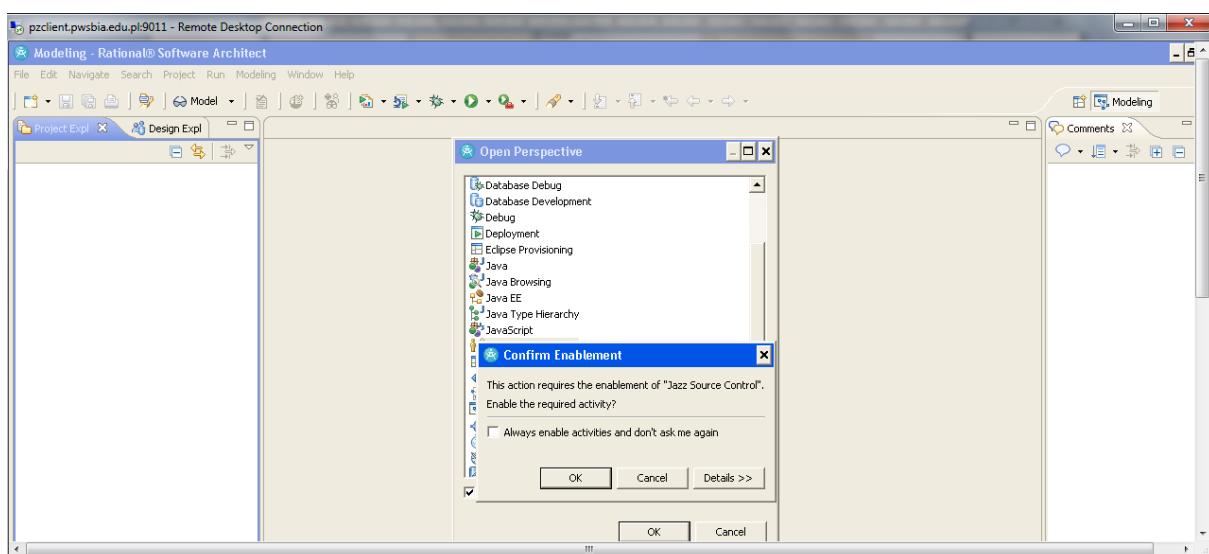
- ✓ W wyświetlonym oknie zaznacz opcję Show All:



- ✓ wybierz Jazz Administration:



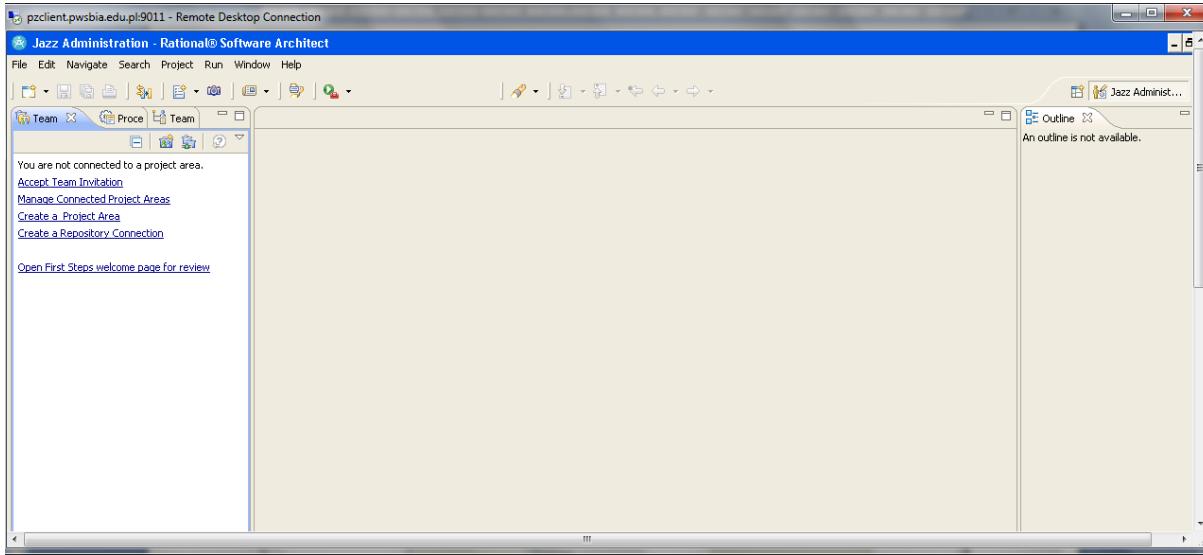
- ✓ Pojawiło się okno z pytaniem o potwierdzenie uaktywnienia Jazz Source Control, daj OK.



12. Tworzenie połączenia z repozytorium.

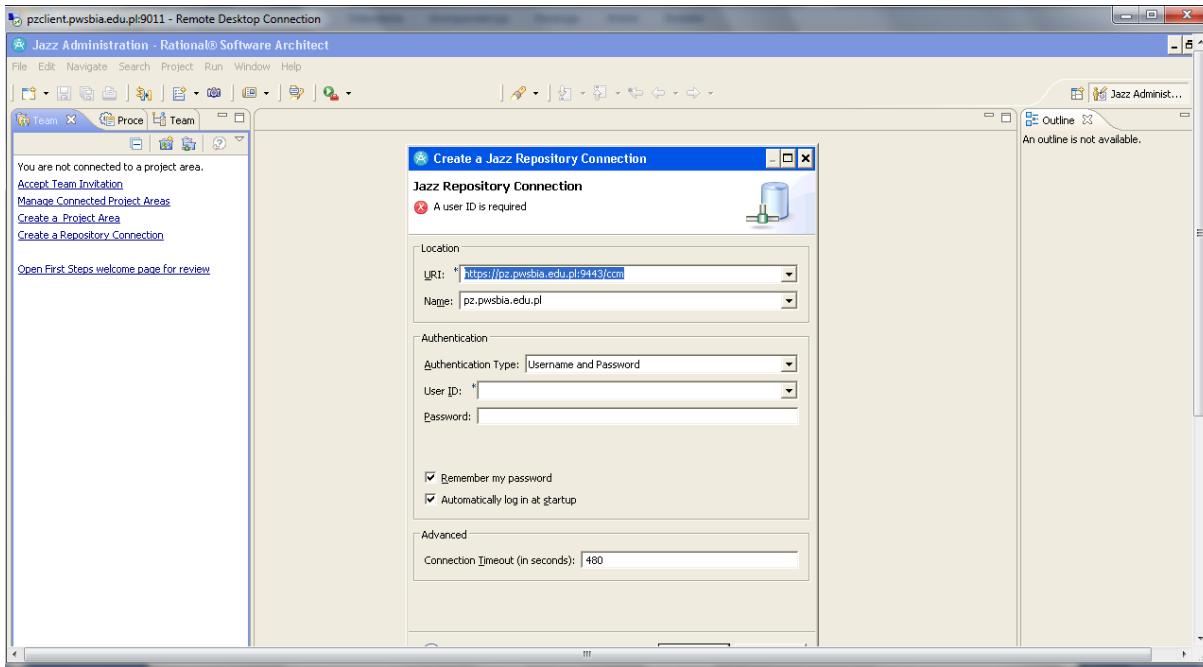
Utwórz połączenie z repozytorium.

- ✓ wybierz Create a Repository Connection, może być w zakładce TEAM lub TEAM ARTIFACTS

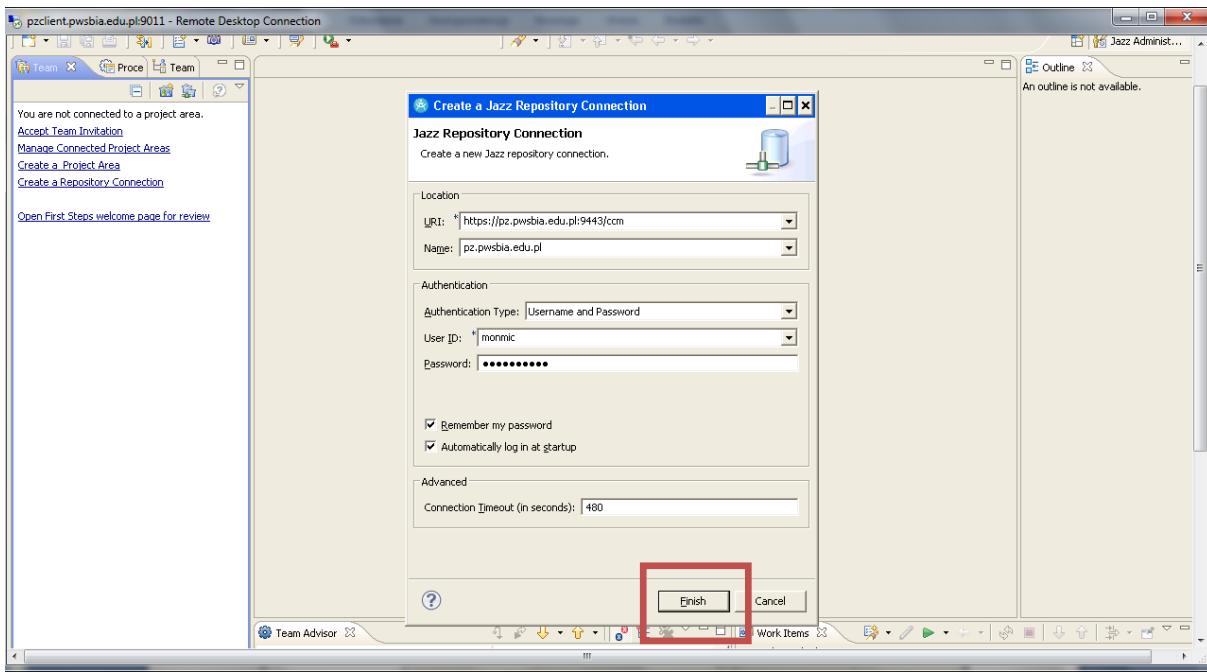


Uzupełnij wymagane dane:

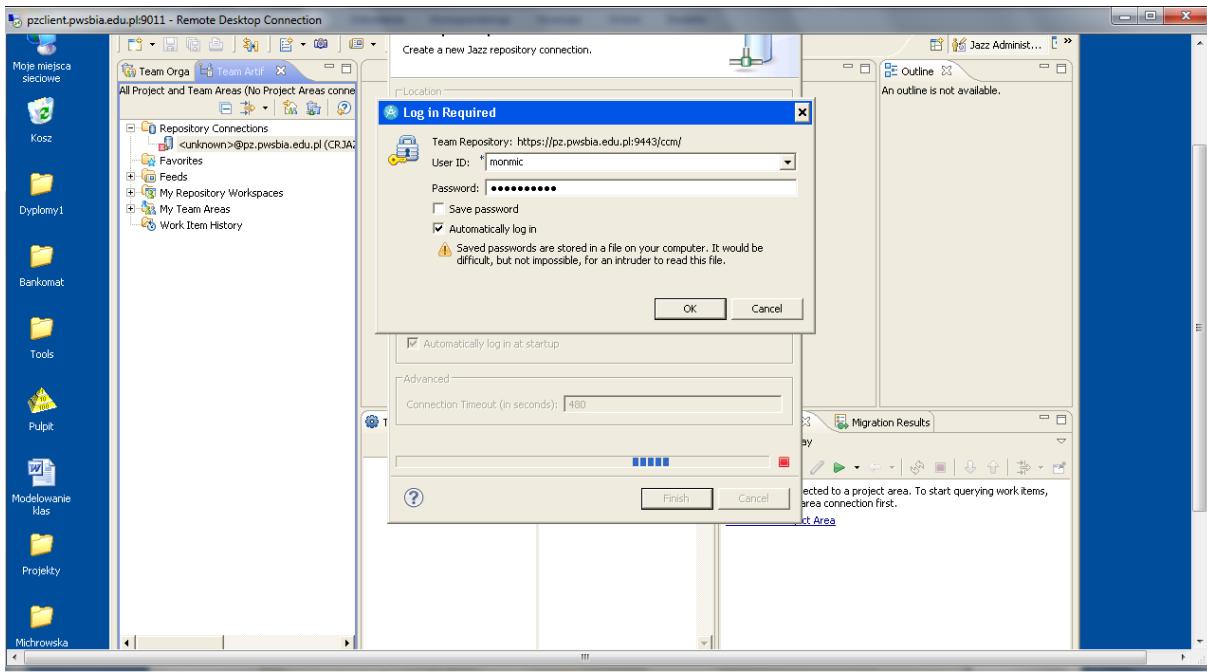
- ✓ URI: <https://pz.pwsbia.edu.pl:9443/ccm>
- ✓ User ID: login do Jazz
- ✓ Password: hasło do Jazz
- ✓ Odznacz opcję Remember my password



Button Finish jest teraz aktywny, wybierz go:



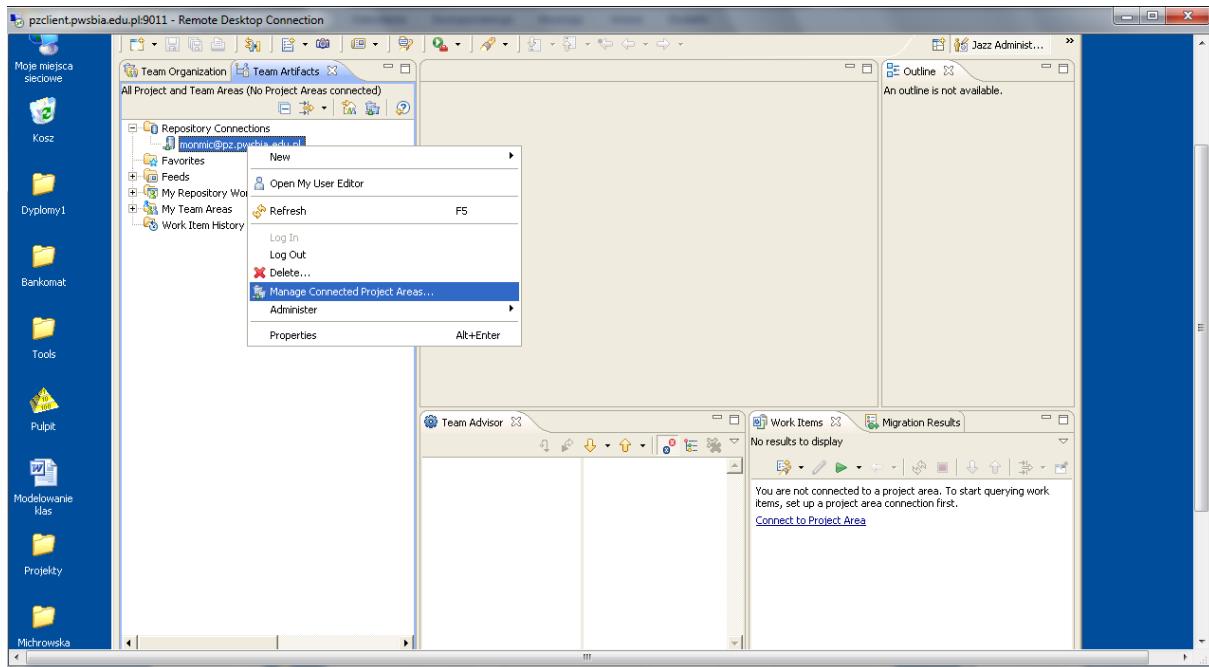
- ✓ Wybierz Finish.
- ✓ Podaj ponownie hasło do repozytorium.
- ✓ Nie wybieraj opcji Save Password.



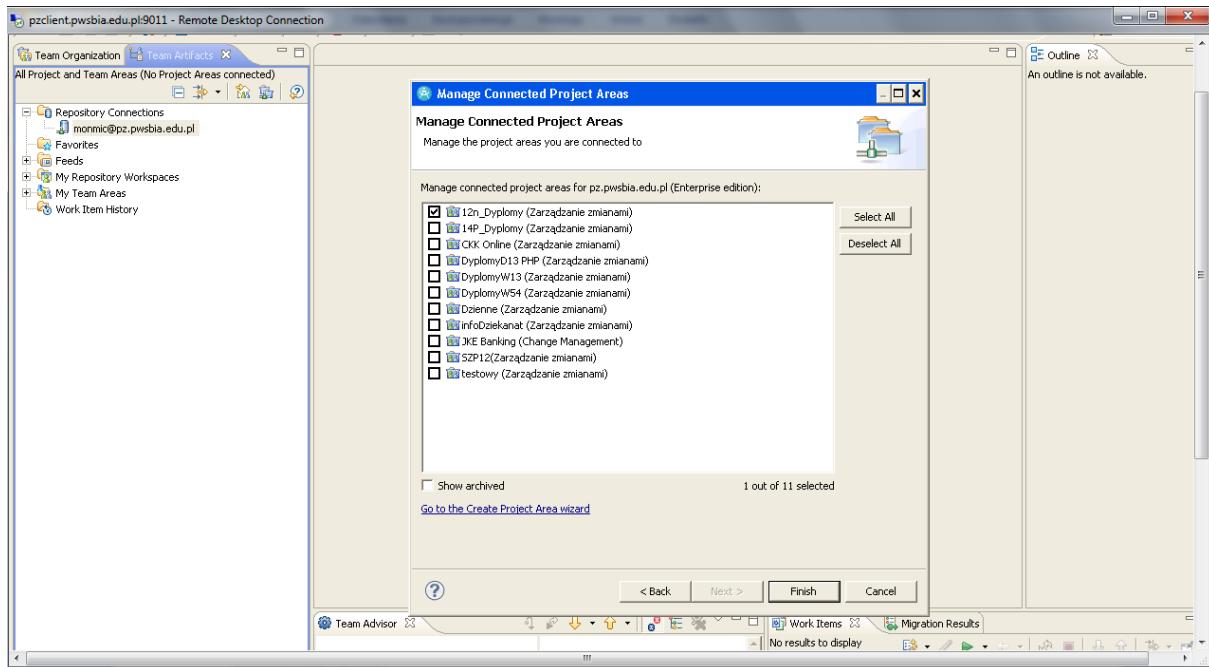
- ✓ Wybierz OK.
- ✓ Połączenie z repozytorium zostało skonfigurowane, jest potrzebne aby mieć dostęp do Twoich zasobów, oraz aby móc przesyłać realizowane zmiany na serwer w celu udostępnienia ich innym członkom zespołu (w tym profesorowi do oceny).

13. Zarządzanie obszarami projektu.

- ✓ W zakładce **Team Artifacts** rozwiń folder repository connection
- ✓ wybierz połączenie z repozytorium, następnie
- ✓ wybierz opcję zarządzania obszarami projektu:



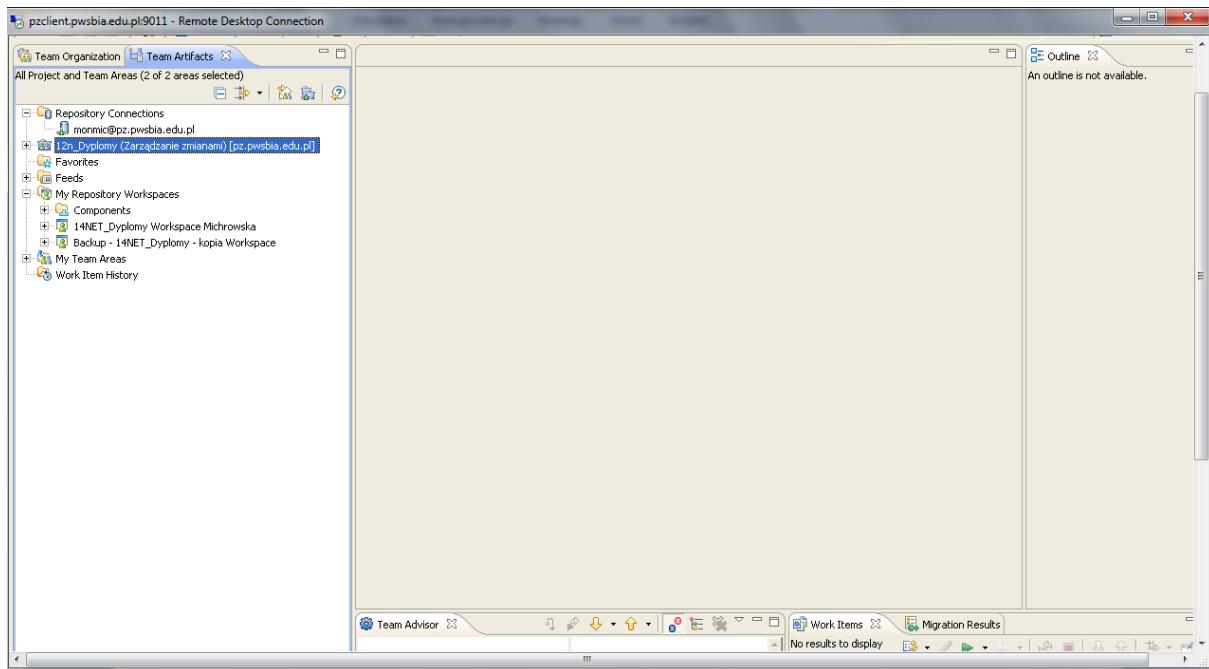
- ✓ Wybierz 12n_Diplomaty lub 14P_Diplomaty w zależności od grupy, w której pracujesz.



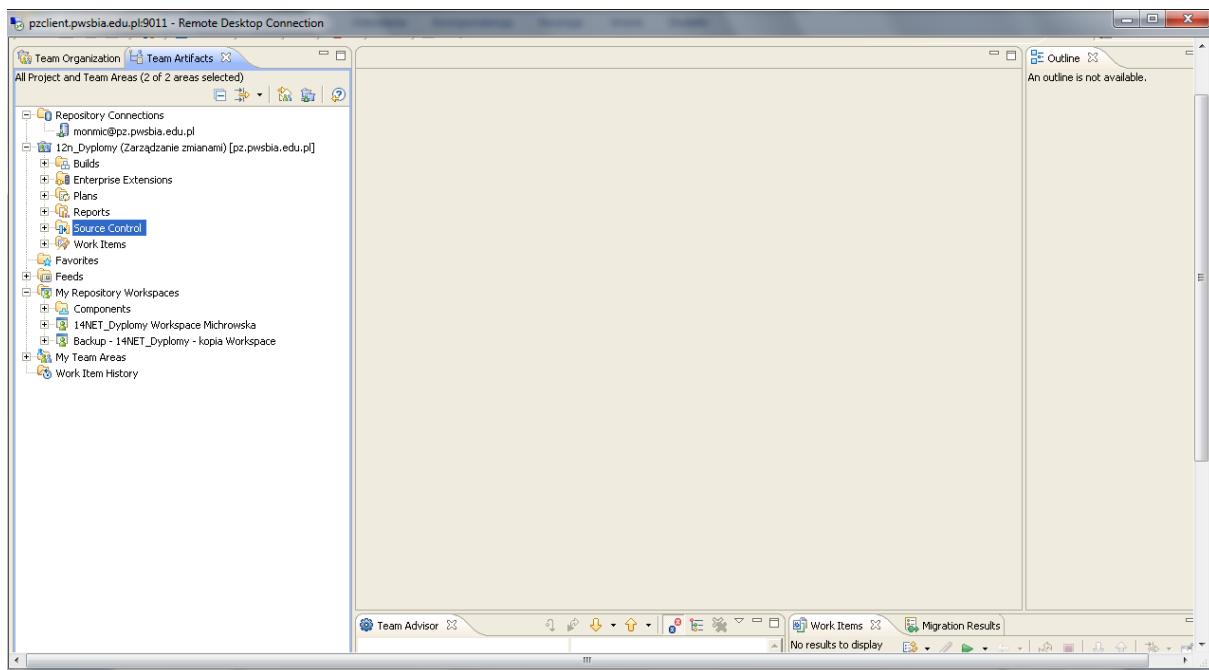
- ✓ Wybierz Finish.
- ✓ Połączenie z obszarem projektu, do którego będą przesyłane zmiany zostało utworzone.

14. Tworzenie nowego workspace w repozytorium.

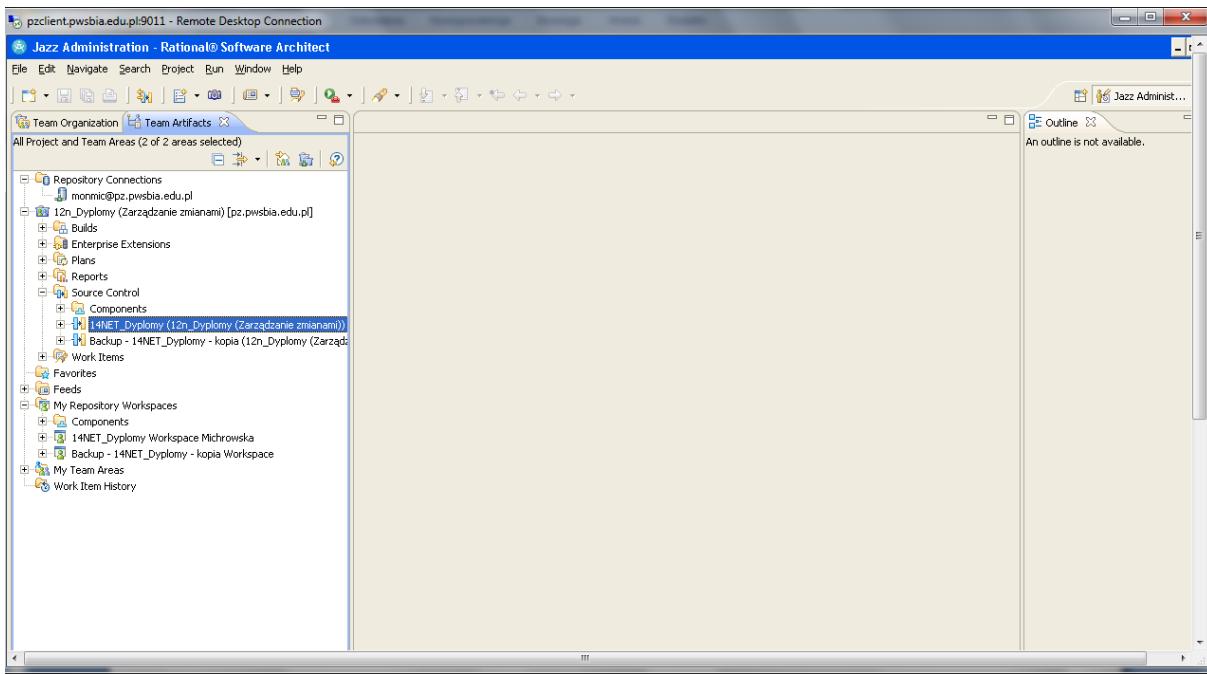
Na zakładce Team Artifacts pojawiło się wybrany obszar projektu:



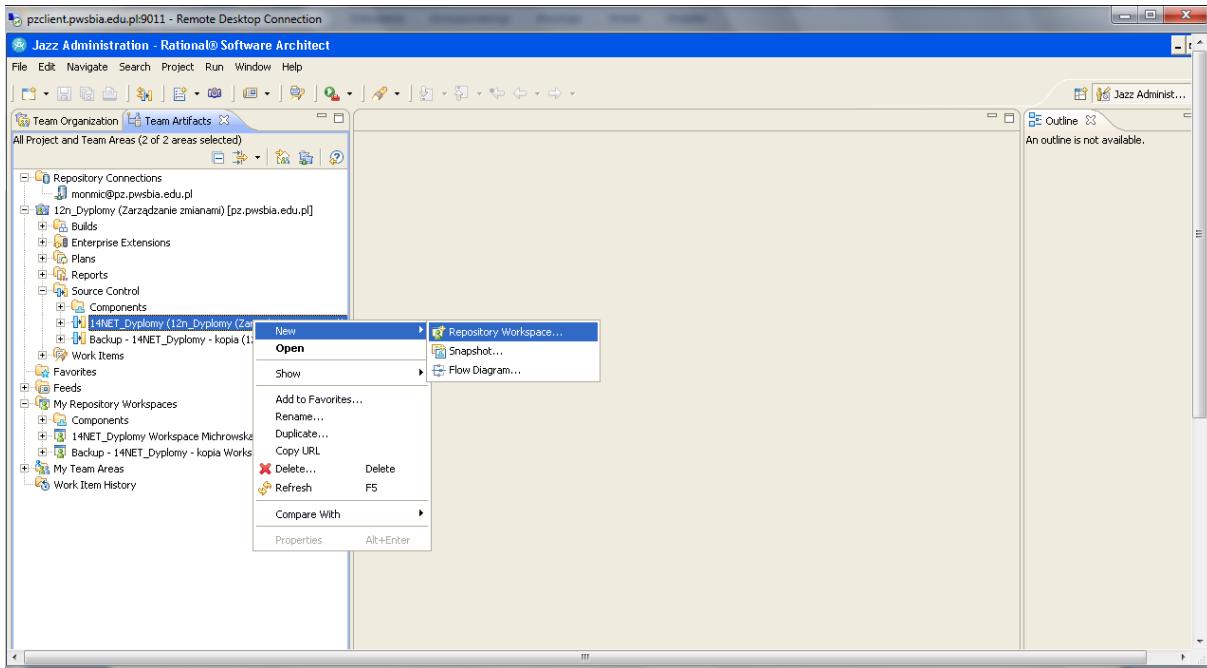
- ✓ Rozwiń go.
- ✓ Przejdź do folderu Source Control i rozwiń ten folder.



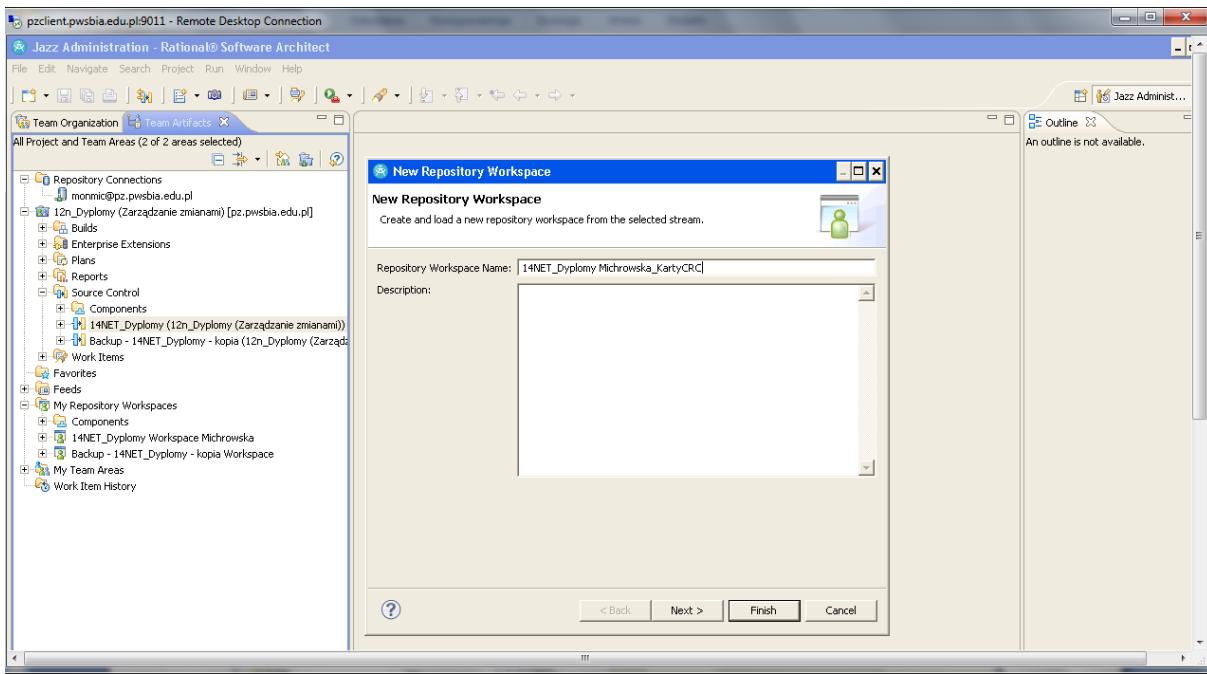
- ✓ Wybierz komponent 14NET_Diplomu (12n_Diplomu (zarządzanie zmianami)) czy jaką tam masz nazwę obszaru dostępną, z wyjątkiem backup.



- ✓ Wybierz prawym na komponencie >>New >>Repository Workspace

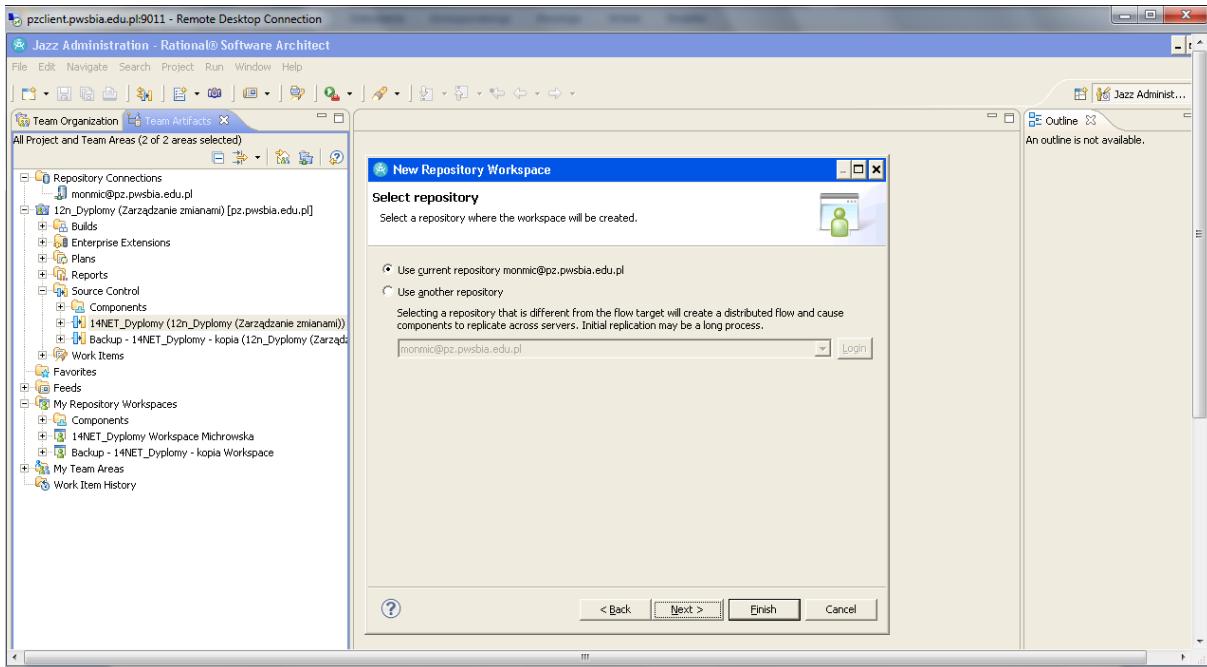


- ✓ Nadaj nazwę workspace, która będzie dla Ciebie znacząca. To właśnie tą przestrzeń będziesz ładował do przestrzeni roboczej i za jej pośrednictwem przesyłał zmiany do repozytorium.



✓ Wybierz Next

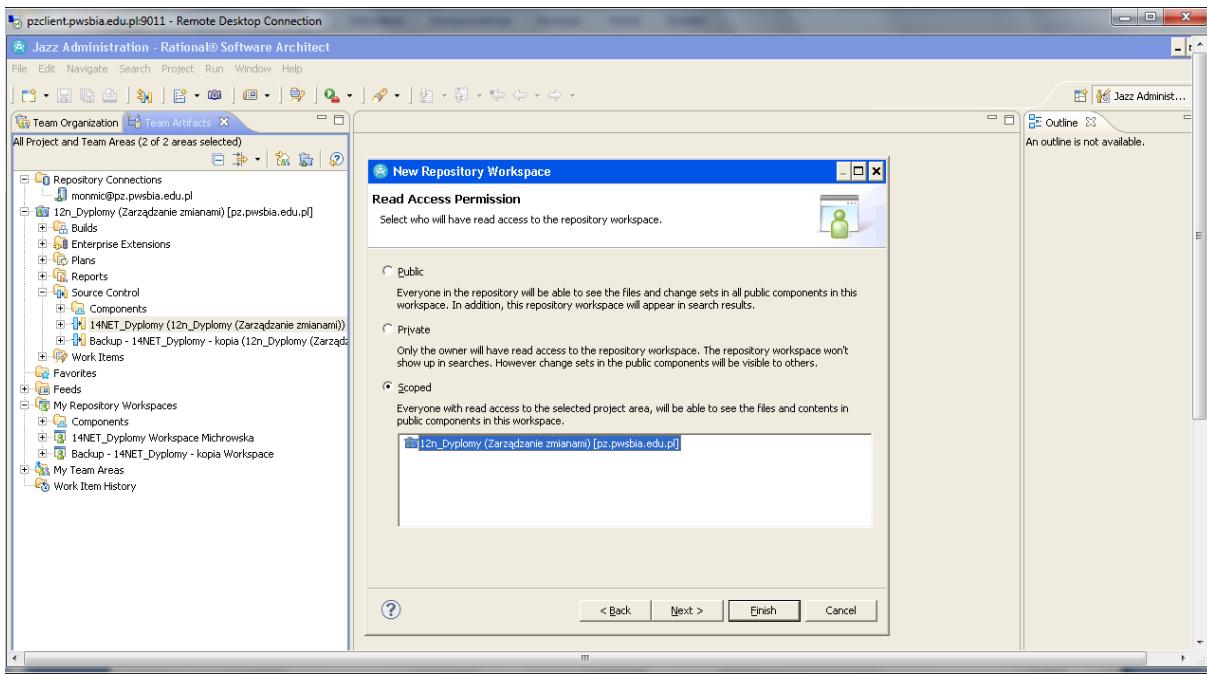
Na następnej stronie pozostaw wszystko bez zmian:



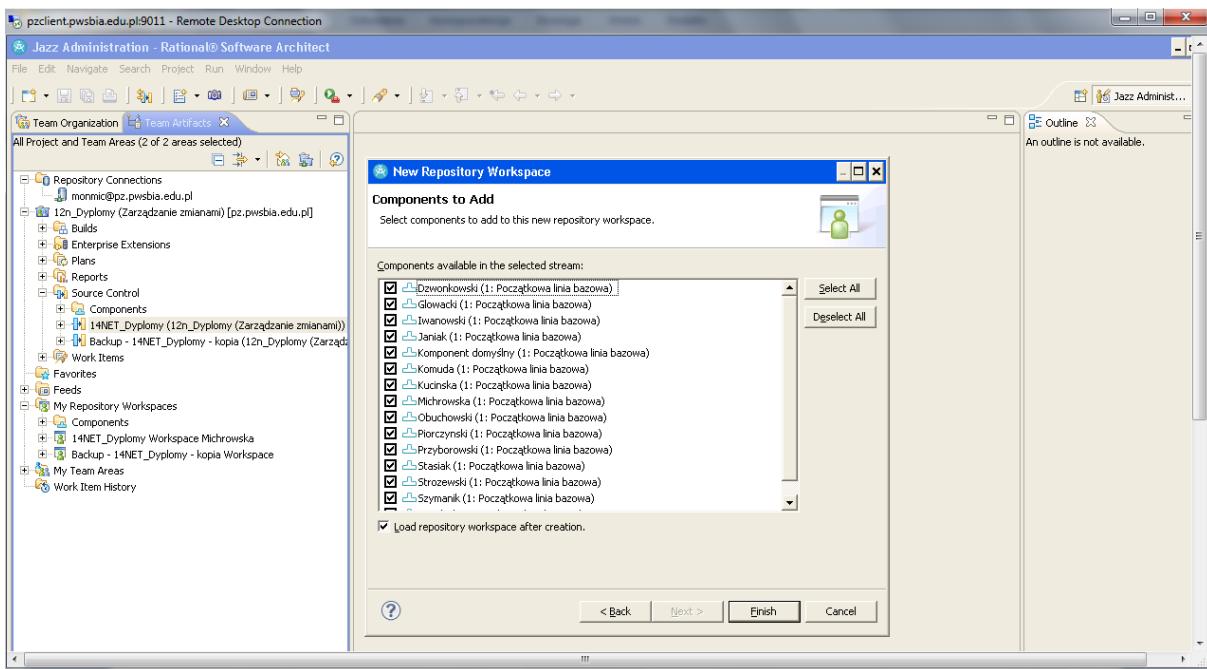
✓ Wybierz Next

✓ Na ekranie Read Acces Permission wybierz **SCOPED**.

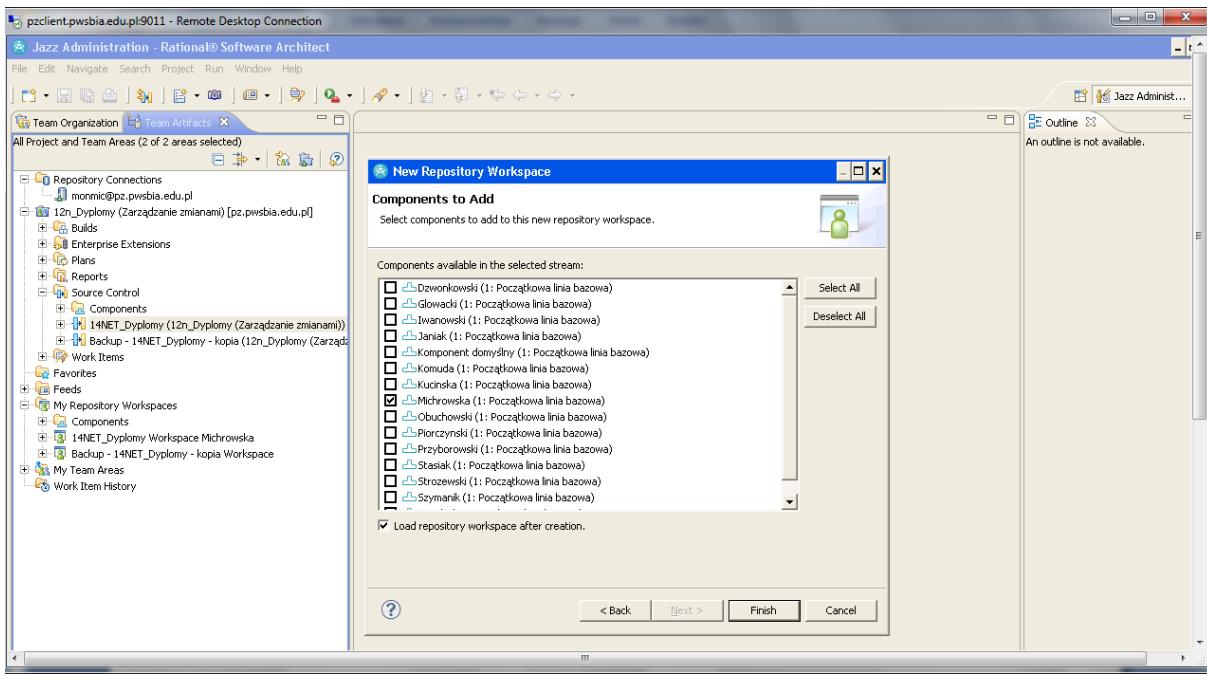
✓ Zaznacz nazwę projektu w celu uaktywnienia buttonów.



- ✓ Wybierz Next
- ✓ Na ekranie Componens to add wybierz Deselect All.

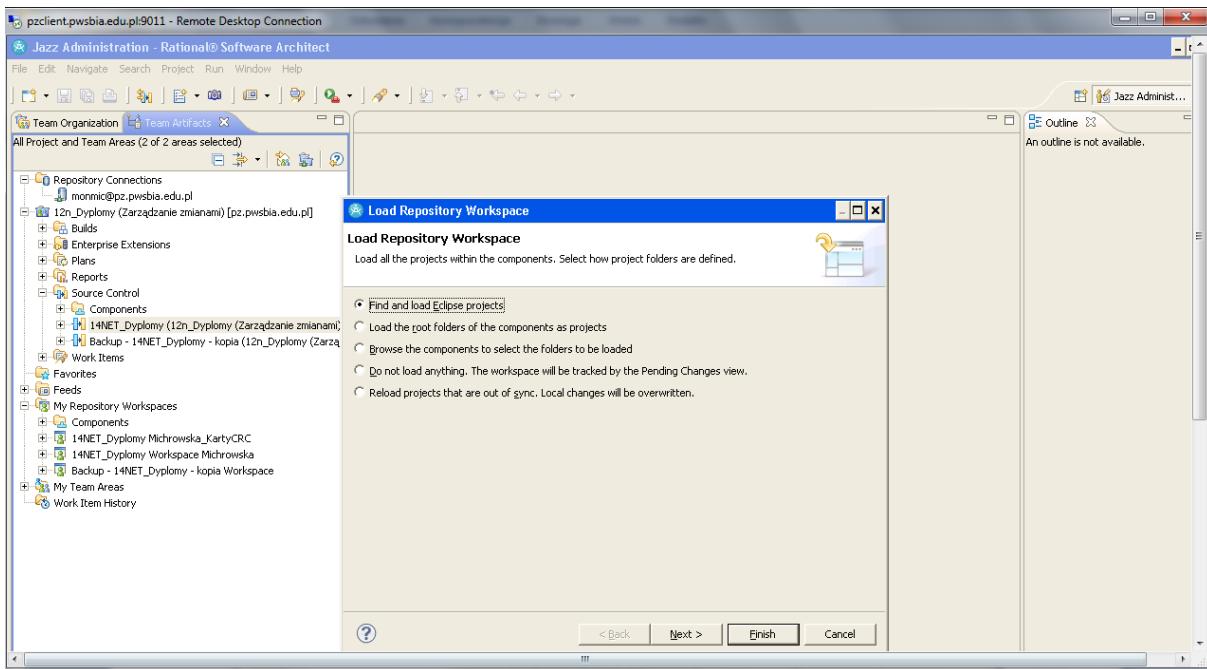


- ✓ Następnie zaznacz komponent z Twoim nazwiskiem lub komponent domyślny, jeśli praca ma się odbywać w projekcie zespołowym.

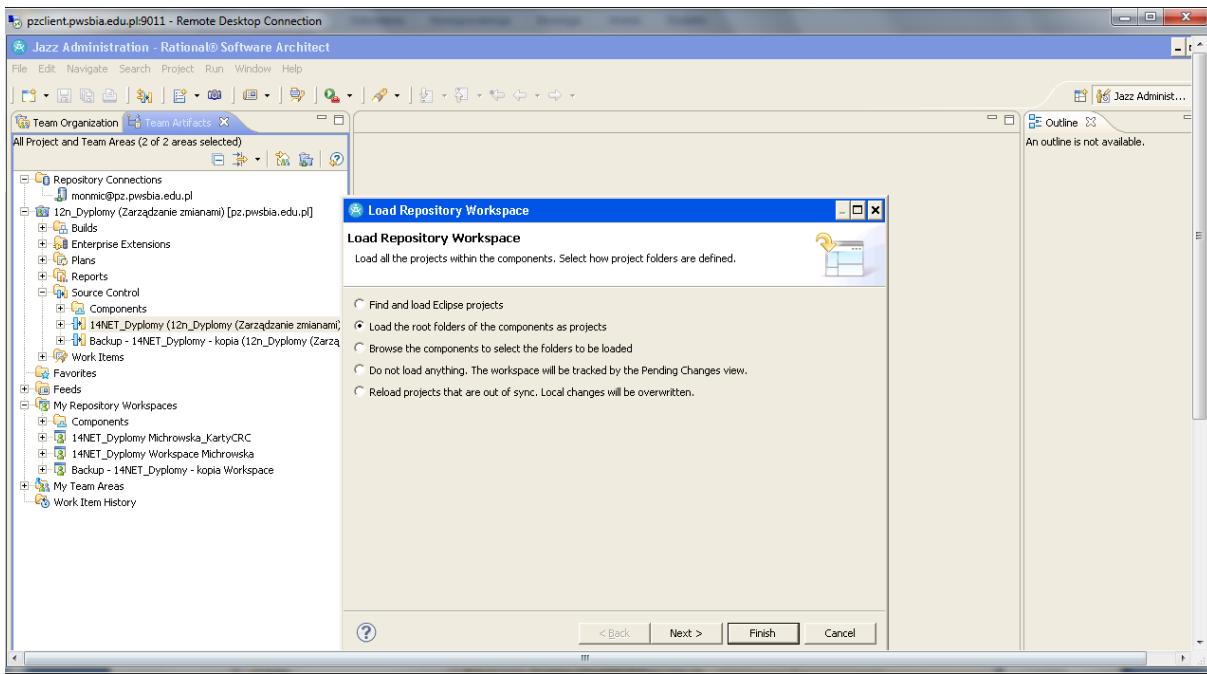


✓ Wybierz Finish.

Po utworzeniu workspacu, projekt będzie chciał się zsynchronizować z danymi na serwerze.

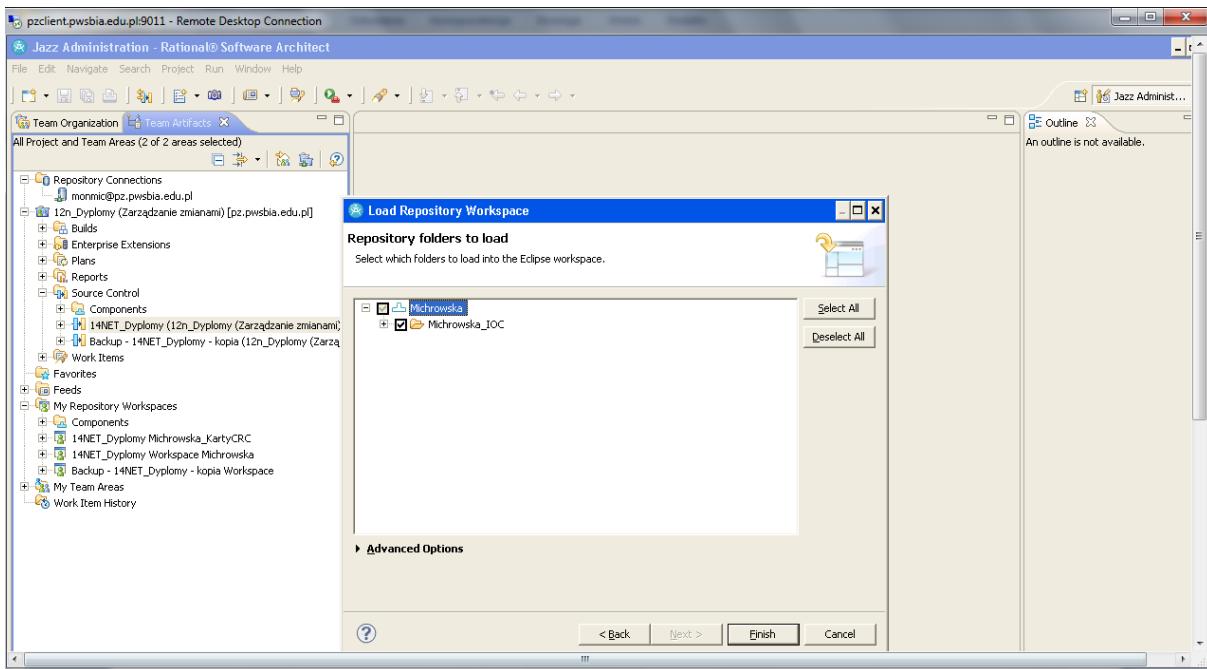


✓ W oknie Load Repository Workspace wybierz **Load the Root folders of the component as projects**.



✓ Wybierz Next

Do wyboru powinien być tylko zdefiniowany komponent z Twoim nazwiskiem .

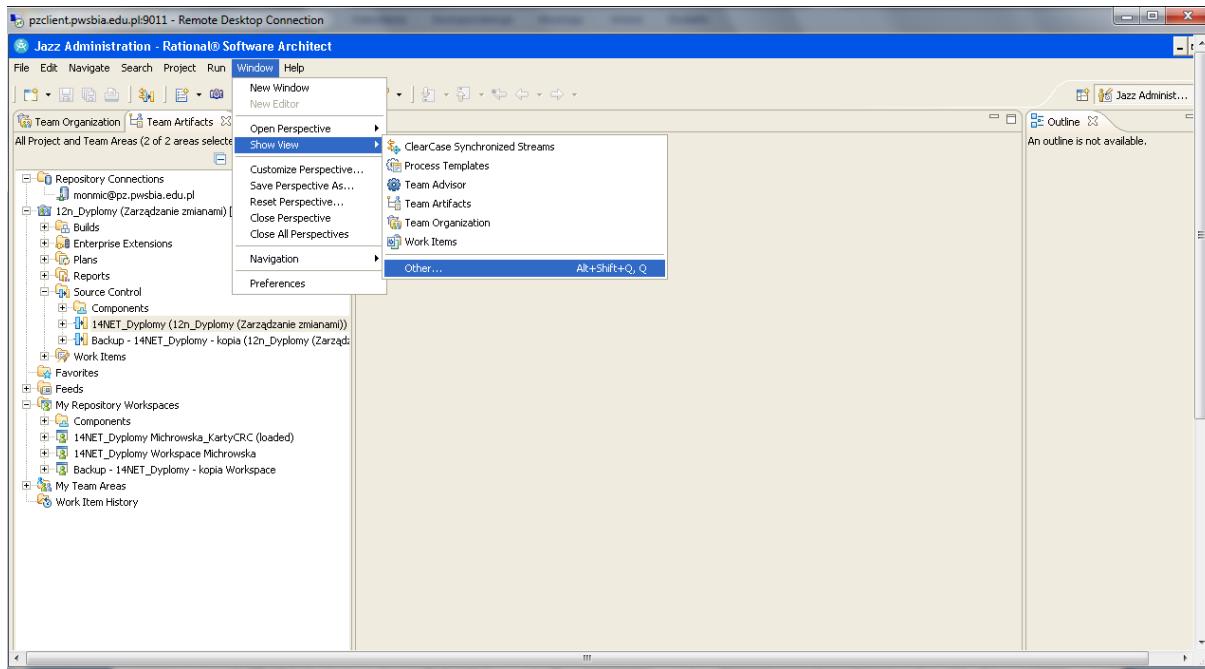


✓ Wybierz Finish.

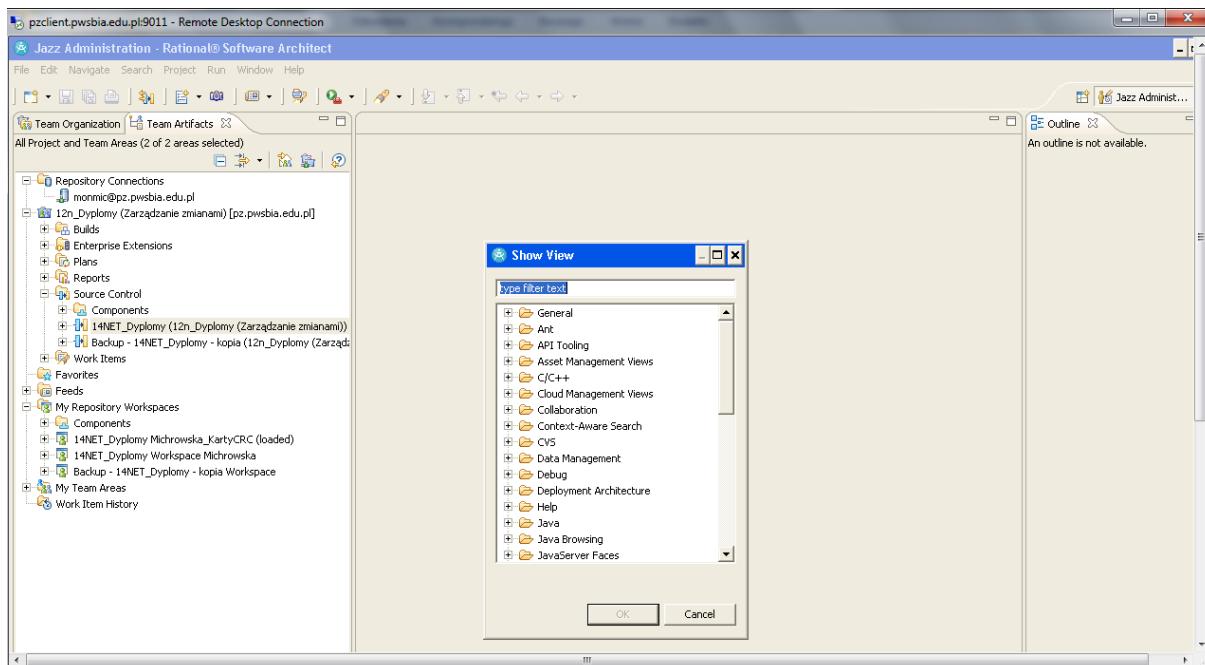
15. Pending Changes

Następnym krokiem będzie sprawdzenie, czy do wybranego projektu nie ma żadnych zmian przychodzących z serwera. W tym celu należy otworzyć odpowiedni widok **Pending Changes**.

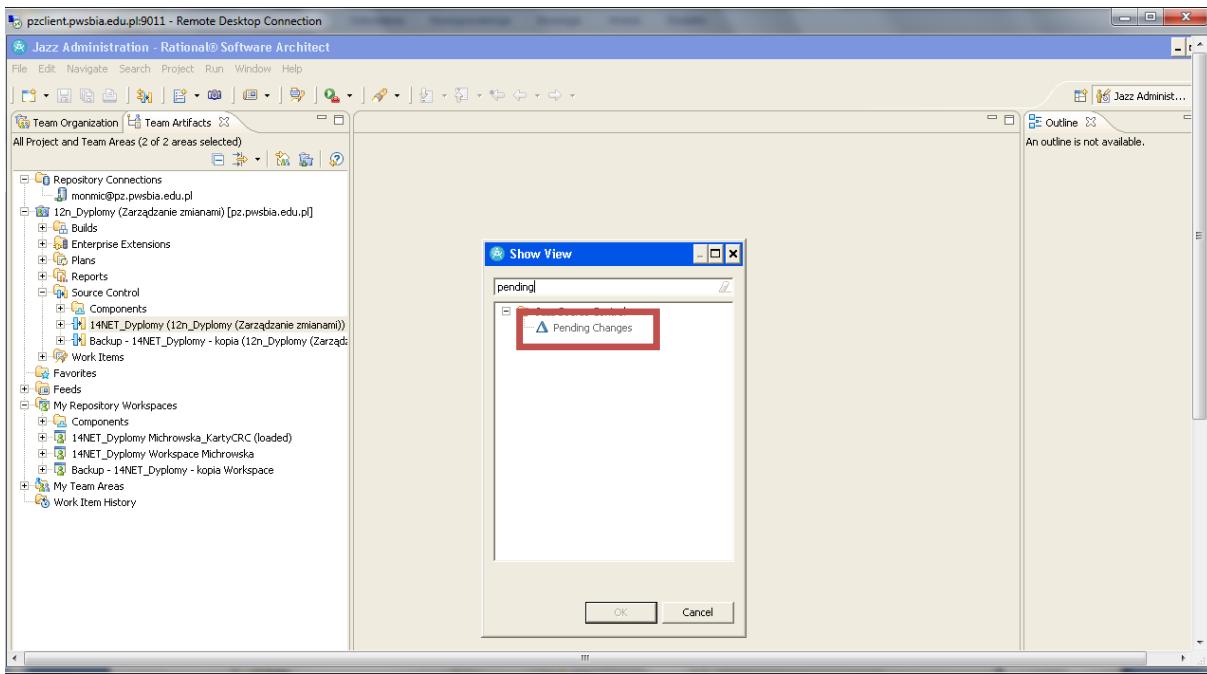
Jeśli widok nie jest domyślnie widoczny w Twojej przestrzeni można otworzyć go sięgając do zakładki **>>Window>>Show View>>Other...** (zazwyczaj nie ma go na prezentowanej liście)



- ✓ Pojawi się okno wyboru. W polu wyszukiwania wpisz "**pending**".

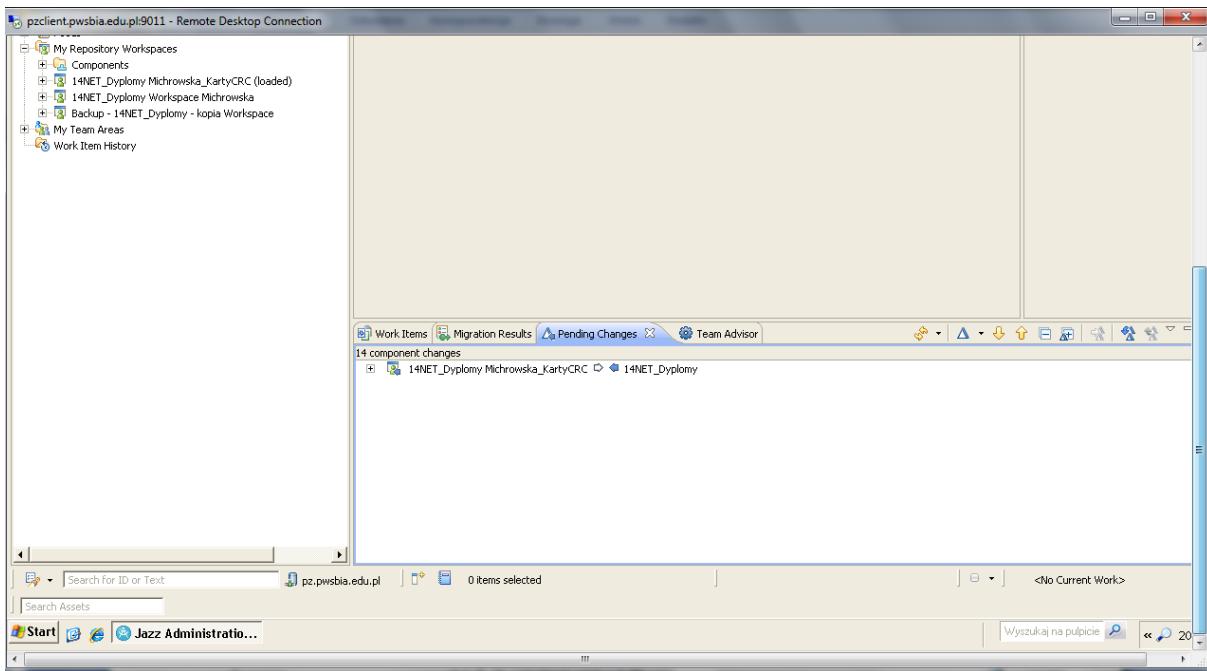


Dzięki temu zostanie wyszukany interesujący nas widok:

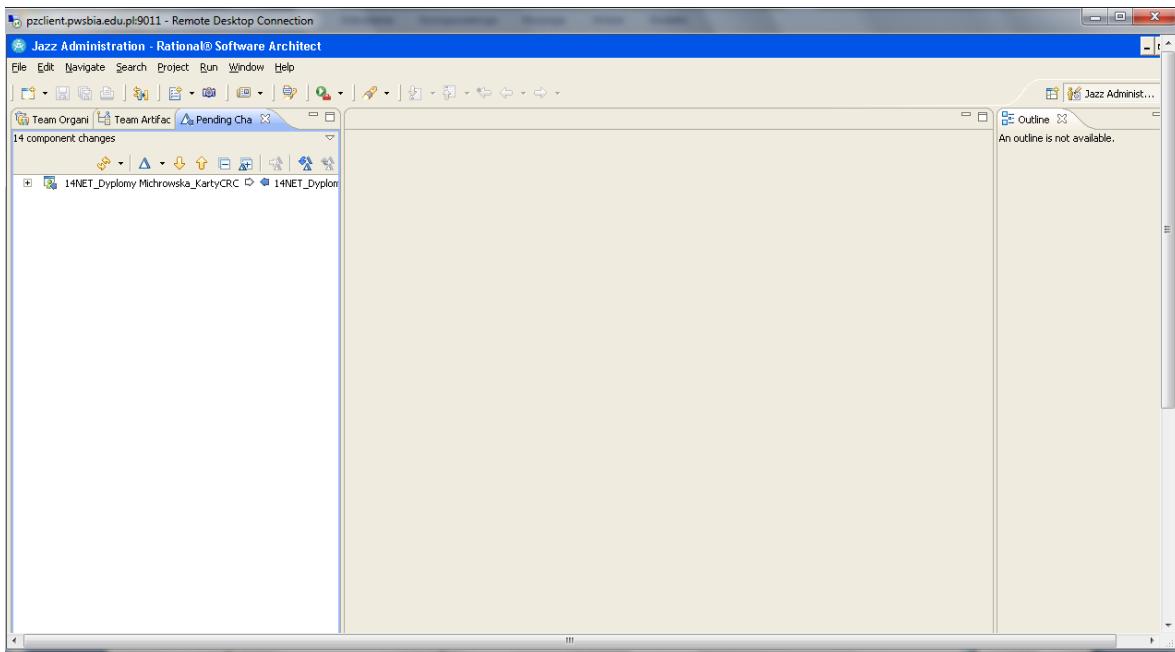


✓ Wybierz go i zatwierdź OK.

W moim przypadku okno otworzyło się w dolnej części i aplikacji.



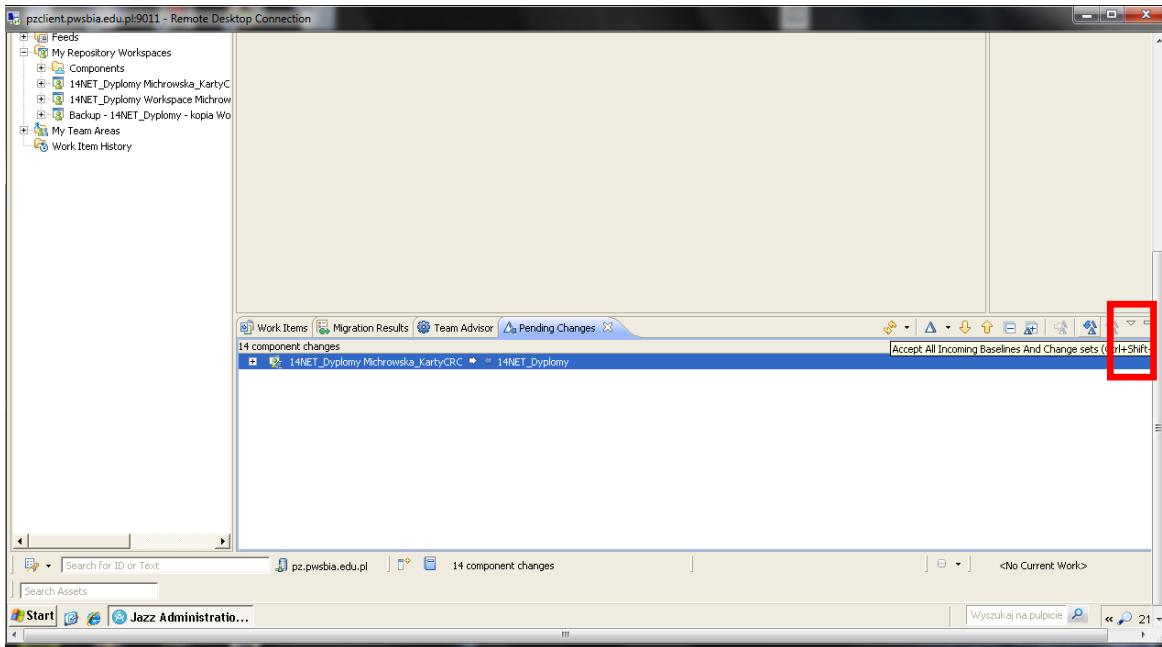
Jednak jeśli układ, który zaproponowało Ci środowisko nie jest dla Ciebie odpowiedni, wystarczy chwycić za nazwę tego widoku i przenieść je w dowolne miejsce np. : tu:



Teraz najważniejsze! W widoku Pending Changes, widać zmiany przychodzące. Bardzo ważne jest aby akceptować zmiany przychodzące, przed wysłaniem czegokolwiek do repozytorium. Jeśli zostanie pominięty ten krok, spowodujesz utratę danych w repozytorium, co jest niepowołanym działaniem.

16. Odbieranie zmian przychodzących.

Jeśli guzik zaznaczony na ekranie jest aktywny, oznacza to, że należy zaakceptować zmiany przychodzące z serwera.



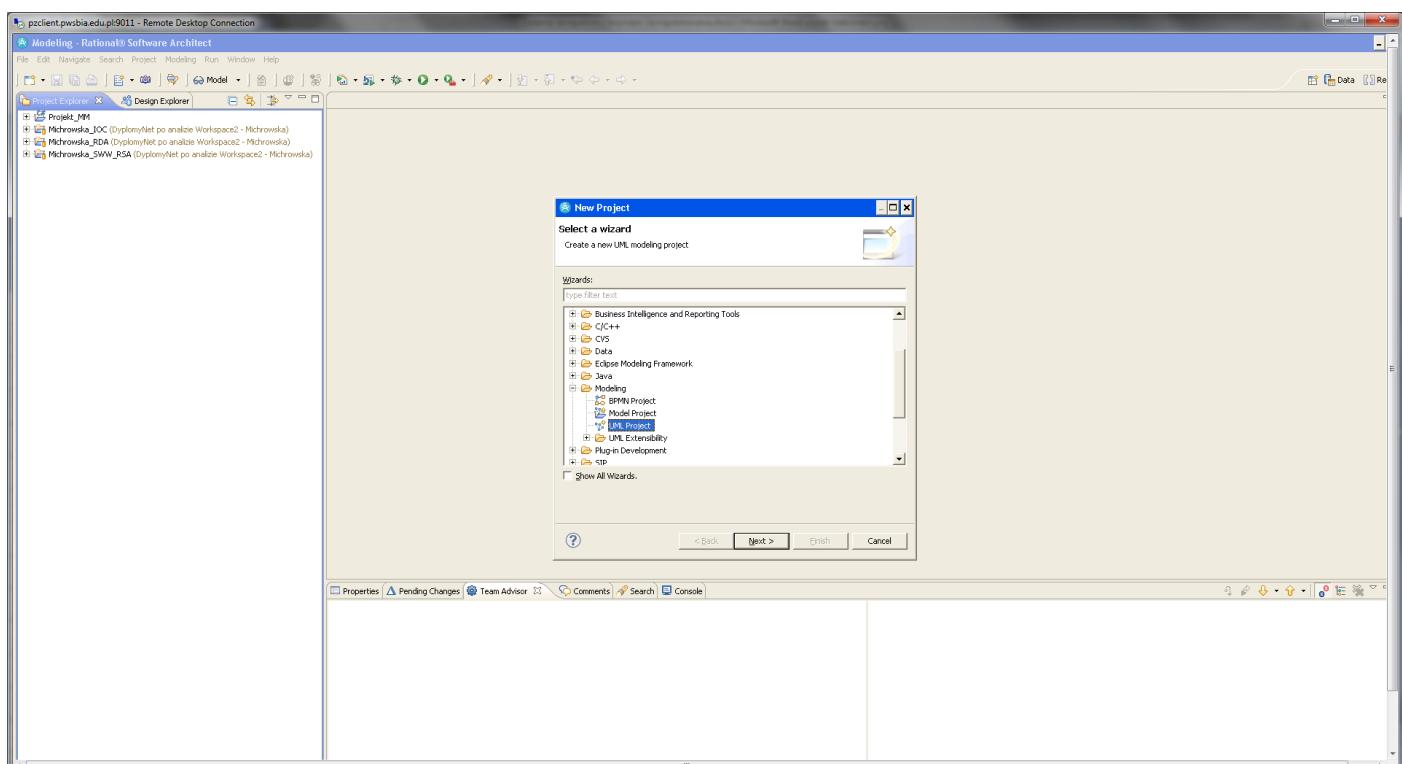
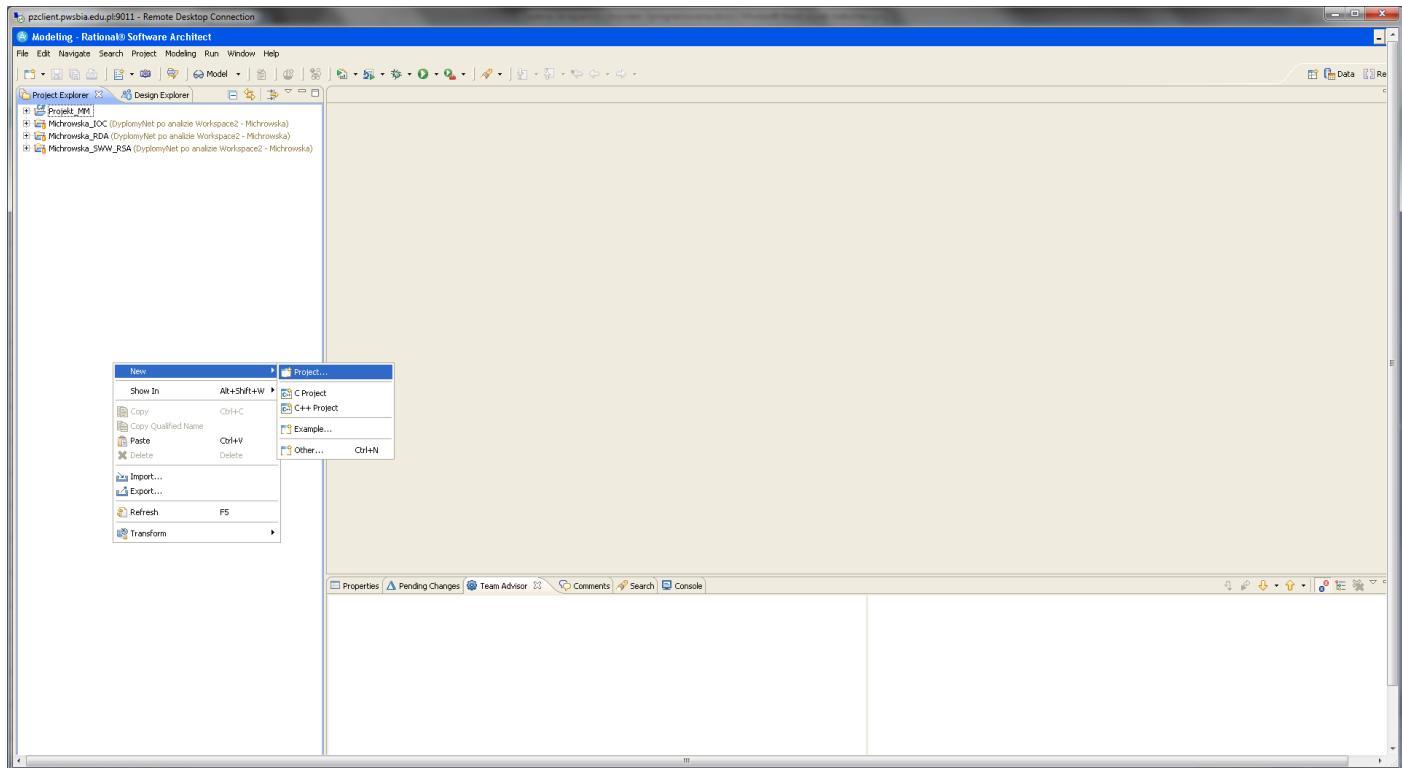
- ✓ Pobierz zmiany.

Nigdy nie rozpoczynaj pracy bez pobrania zmian przychodzących!

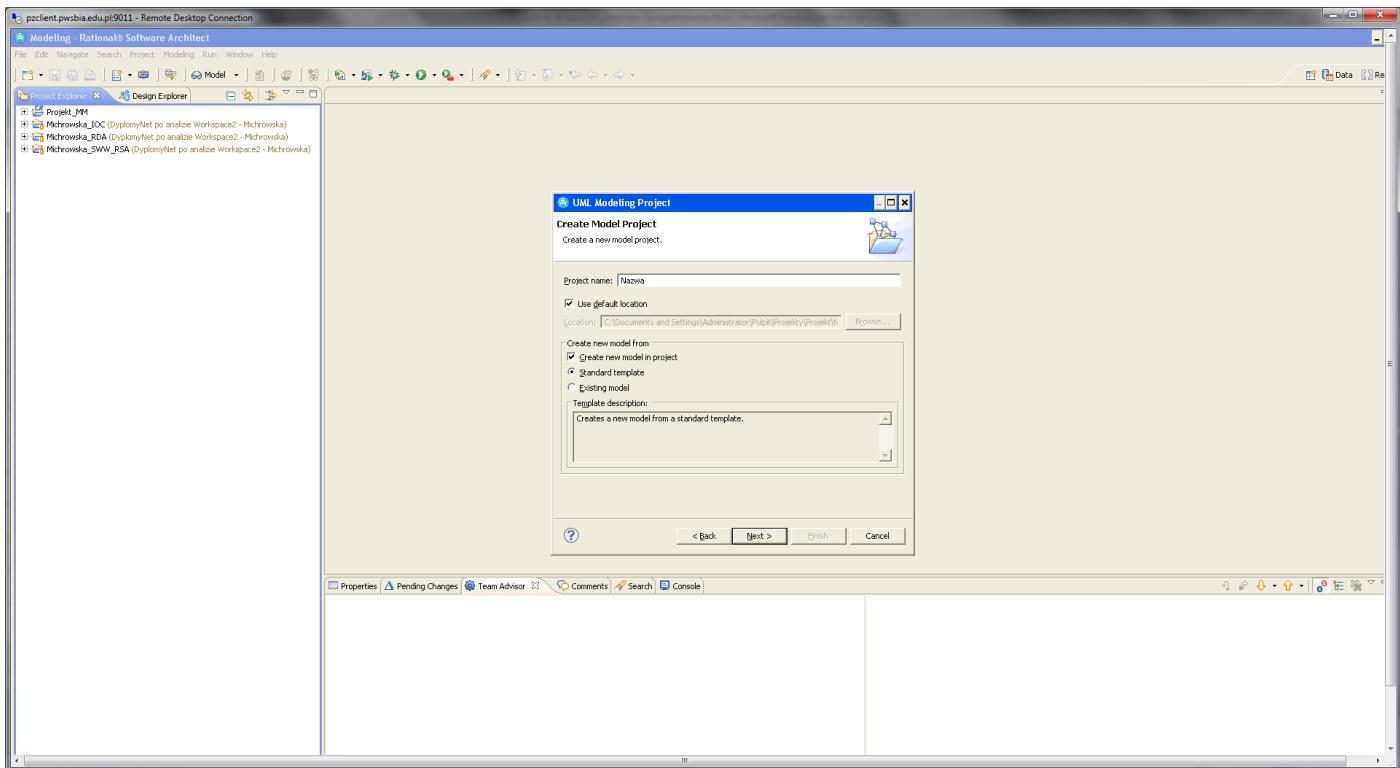
- ✓ Wybierz Yes.

17. Utwórz model przypadków użycia poprzez asocjację repozytoriów

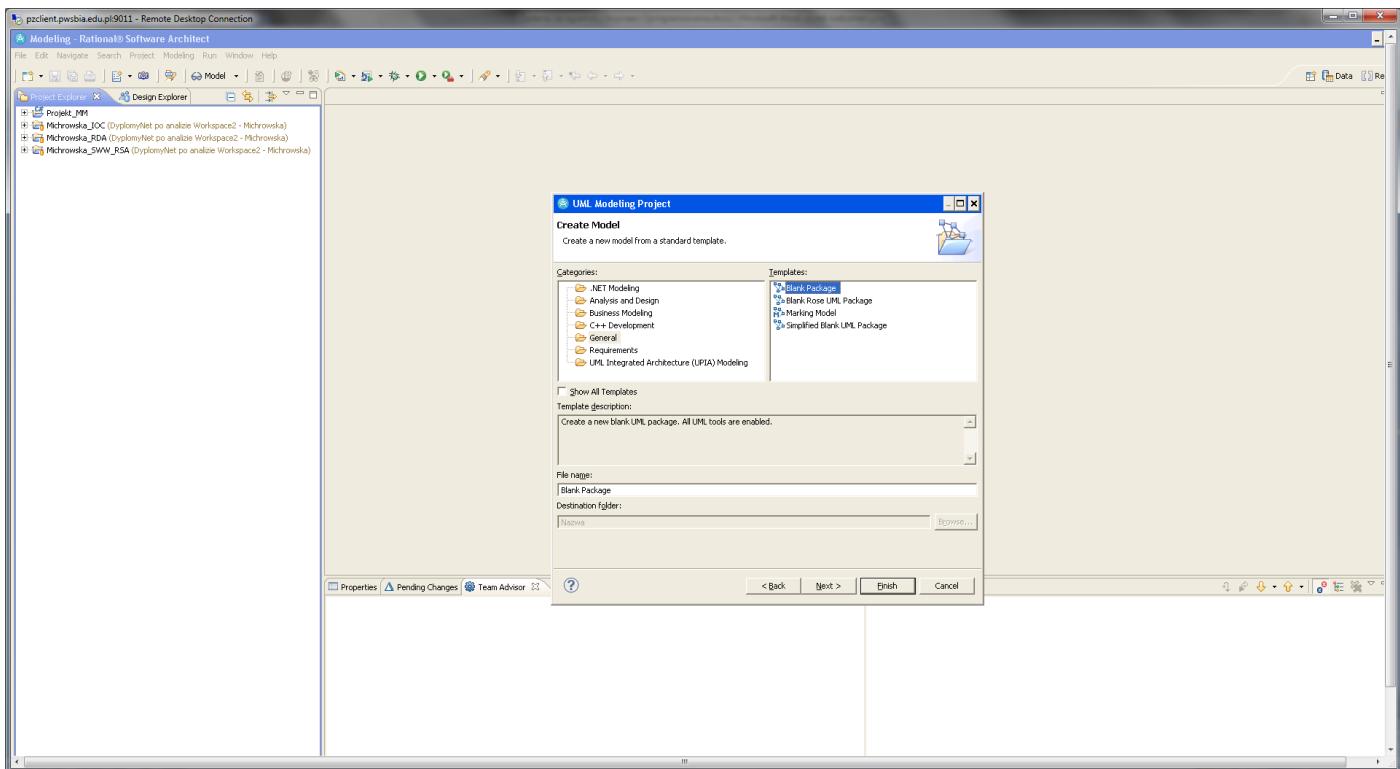
- ✓ Otwórz perspektywę modeling. (Jeśli jeszcze nie wiesz jak to zrobić wróć po rozdziału 11.)
- ✓ W oknie Project Explorer jest Twój komponent.
- ✓ Jeśli w celach egzaminacyjnych należy utworzyć Nowy model, wybierz prawym na oknie eksplorer i utwórz nowy model na bazie UML Project



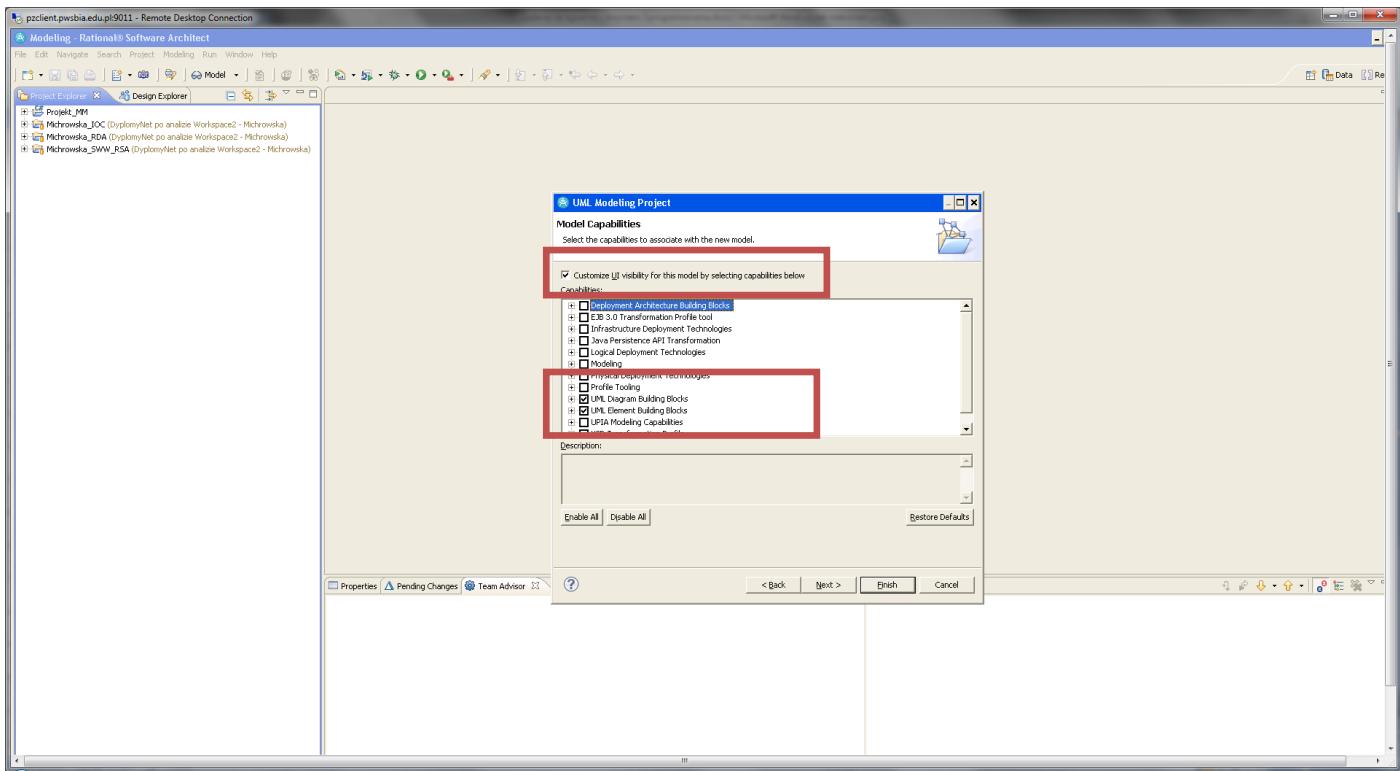
- ✓ Nadaj nazwę



- ✓ Next
- ✓ wybierz General i Blank Package na dole ekranu wpisz **File name** jako nazwę tworzonego projektu.



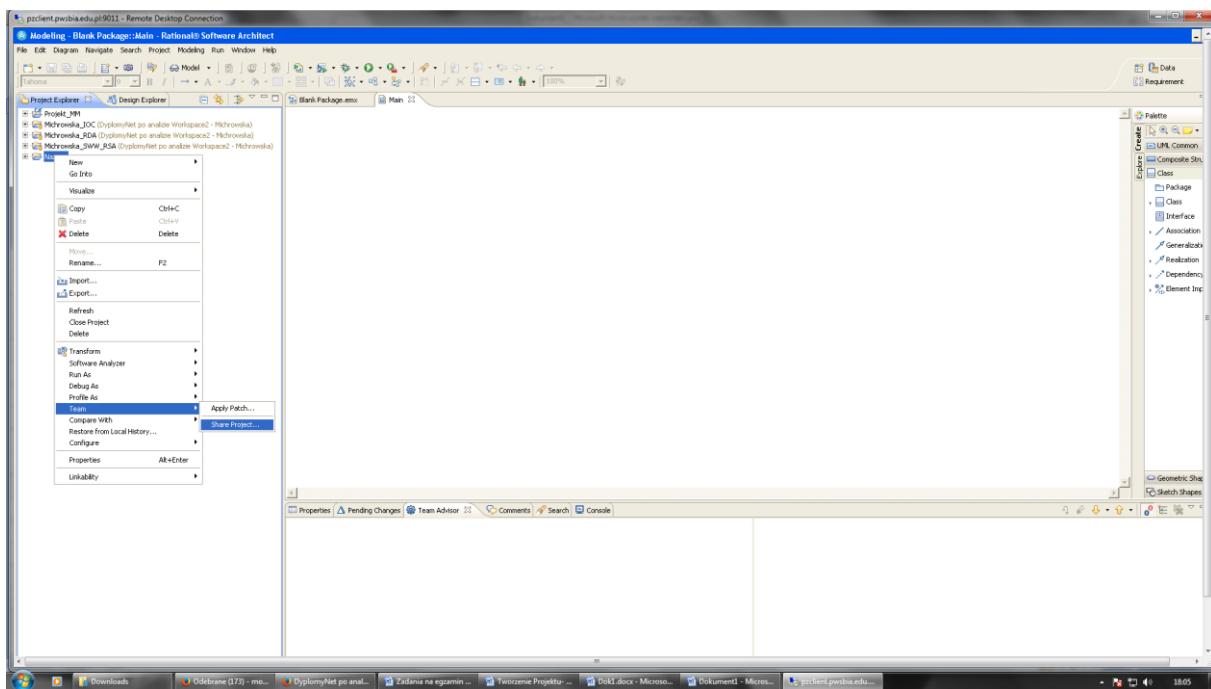
- ✓ Next
- ✓ Zaznacz ptaszka w oknie Customize...
- ✓ Wybierz obydwa UML jako komponenty do modelowania



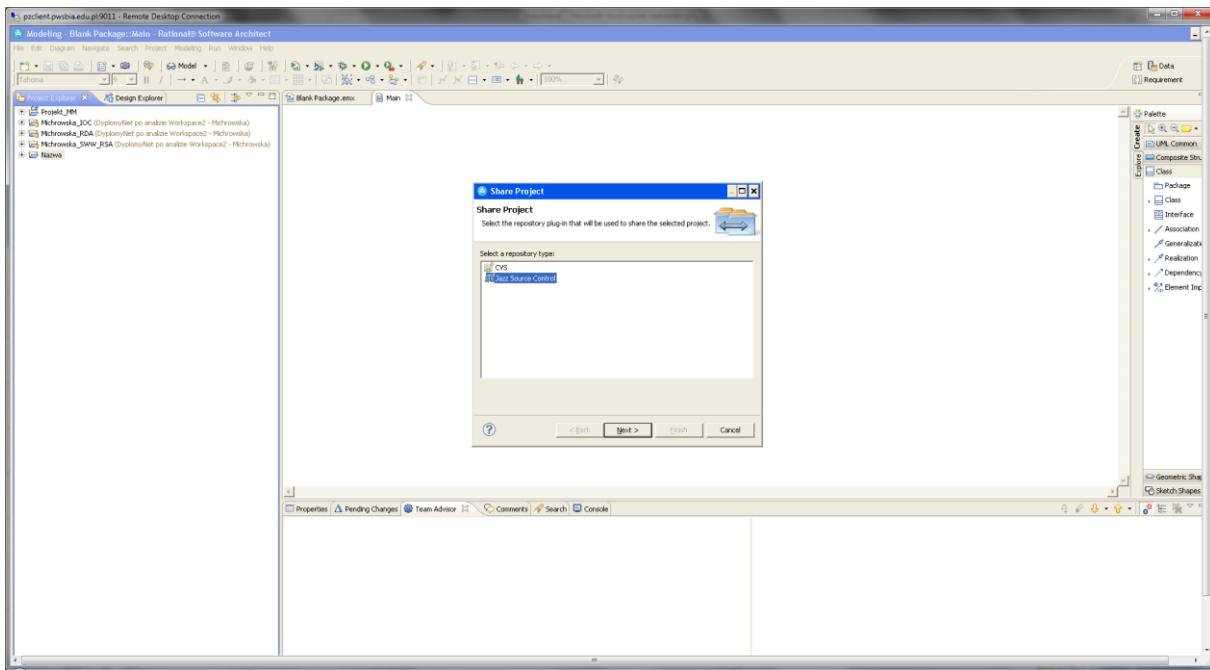
- ✓ Next, Next i Finish

Żeby wysłać zmiany z tego folderu musisz go współdzielić w tym celu:

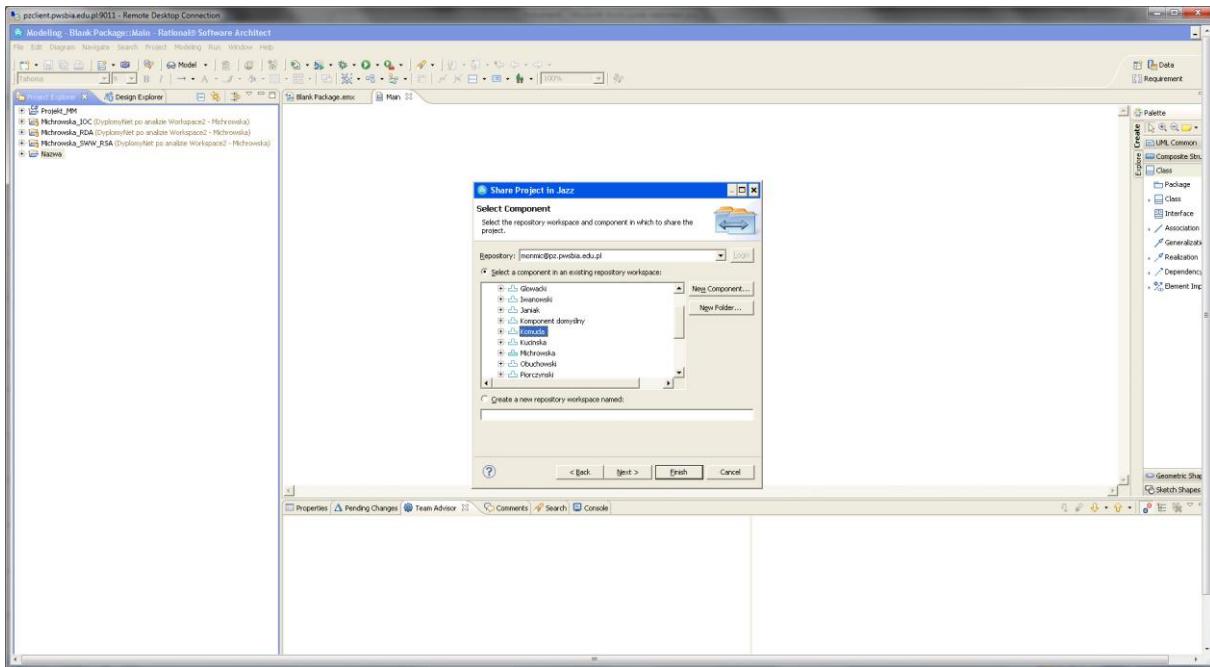
- ✓ Wybierz TEAM>>> Share Project...



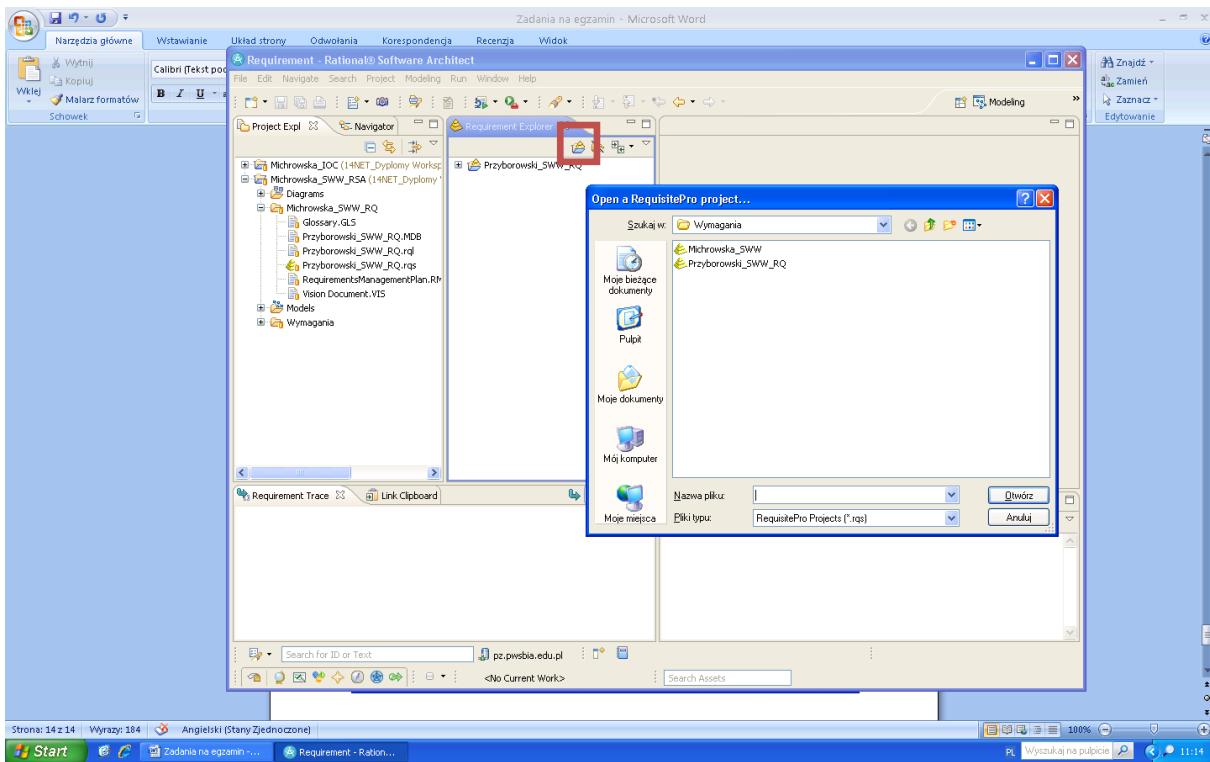
- ✓ Potem wybierz JAZZ



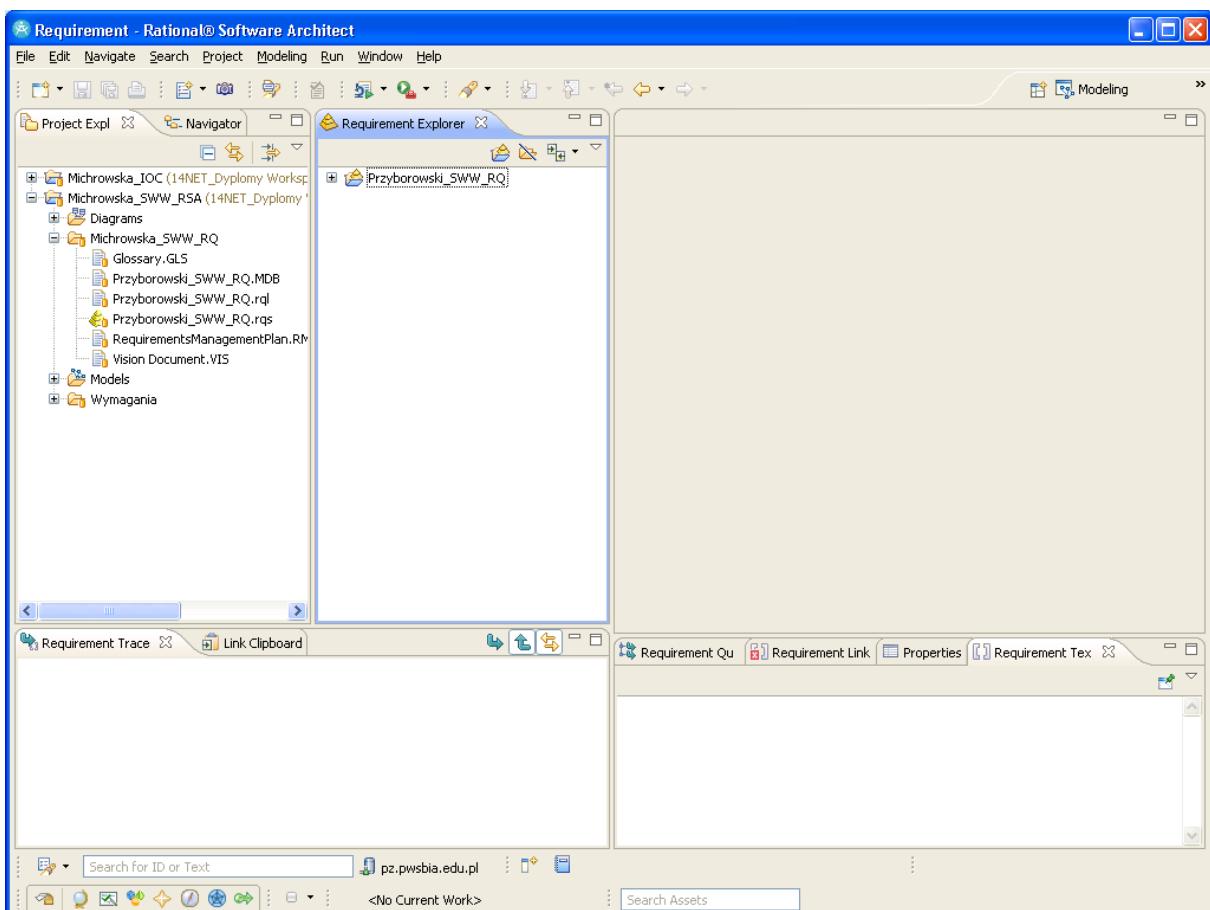
- ✓ wybierz swój komponent



- ✓ Finish i od tej pory powinno wysyłać się na serwer bo jest to zasób współdzielony
- ✓ Utwórz w projekcie w folderze Models pakiet o nazwie Wymagania lub innej wymienionej w zadaniu.
- ✓ Zmień perspektywę na REQUIREMENT. Otworzy się dodatkowe okno do połączenia z repozytorium.
- ✓ Otwórz RequisitePro project, który utworzyłeś dla wizualnego modelowania wymagań, w tym celu wybierz zaznaczoną na poniższym ekranie ikonę.

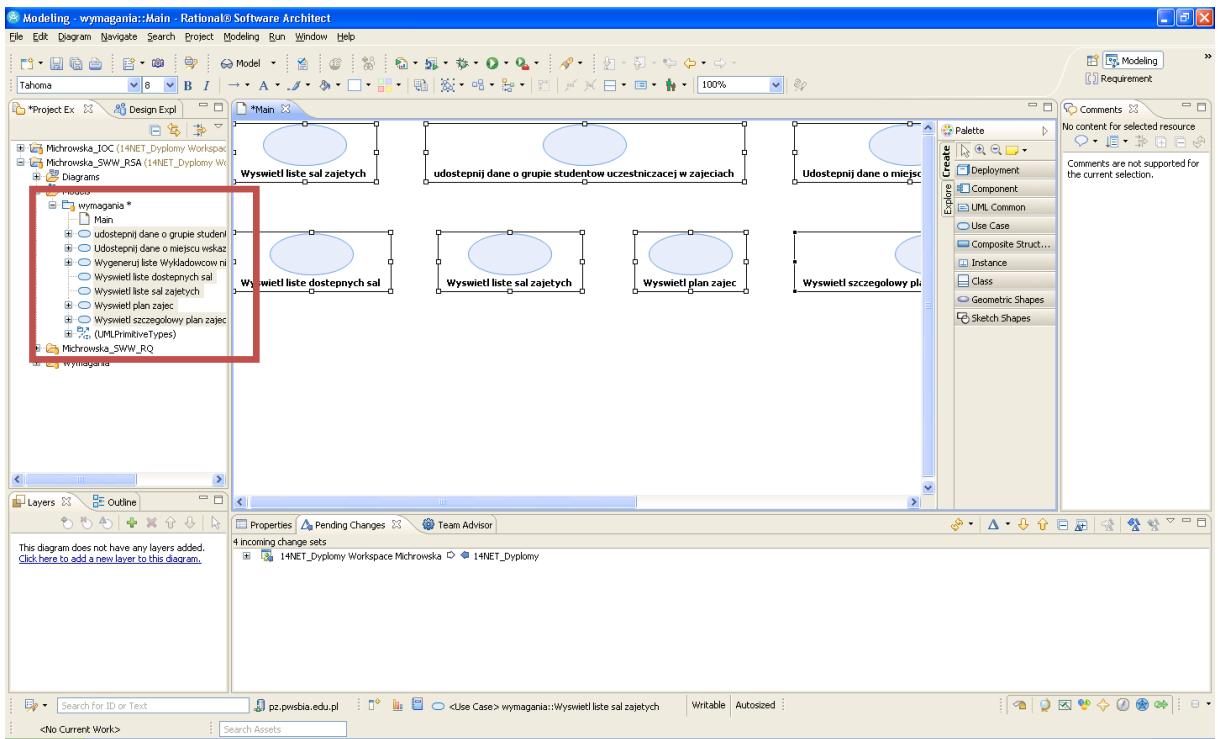


- ✓ Połącz się z wybranym projektem, wybierając admin jako użytkownika.
- ✓ Rozwiń folder z projektem i odszukaj swoje UC.



- ✓ Przenieś UC do folderu Wymagania przeciągając je z Repozytorium Rational Requisite PRO.

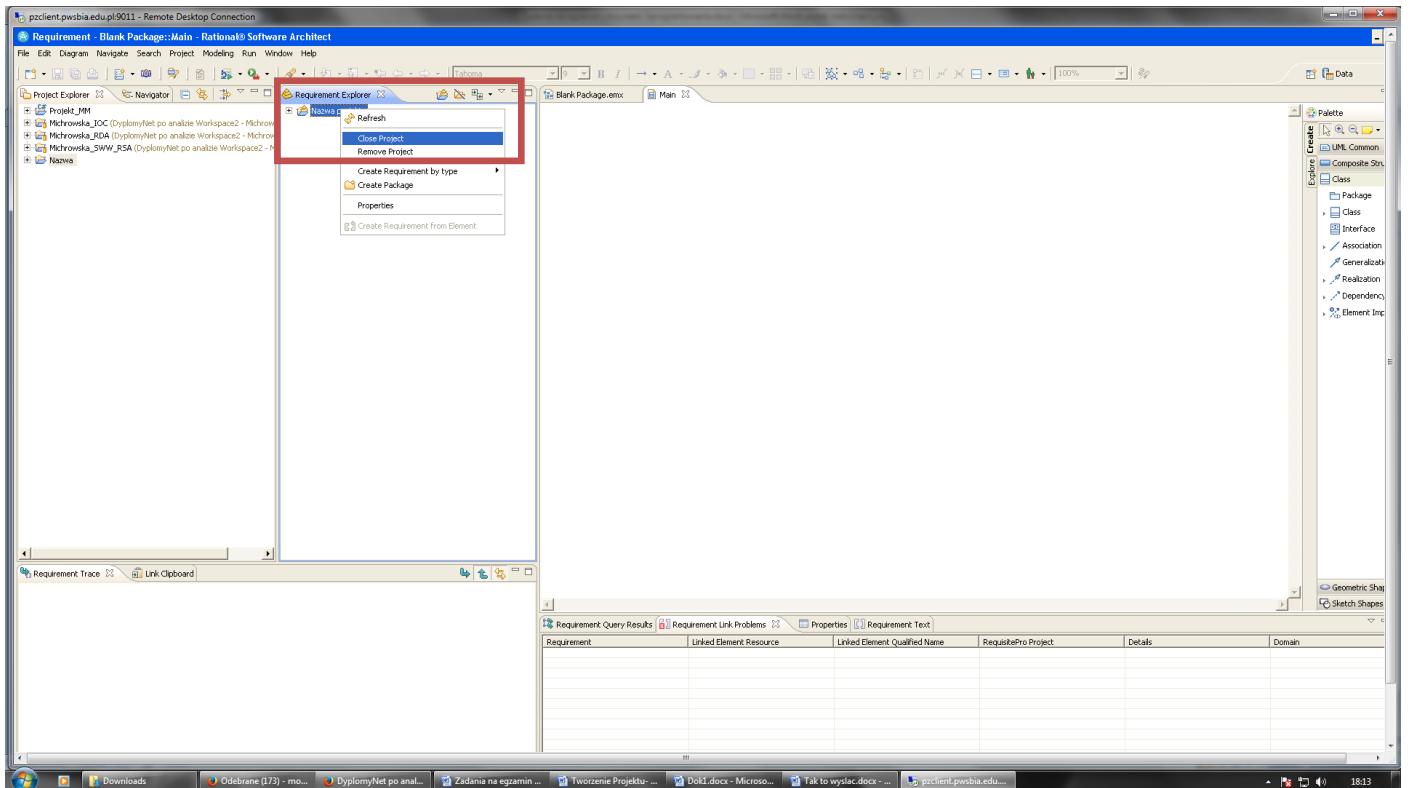
W oknie twojego projektu widzisz teraz utworzone przez Ciebie UC w Rational Requisite PRO.



✓ Została zakończona asocjacja repozytorium.

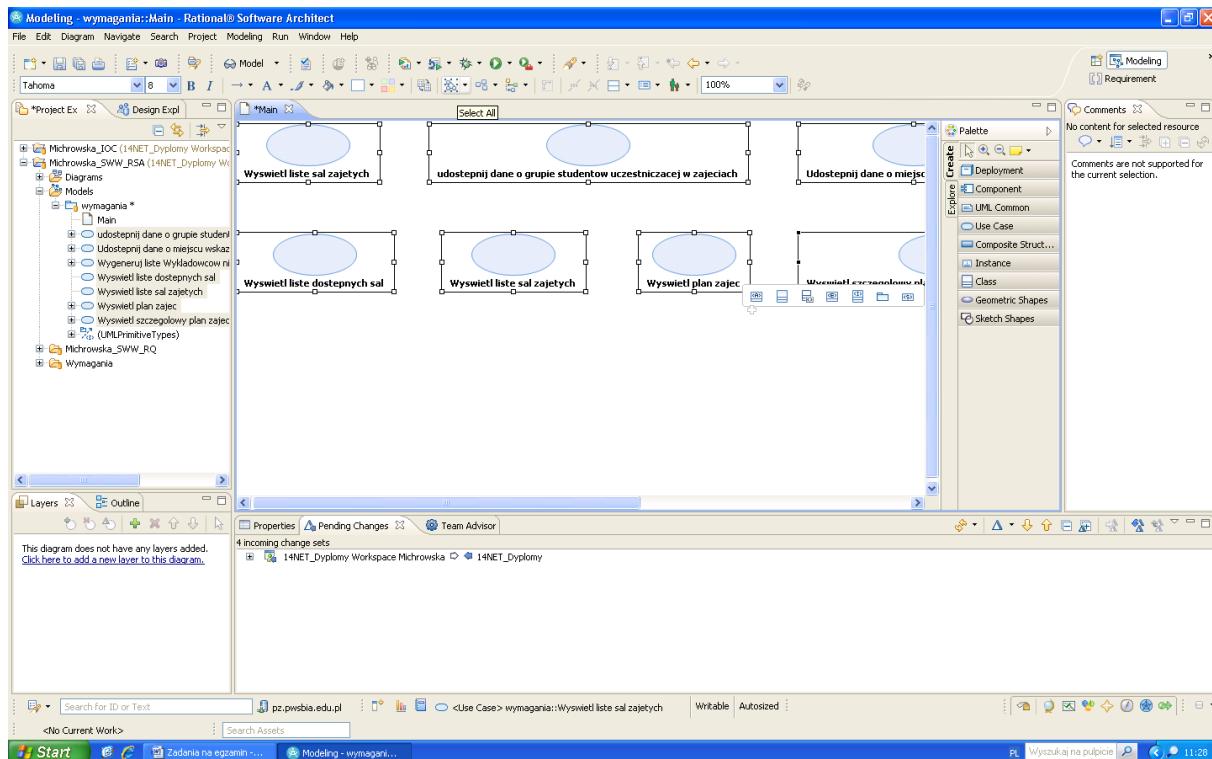
Przed wysłaniem zmian na serwer należy rozłączyć się z repozytorium Rational Requisite PRO, w tym celu

wybierz opcję Close Project na połączaniu z RRQ.



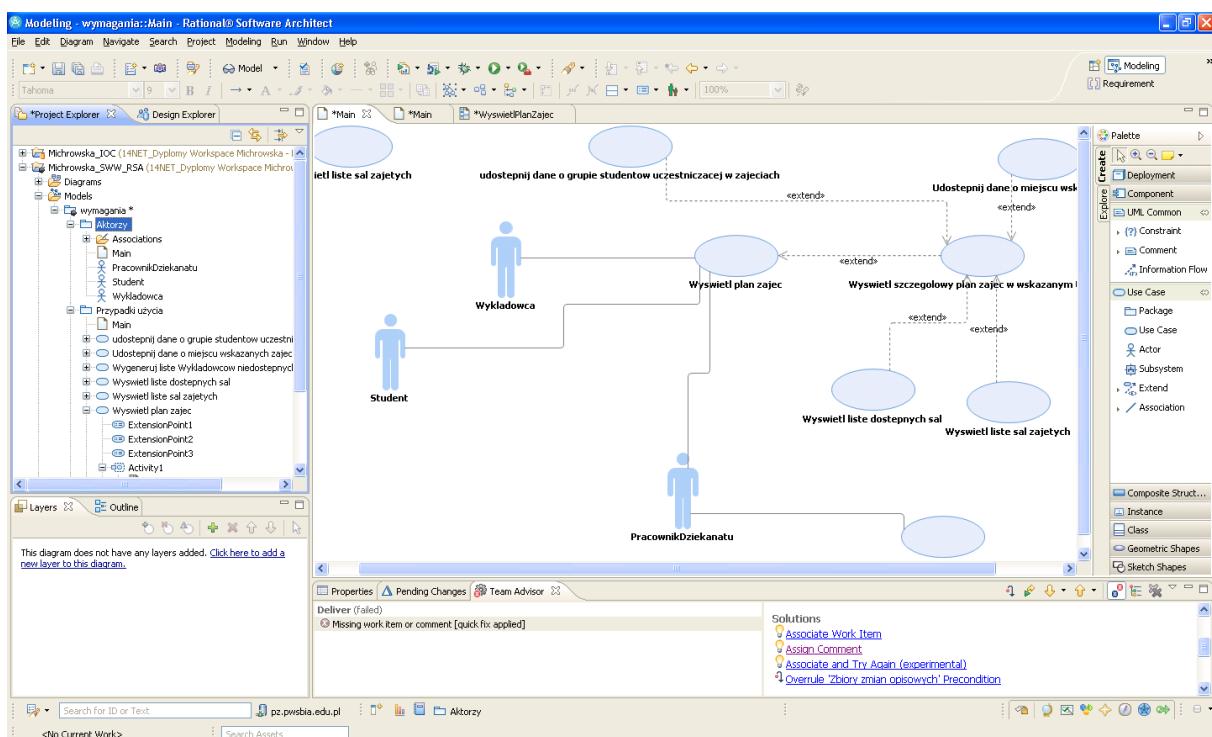
18. Utwórz model przypadków użycia

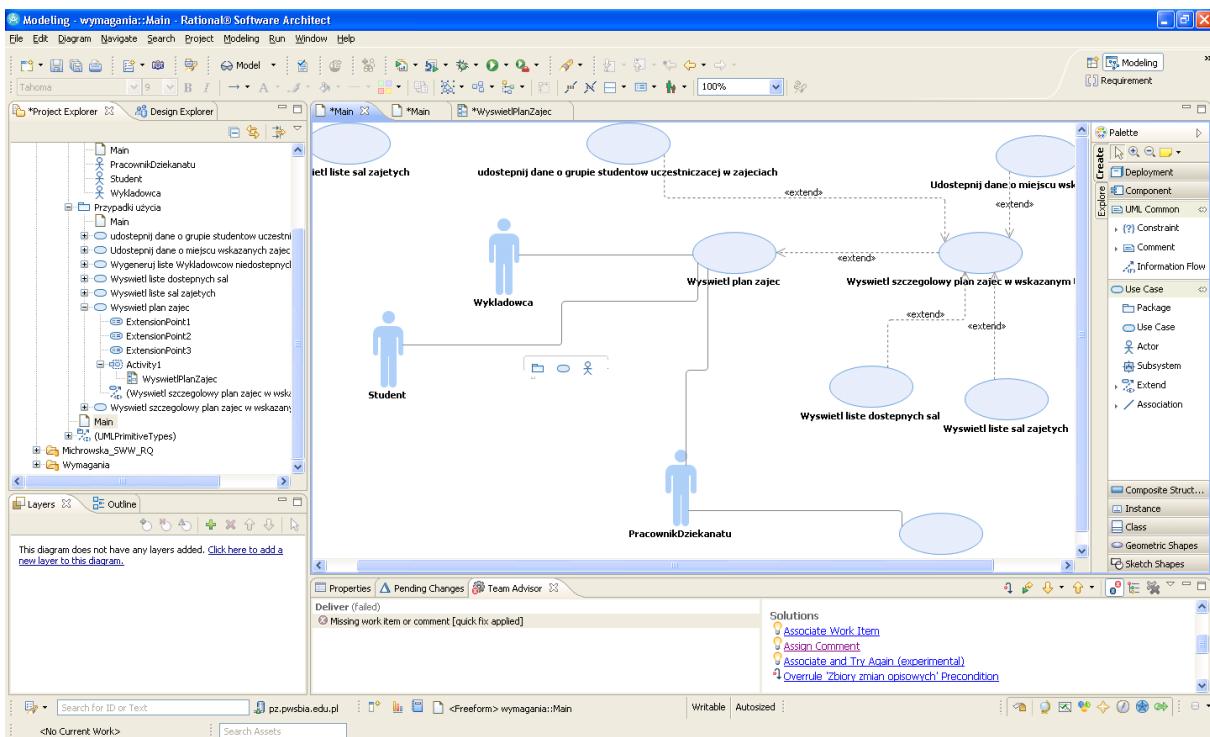
- ✓ Utwórz diagram klas i nadaj mu nazwę Model Przypadków Użycia
- ✓ Zaznacz wszystkie UC w Twoim projekcie i przeciagnij je na diagram



19. specyfikacja aktorów

- ✓ Dodaj pakiet o nazwie Aktorzy
- ✓ Utwórz tam aktorów Twojego systemu poprzez Add UML>> Actor
- ✓ Dodaj aktorów do poprzedniego diagramu
- ✓ Twórz asocje, zgodnie z zależnościami przypadków użycia. Pamiętaj o include i extend.

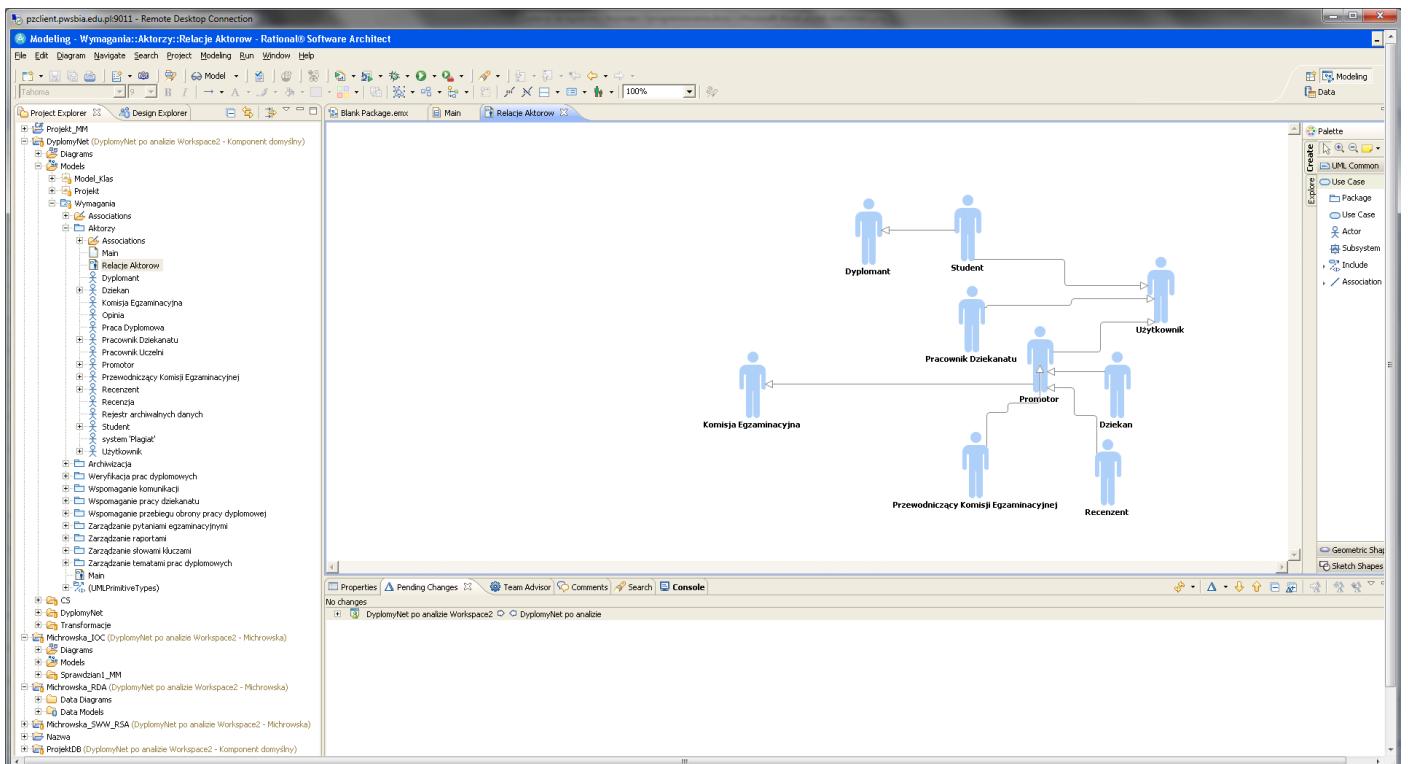




20. Model powiązań aktorów

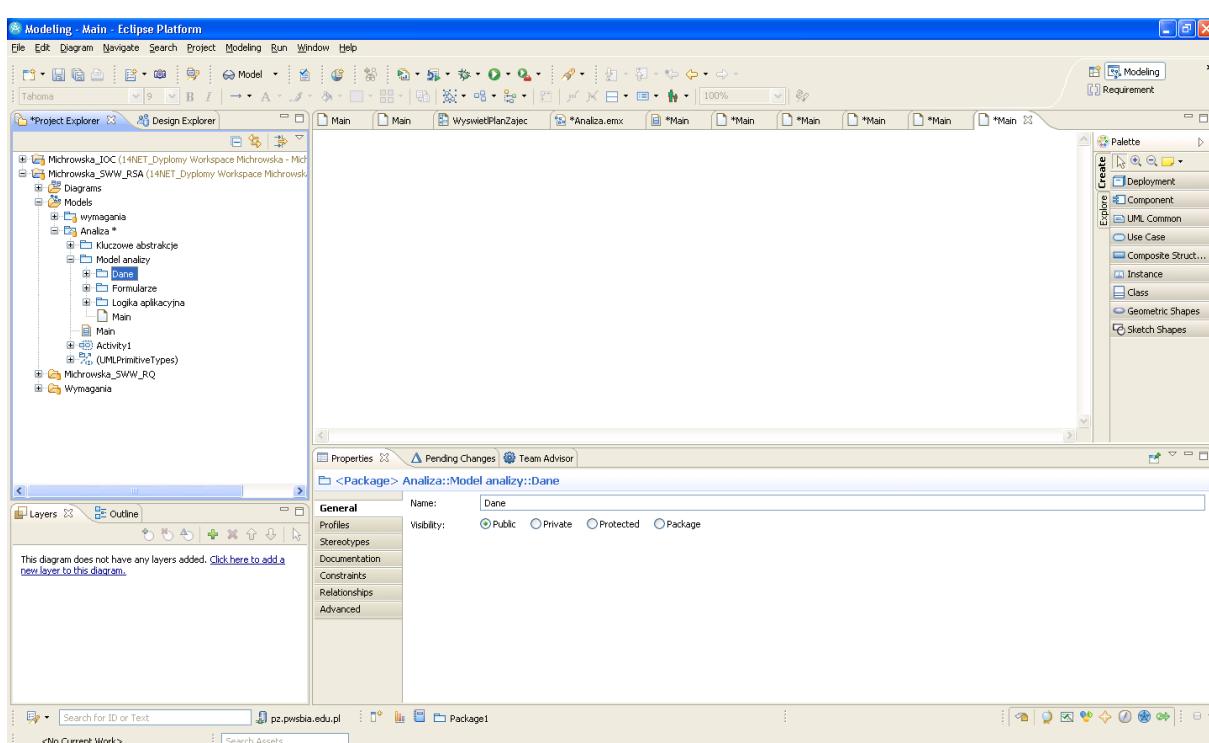
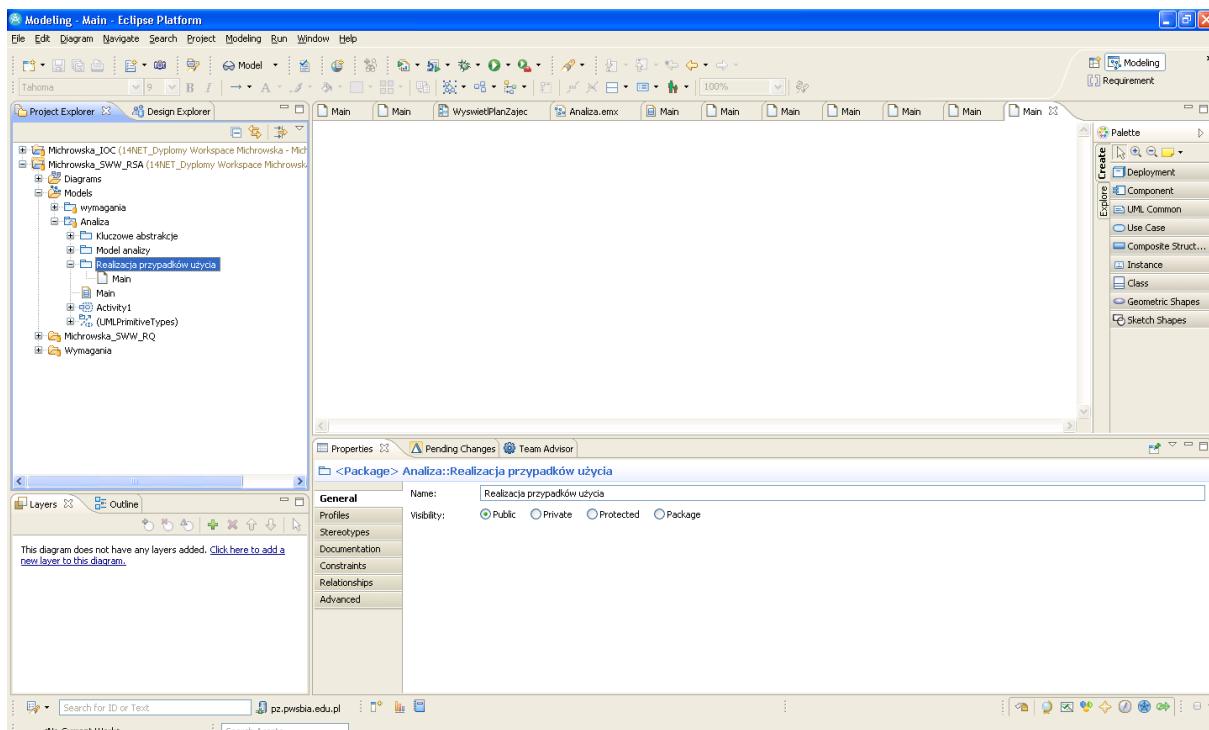
Jako dodatkowy element można zrealizować model powiązań aktorów ich generalizacji lub asocjacji jeśli taka byłaby potrzeba z punktu widzenia biznesowego.

- ✓ Utwórz diagram klas UML, nadaj mu nazwę model powiązań aktorów
- ✓ Dodaj do diagramu wszystkich aktorów i twórz relacje



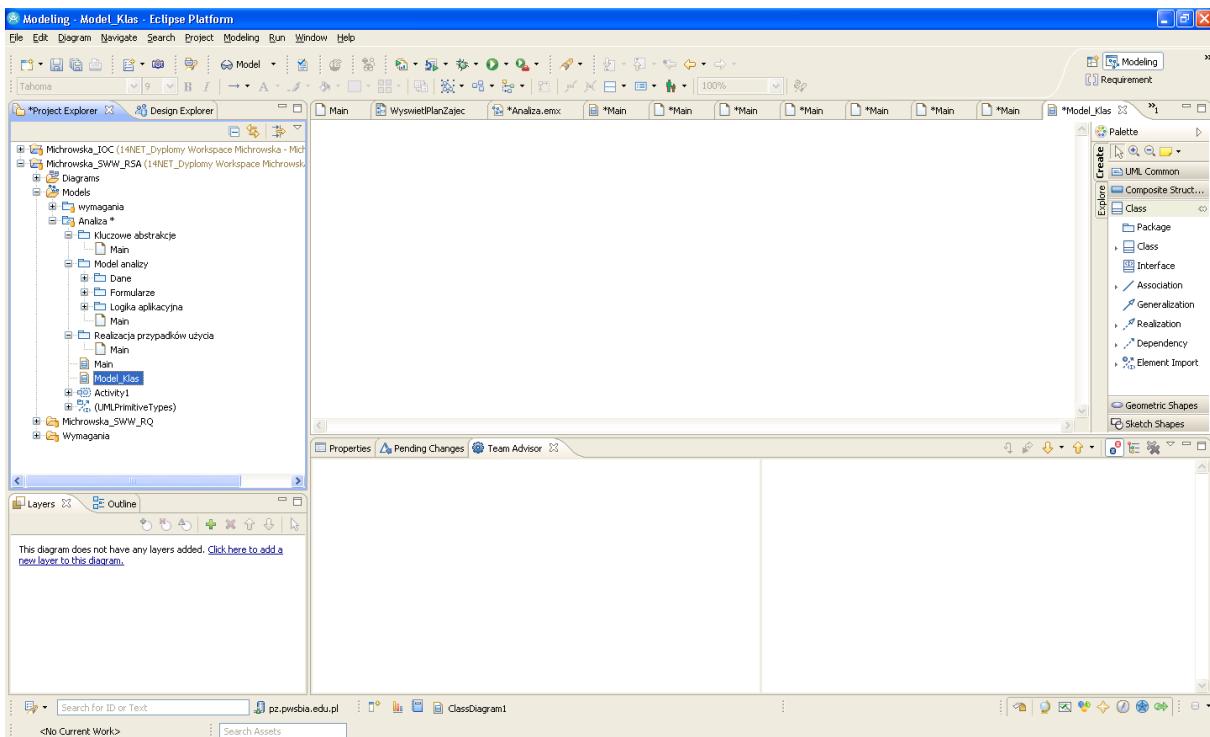
21. model analizy systemu (jako element repozytorium RSA)

- ✓ Twórz pakiety zgodnie z poniższymi nazwami i hierarchią:
 - Utwórz Pakiet Analiza (na bazie szablonu analiza, Blank Package)
 - Utwórz pakiet Kluczowe Abstrakcje
 - Utwórz pakiet Model Analizy
 - Utwórz pakiet Dane
 - Utwórz pakiet Formularze
 - Utwórz pakiet Logika aplikacyjna
 - Utwórz pakiet Realizacja przypadków użycia
 - Utwórz pakiet Kooperacje



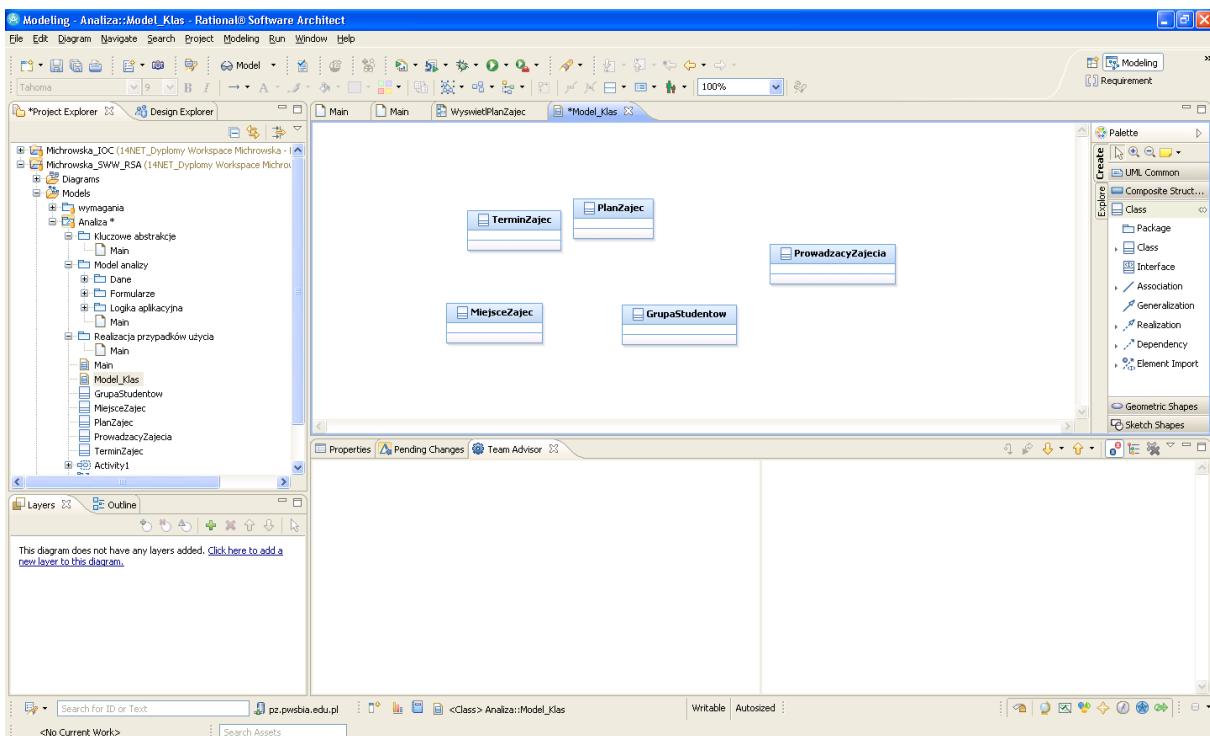
22. Utwórz model Klas

- ✓ W folderze **Kluczowe Abstrakcje** utwórz diagram klas
- ✓ Dodaj klasy Twojego systemu. Nazwy ustal samodzielnie, na bazie kluczowych abstrakcji systemu (Modeluj to tak, jakbyś patrzył na to z perspektywy kart CRC- ustal główne obiekty biorące udział w procesie)

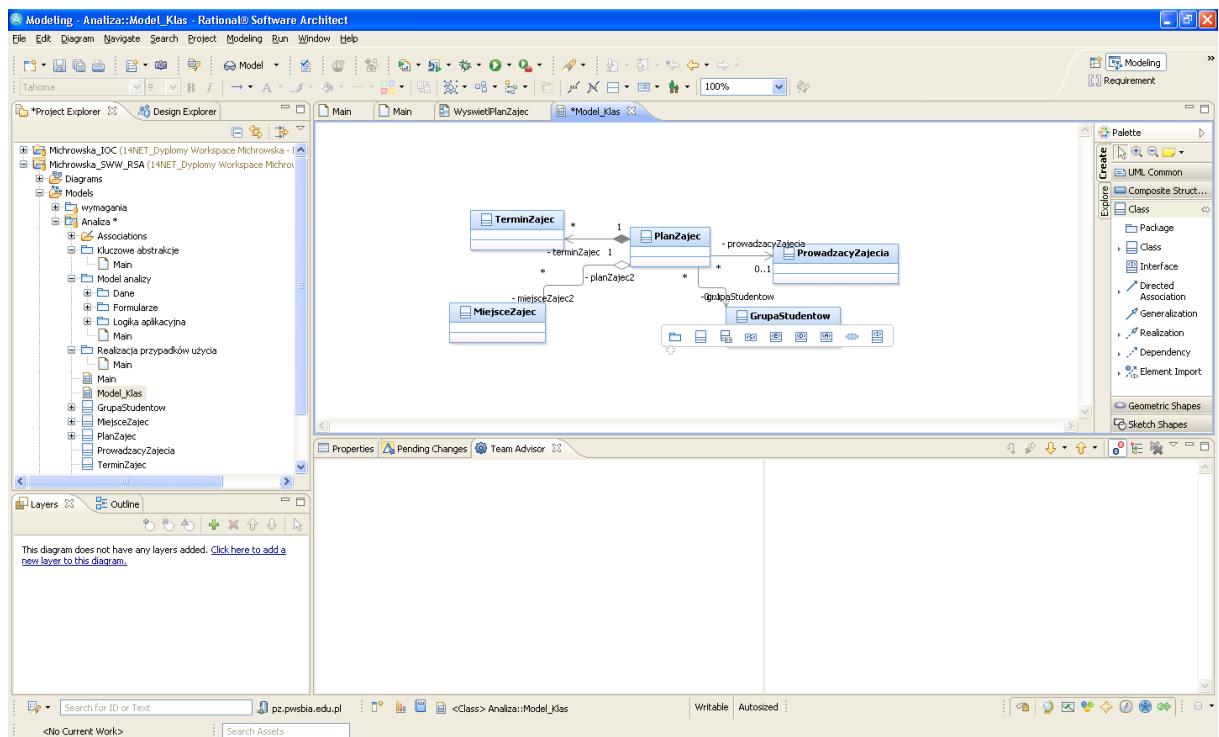


Przykładowe Klas:

Plan Zajęć, Termin Zajęć, Miejsce Zajęć, Grupa Studentów, Prowadzący zajęcia

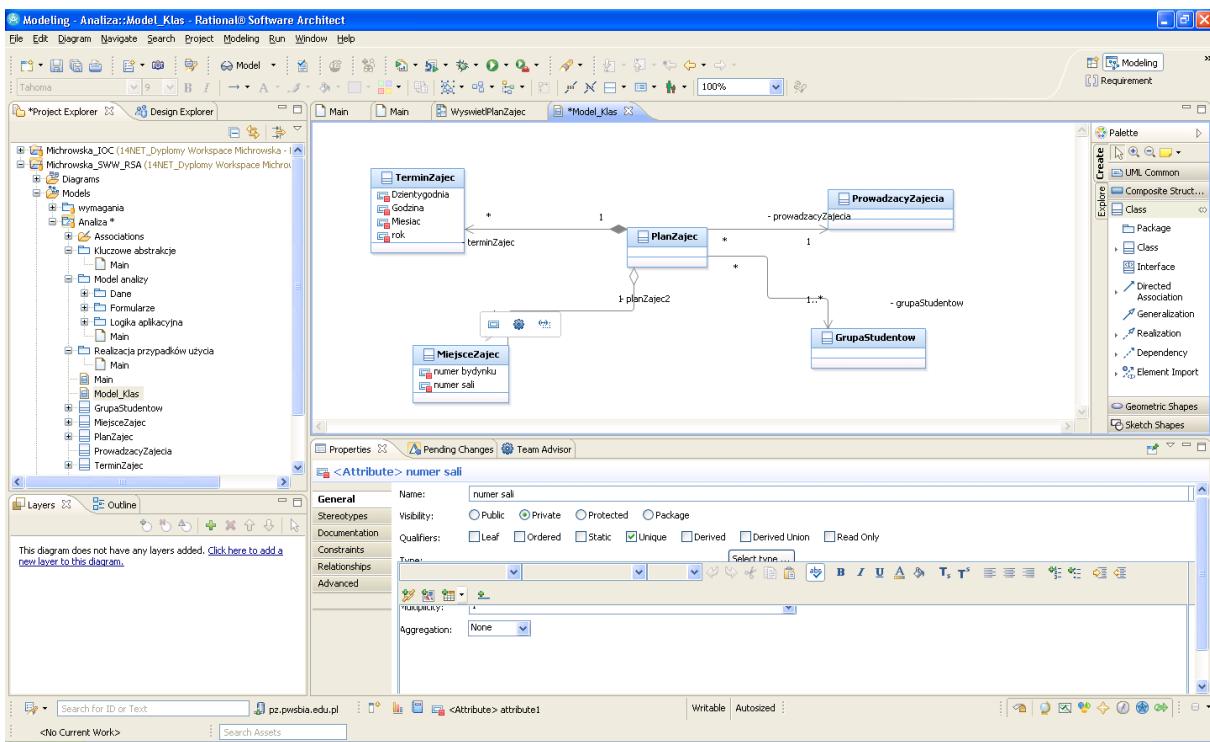


- ✓ Twórz powiązania między zdefiniowanymi klasami.



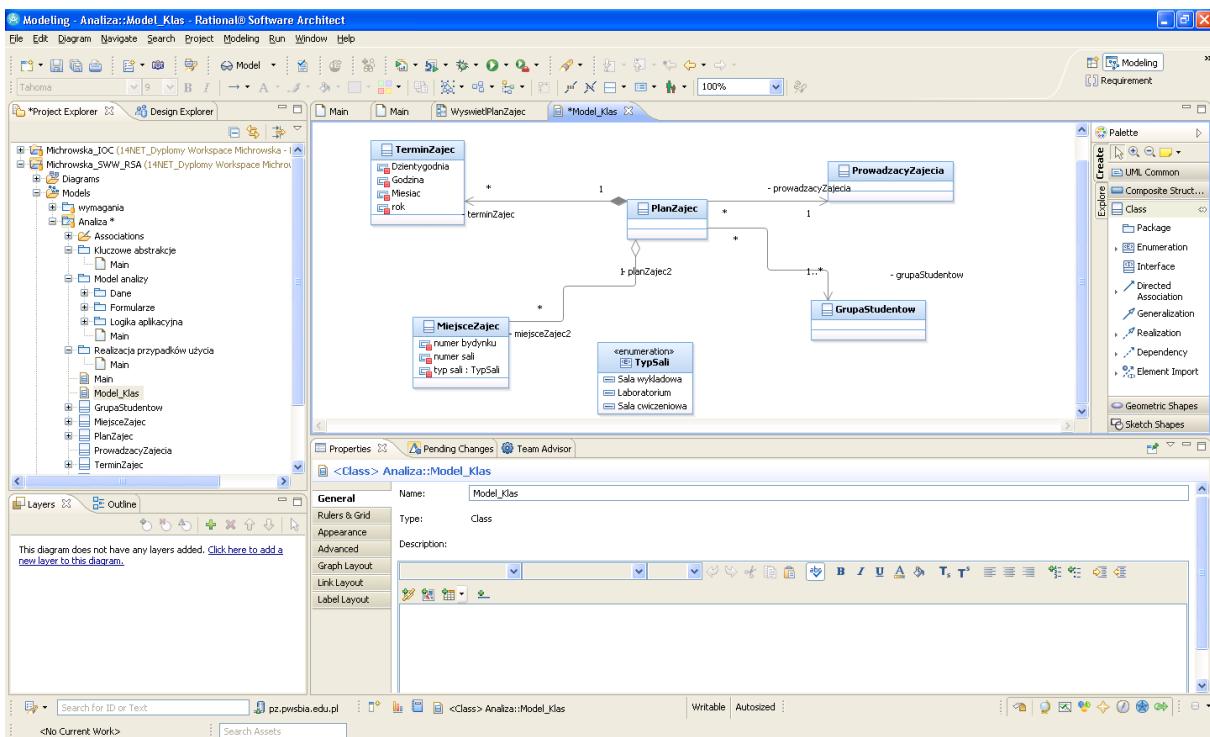
Pamiętaj! o ustaleniu prawidłowych krotności.

- ✓ Dodaj atrybuty każdej klasie:

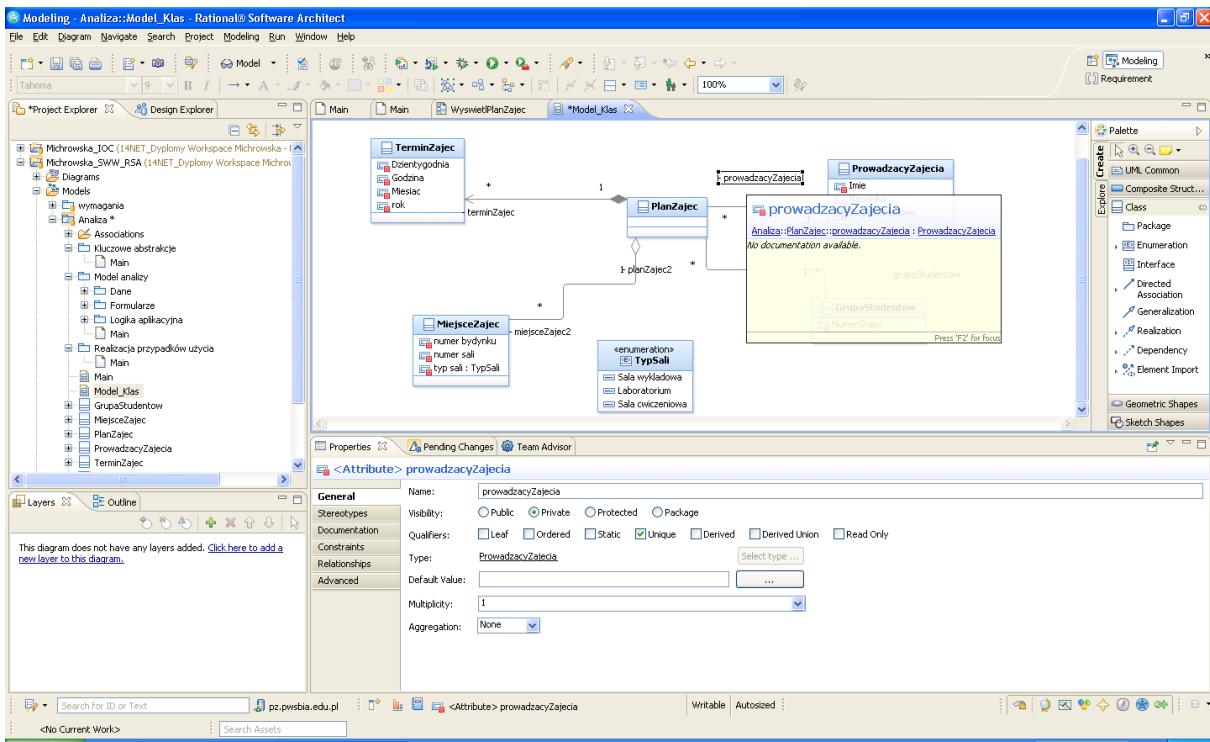


- ✓ Jeśli ustalisz, że potrzebujesz dodatkowej klasy, zrób to i śmiało dodają ją do diagramu.

Przykład: Dodajemy klasę enumeratywną z typami sal.



Aby powiązać klasę enumeratywną należy w nazwie typu dodać „:” otworzy się menu kontekstowe, gdzie będzie można dodać klasę enumeratywną.

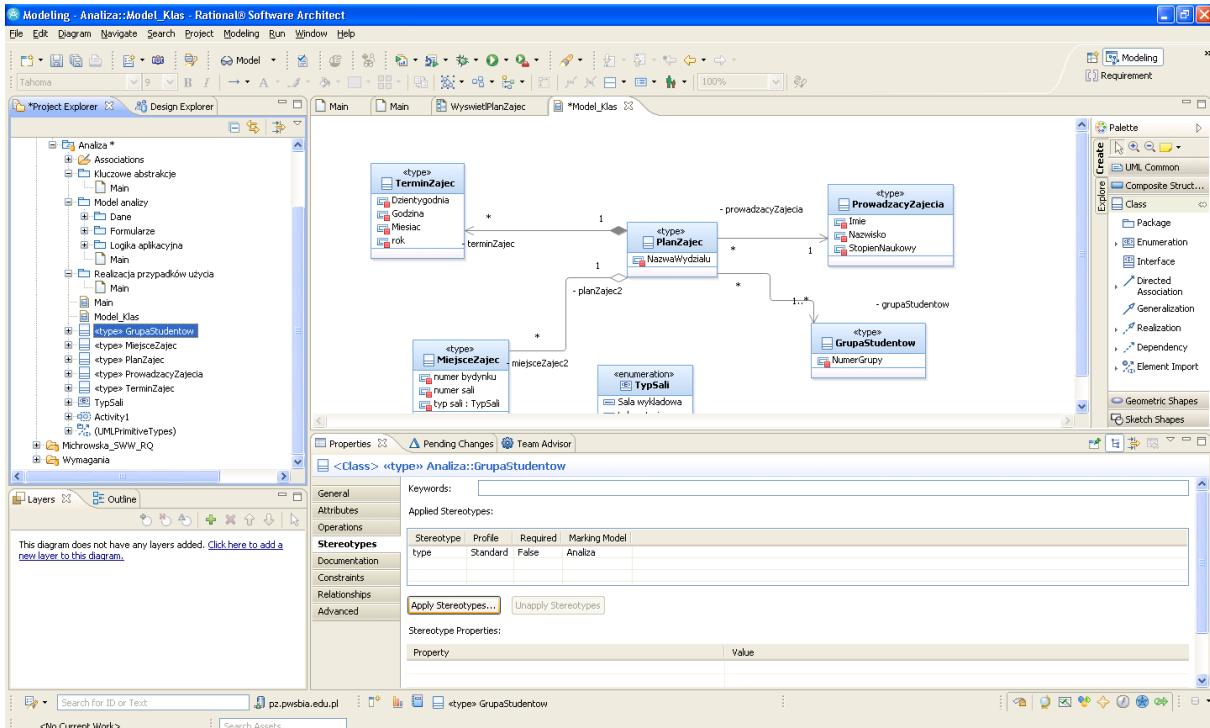


- ✓ Następnie zmodyfikuj stereotyp dla wszystkich klas na type.

Klasa powinna mieć stereotyp **type**.

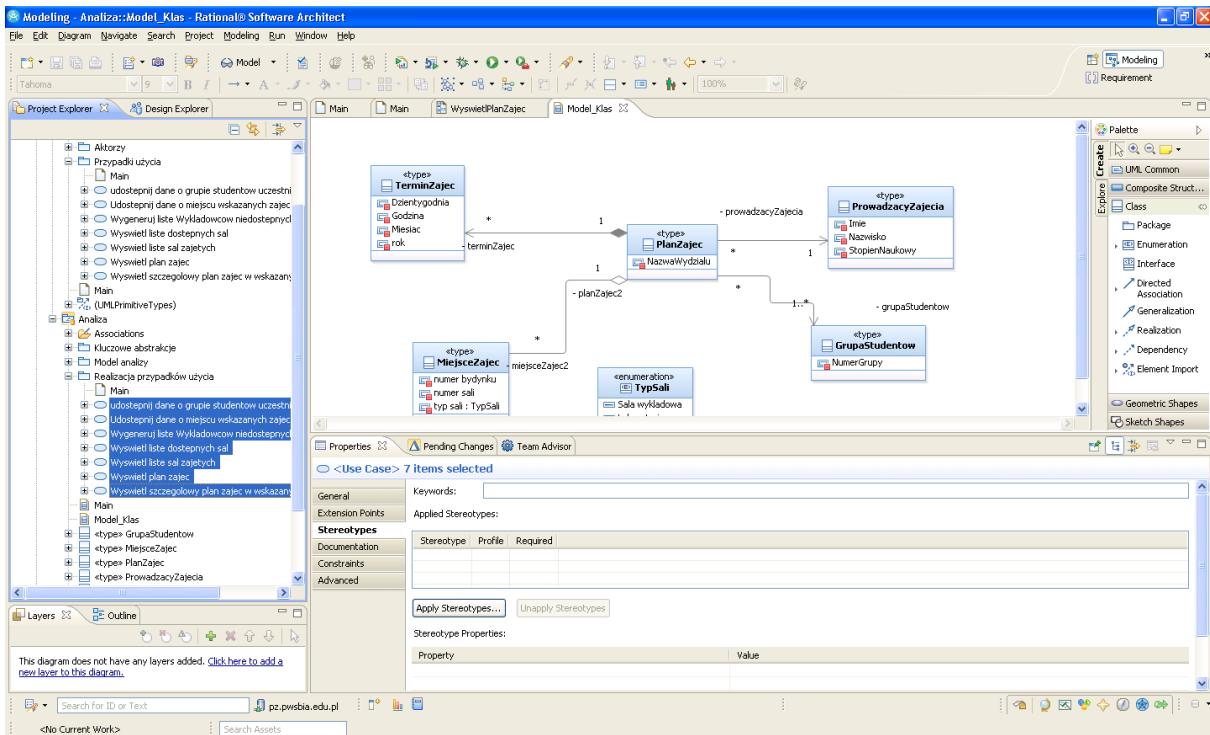
23. Stereotypy

- ✓ Zaznacz wszystkie klasy w oknie Project Explorer.
- ✓ Przejdz do Properties>>> Stereotypes.
- ✓ Apply Stereotypes, wybierz type.

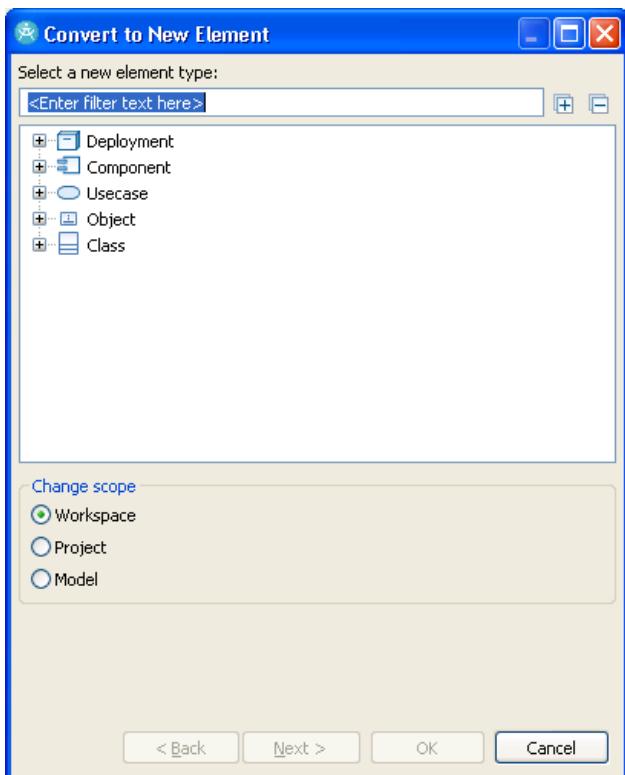


24. Realizacja przypadków użycia - do diagramów interakcji i VOPC

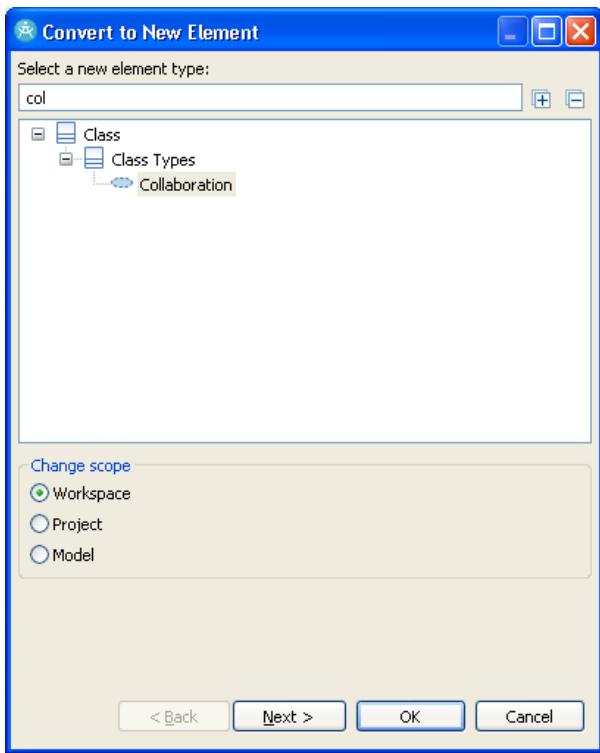
- ✓ Skopiuj z przypadków użycia do Realizacji przypadków użycia wszystkie utworzone wcześniej przypadki. Nie kasuj przypadków z wcześniejszego folderu, muszą tam pozostać. Natomiast po skopiowaniu zostaną zachowane referencje, aktualizujące elementy we wszystkich miejscach projektu.



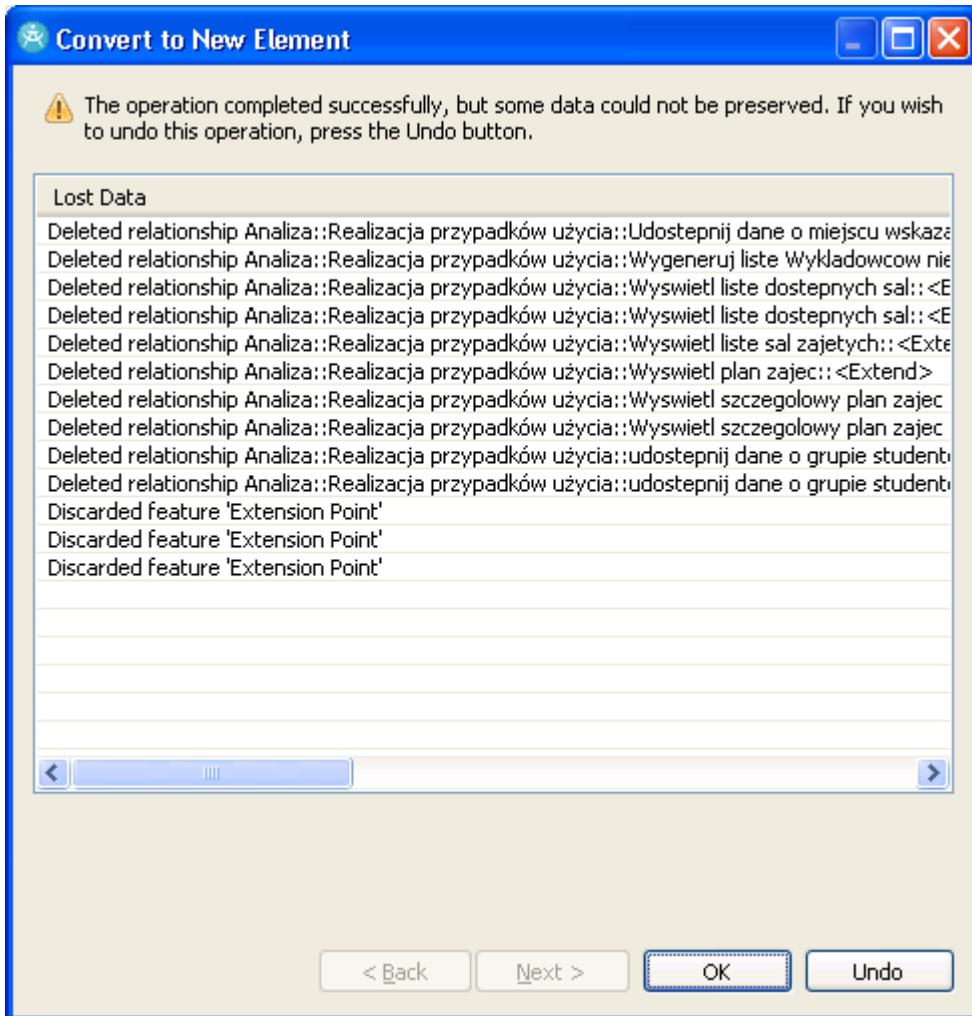
- ✓ Wejdź w opcję Refactor dla wszystkich przeniesionych do pakietu Realizacji przypadków użycia przypadków użycia .



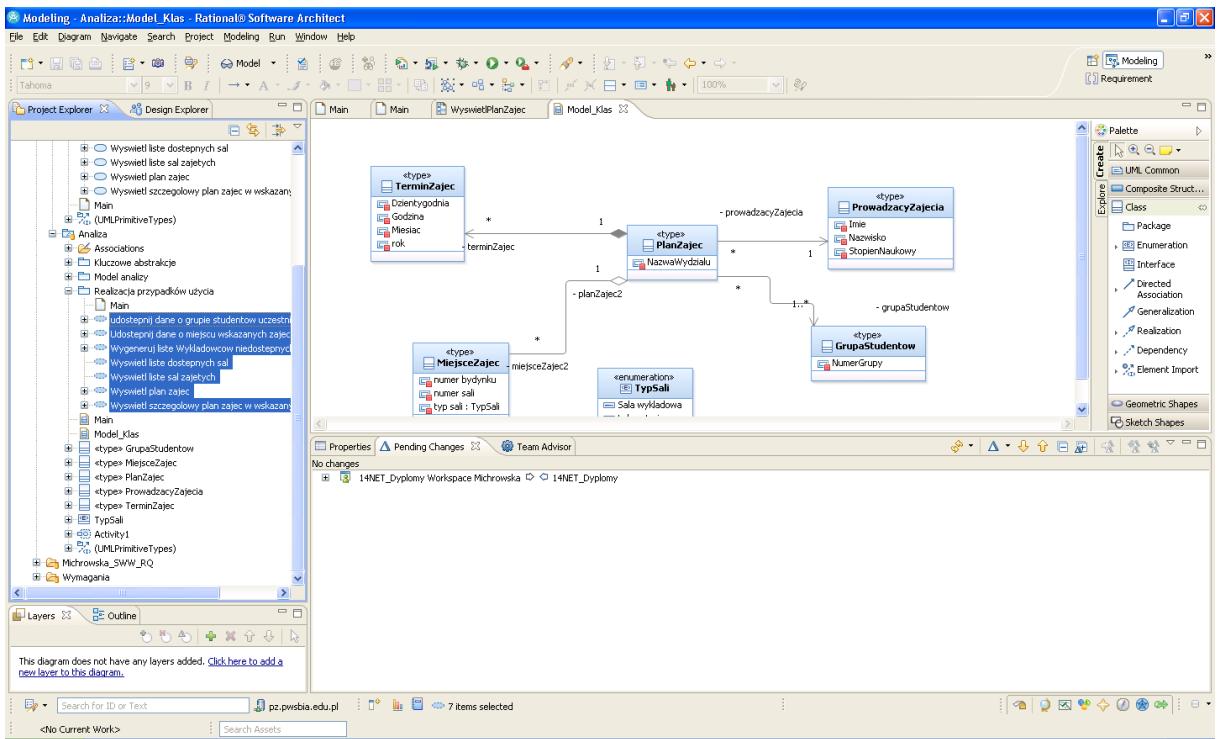
- ✓ Przekształć UC na elementy współpracy: Collaboration
✓ Wpisz w okienku wyszukiwania col aby wyświetlić odpowiedni element.
✓ Zaznacz go.



- ✓ Wybierz next.



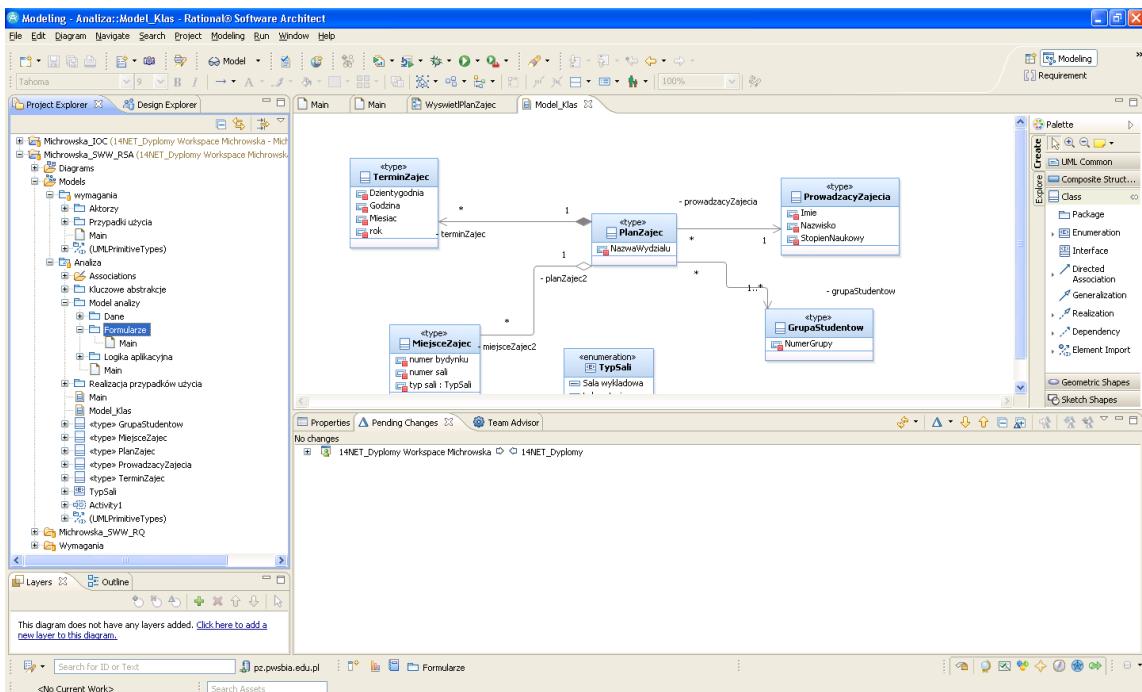
- ✓ Zatwierdź OK.
- ✓ Symbol UC powinien zmienić się na oznaczony przerwaną kreską. Jeśli nie, powtórz operację.

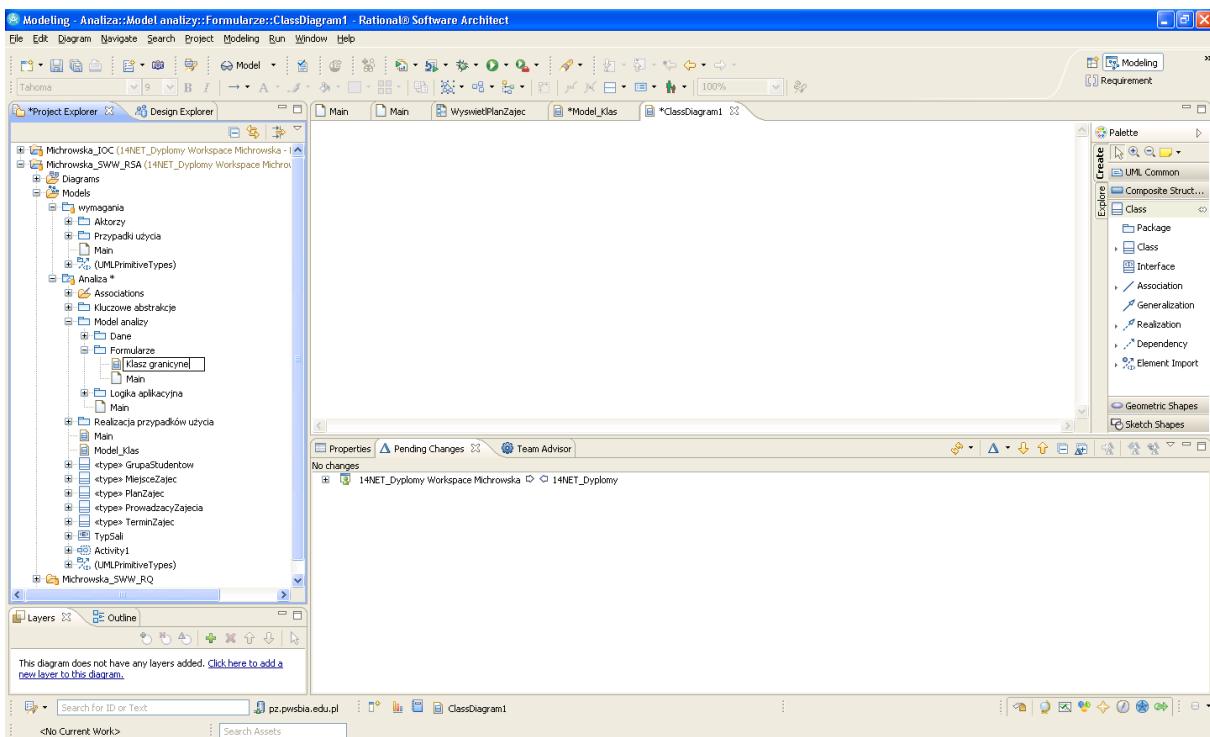


25. Tworzenie diagramów VOPC i sekwencji.

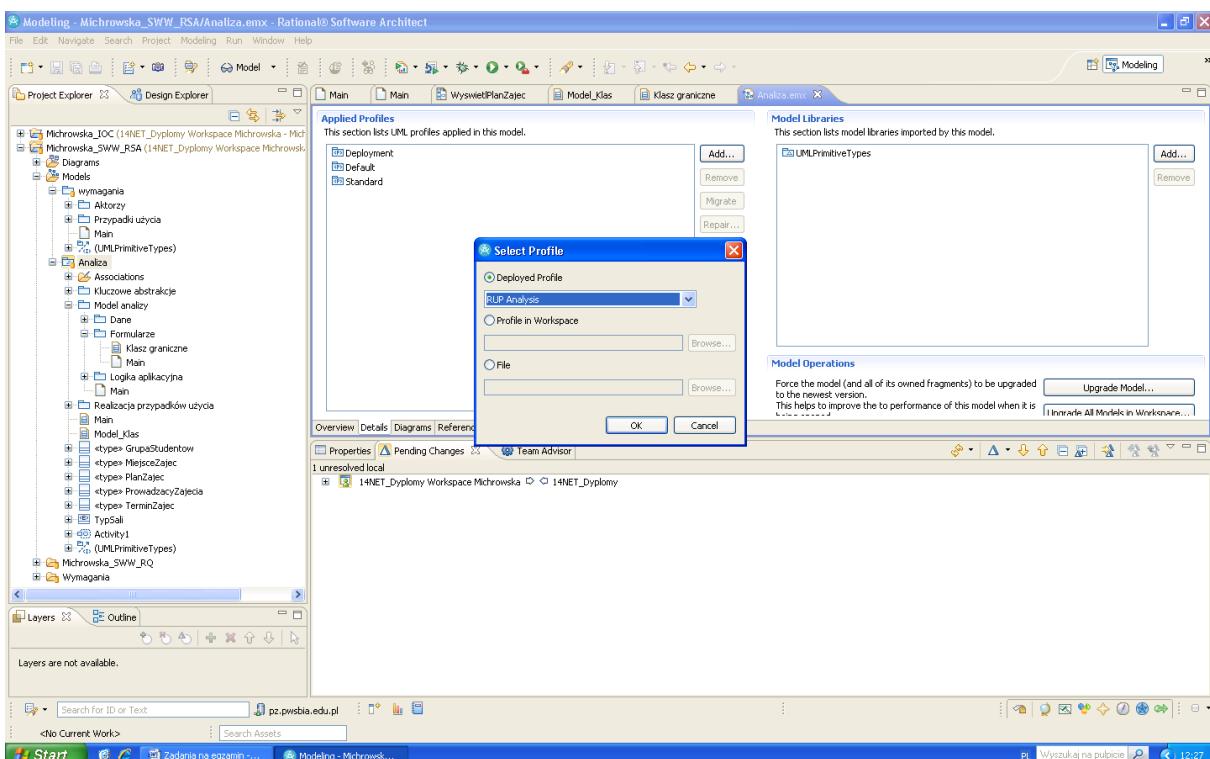
Będziemy tworzyć odpowiednie zbiory klas. Działanie rozpoczynamy od przechowania nazw przypadków użycia a następnie utworzenia im odpowiadających elementów. Klas granicznych, klas danych i klas sterujących. Klasy typu formatki są klasami granicznymi i jest ich tyle ile aktorów bierze udział w przypadku użycia.

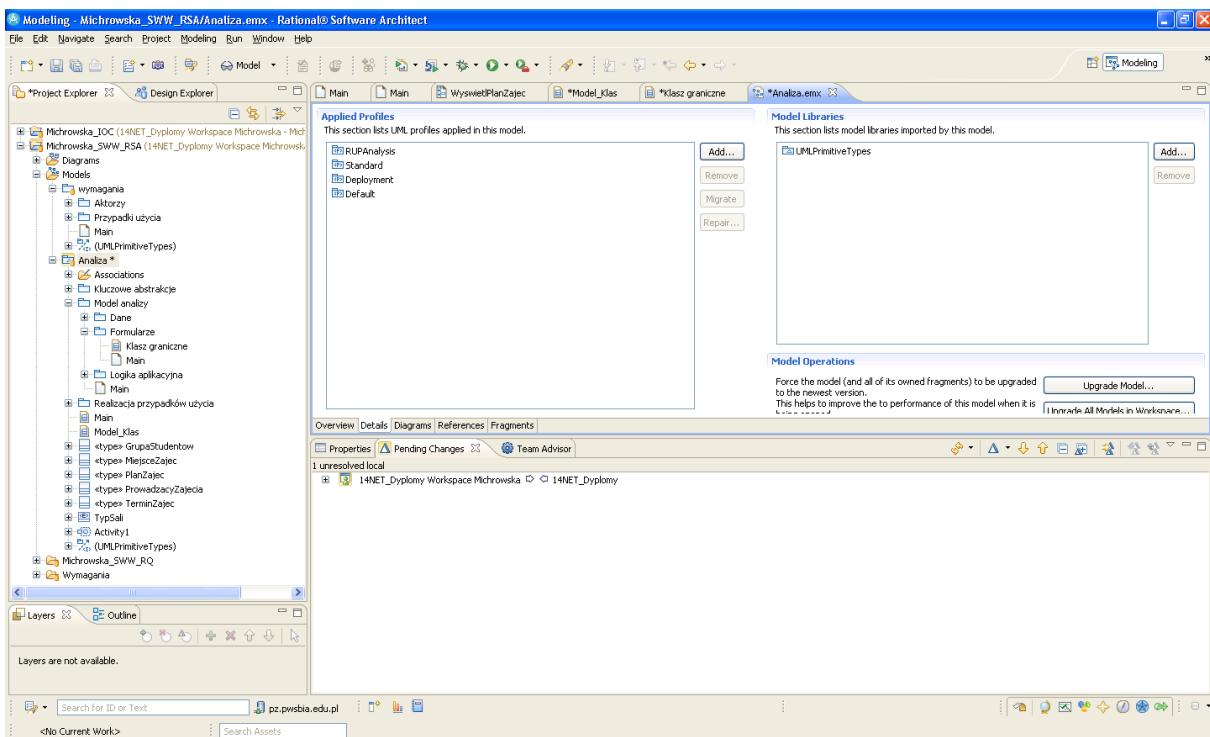
- ✓ W pakiecie Formularze twórz diagram klas o nazwie Klasy Graniczne.





- ✓ Dodaj na folderze analiza profil RUP. Jest to potrzebne żeby zrealizować klasy analityczne.
- ✓ Na folderze Analiza wybierz Open.
- ✓ Przejdź do zakładki Details.
- ✓ Wybierz ADD w Applies Profiles.
- ✓ Wybierz w okienku RUP Analysis.



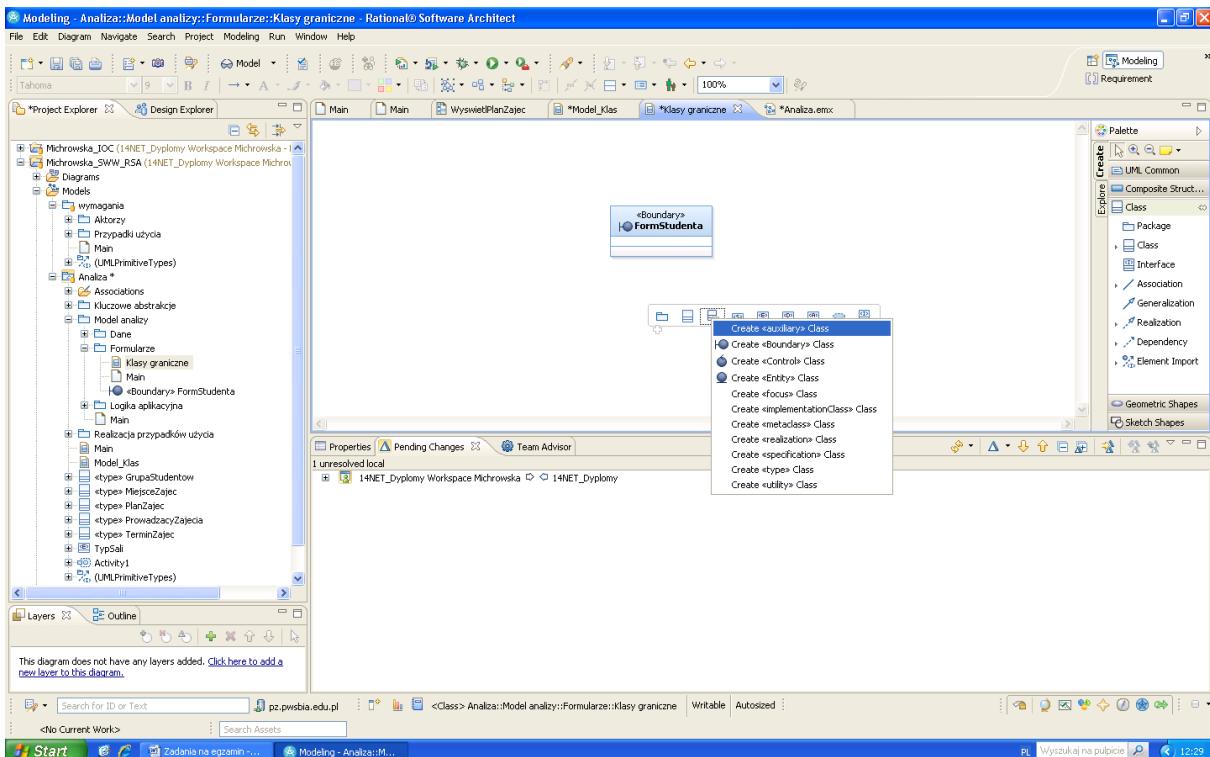


26. Specyfikacja klas analitycznych

27. Dodawanie klasy granicznej

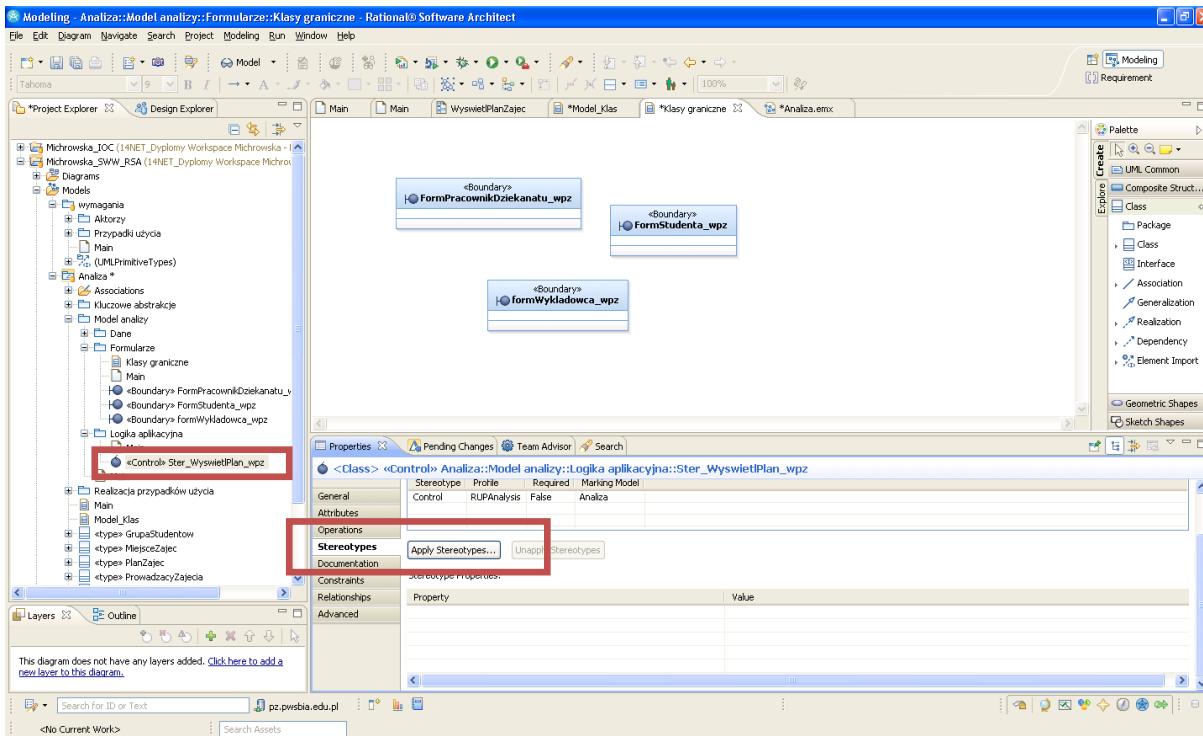
Pamiętaj! Przy tworzeniu klas oraz atrybutów i operacji tych klas nie używaj polskich znaków diakrytycznych oraz spacji, w późniejszym czasie powoduje to problemy z generowaniem kodu.

- ✓ Teraz możesz dodawać na diagramie Klasa graniczna formularze dla użytkowników systemu. Nazwa np.: Form[Aktor]
- ✓ Przytrzymaj mysz na polu diagramu.
- ✓ Wybierz klasę stereotypowaną z małą literą s, w menu masz dostępne klasa graniczne, sterujące i danych.

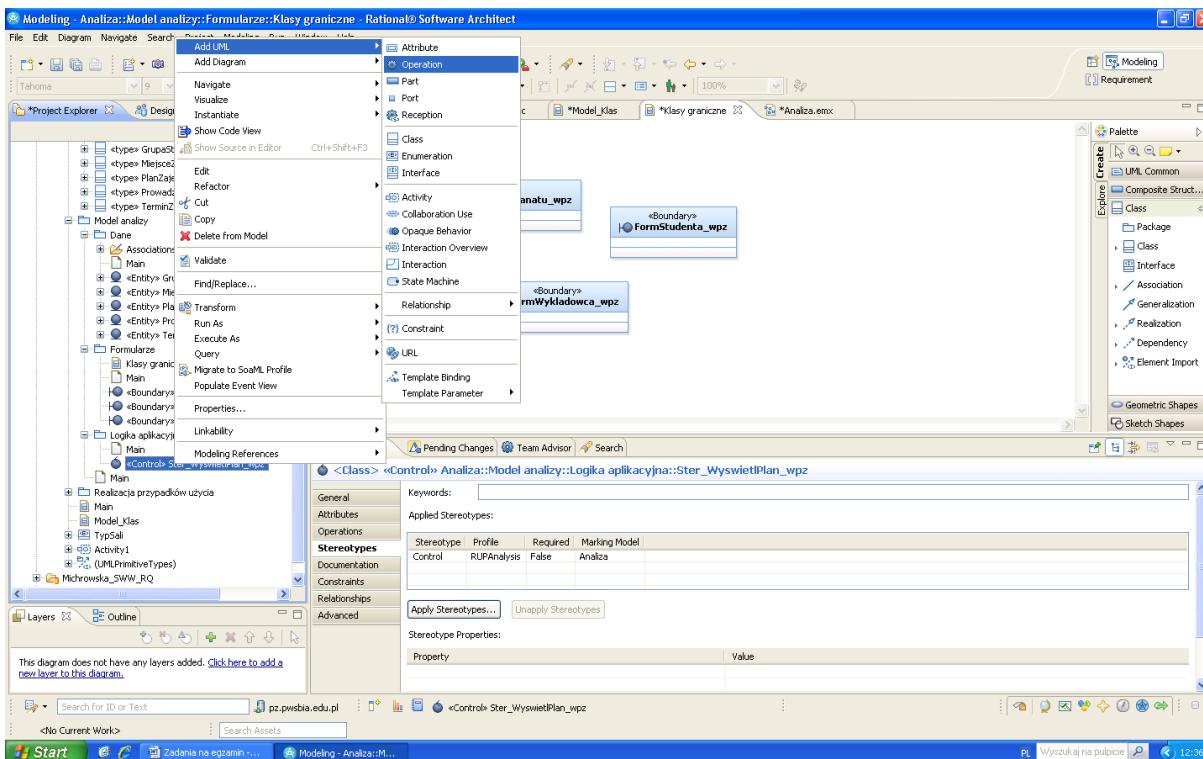


28. Dodawanie klasy sterującą

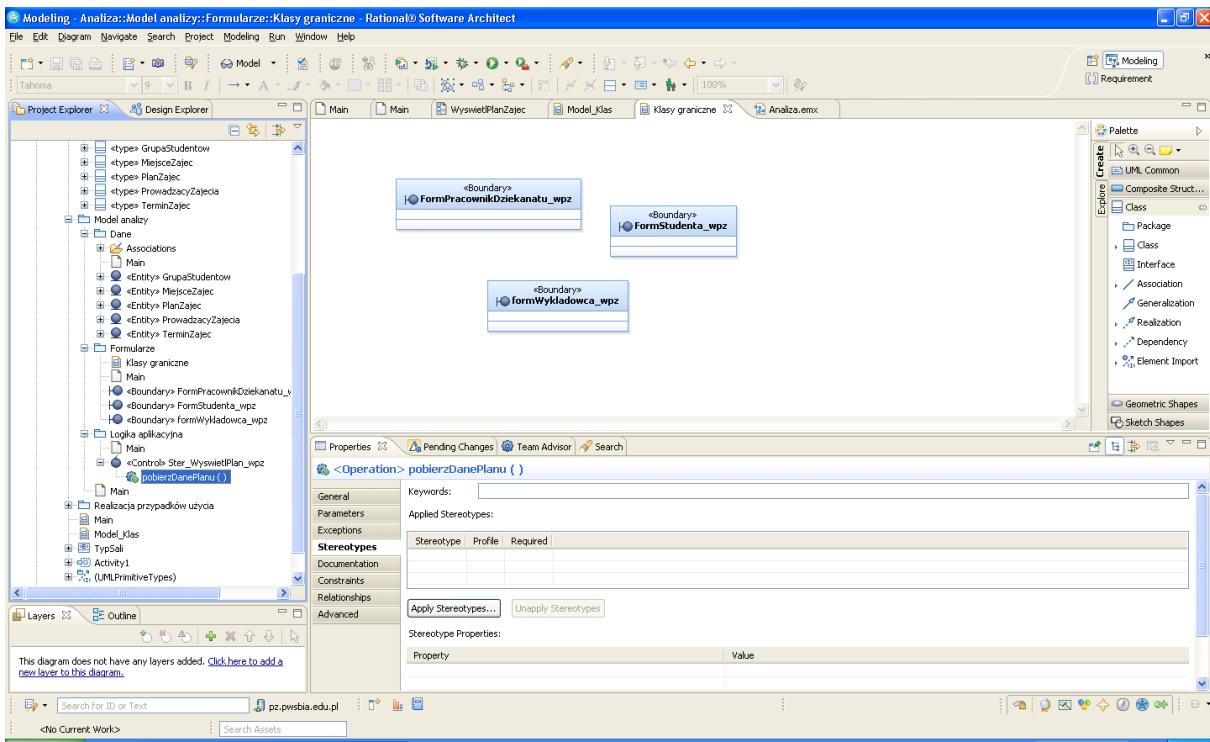
- ✓ W folderze logika aplikacyjna dodaj klasę. Wybierz Add UML>>> Class
- ✓ Nazwij klasę np.: Ster_JakaNazwa
- ✓ Przejdz do propertiese dodanej klasy, wybierz stereotypes>>> Apply Stereotypes... z otwartego okienka wybierz control



- ✓ Dodaj operacje do klasy sterującej, wybierając ją z menu kontekstowego

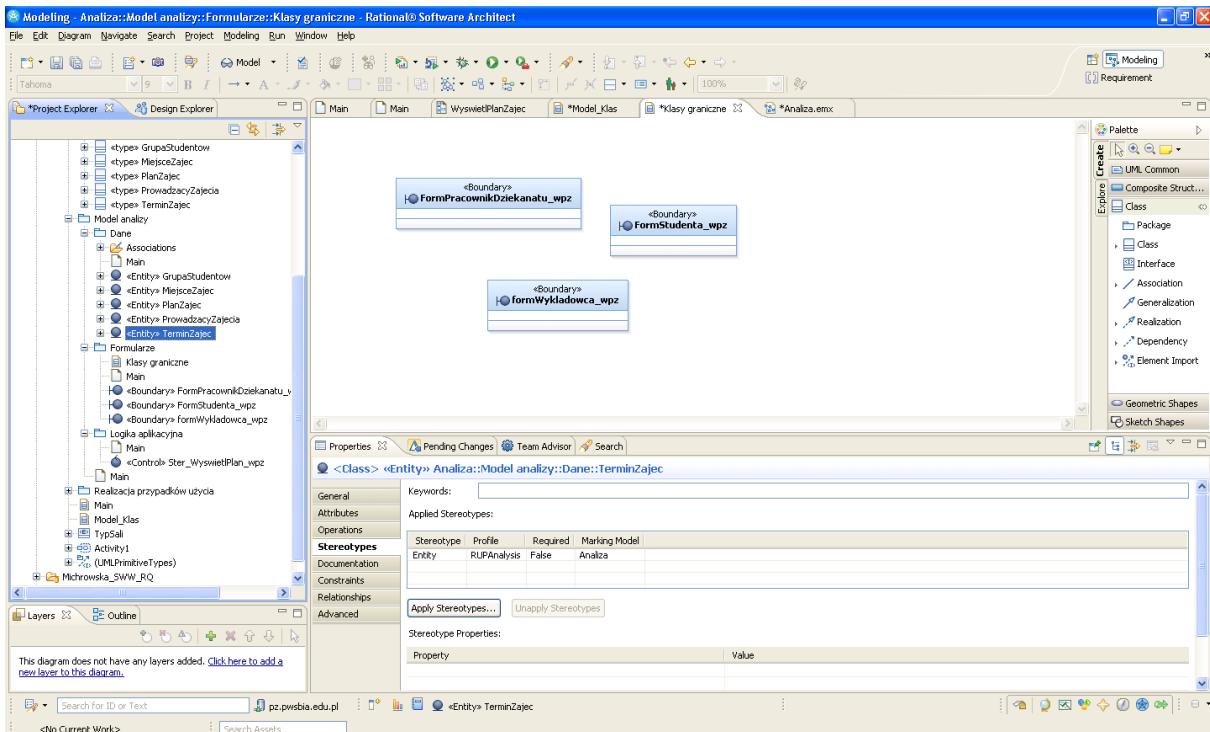


- ✓ Nadaj operacji nazwę.



29. Dodawanie klasy danych

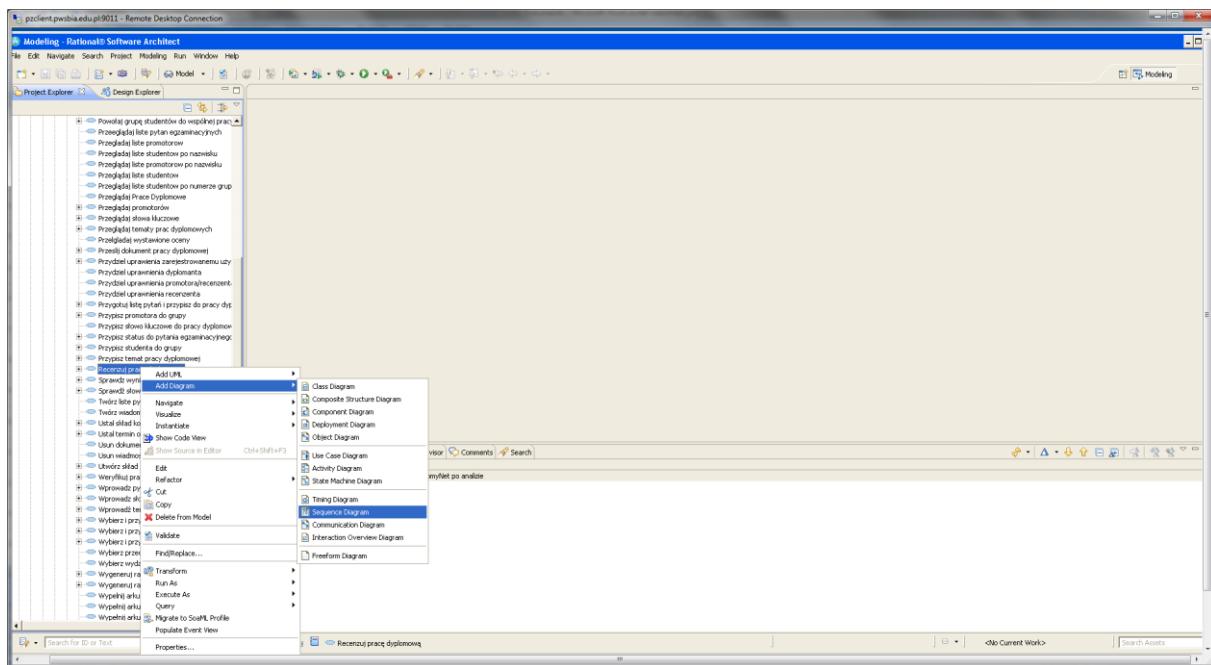
- ✓ Skopiuj wcześniej utworzone klasy ze stereotypem <type> i przenieś je do folderu dane.
- ✓ Zaznacz wszystkie klasy i przejdź do properties wybierz stereotypes>>> Apply Stereotypes... z otwartego okienka wybierz entity.
- ✓ Symbol klas powinien się zmienić na klasę abstrakcyjną, klasę danych.



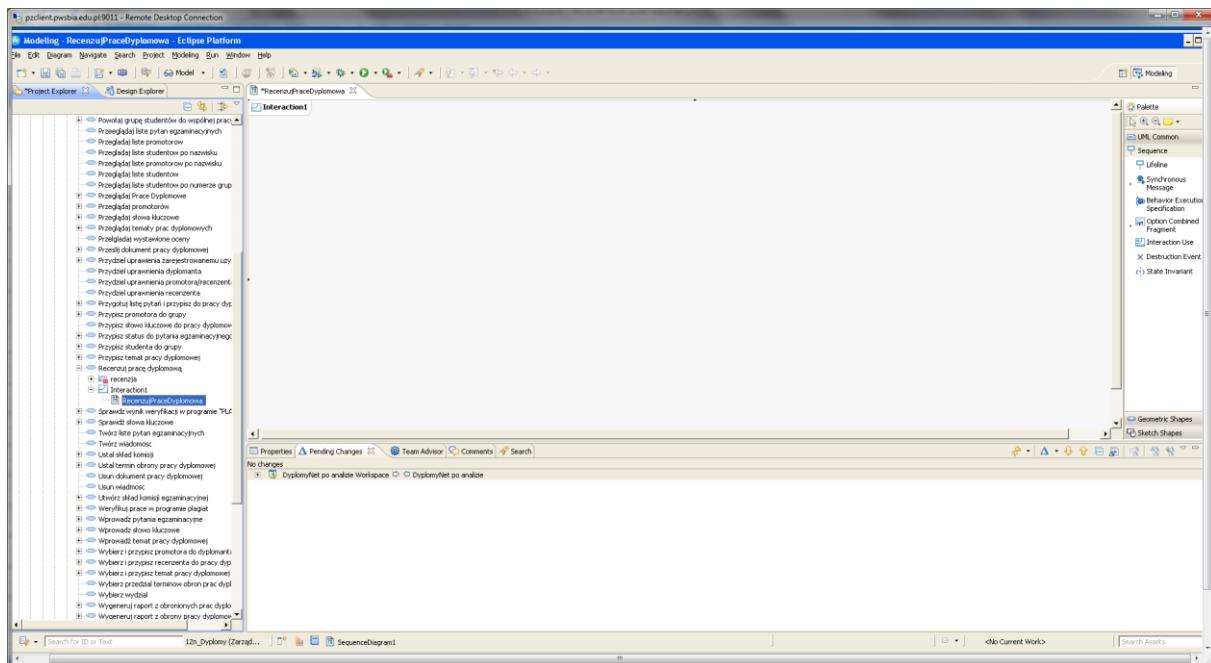
Pamiętaj! Twórz tylko jedną klasę sterującą dla każdego przypadku użycia oraz odpowiednie formularze i klasy danych dla każdego przypadku użycia.

30. Utworzenie diagramu sekwencji; fragment w modelu sekwencji

- ✓ Dla swojego przypadku utwórz Sequence Diagram:



- ✓ Nazwij go tak jak nazywa się przypadek użycia:



Na diagramie należy dodać poszczególne klasy w kolejności:

- ✓ Boundary
- ✓ Control
- ✓ Entity
- ✓ (ewentualnie) Entity- jeśli w przypadku jest potrzeba użycia więcej niż jednej klasy danych
- ✓ (ewentualnie) Boundary- jeśli w przypadku jest potrzebny interfejs jakiejś zewnętrznej aplikacji

Powyższe klasy abstrakcyjne pobieramy z folderu Model_Analizy. Odpowiednio dla każdej z klas z folderów:

- ✓ Formularze
- ✓ Logika Aplikacyjna

- ✓ Dane

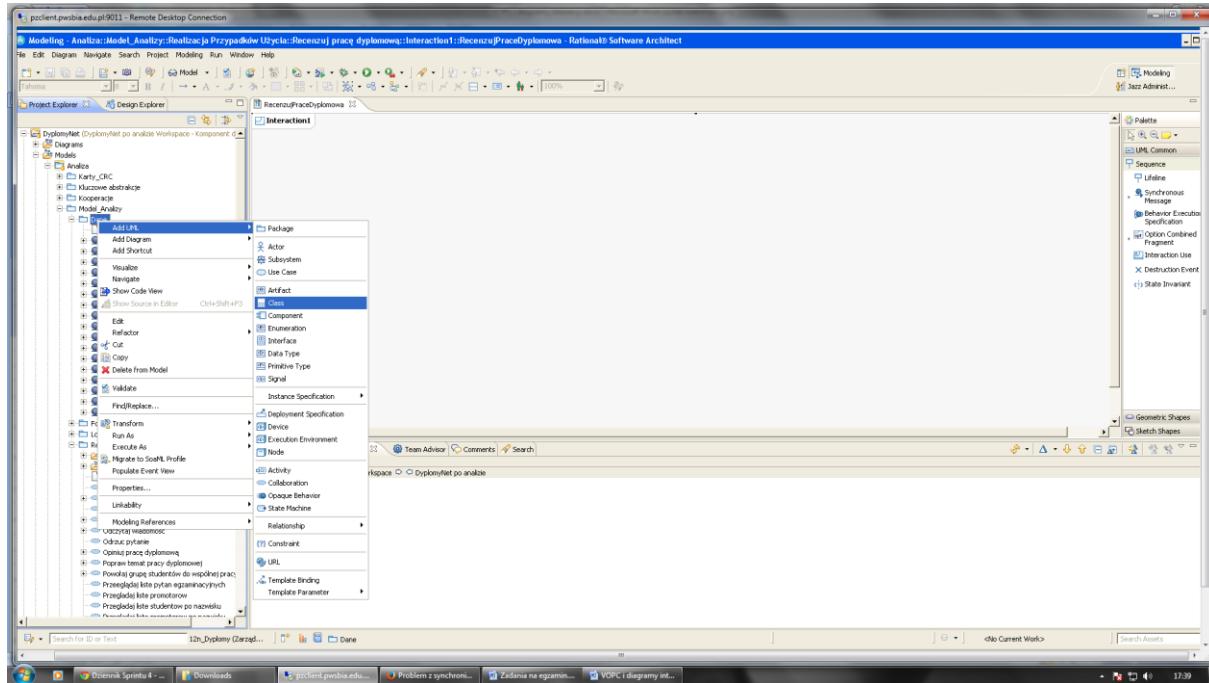
Jeśli w powyższych folderach brak jest formularza, bądź klasy danych należy je sukcesywnie dodawać.

W przypadku klas sterujących dla każdego realizowanego przypadku należy dodać klasę sterującą.

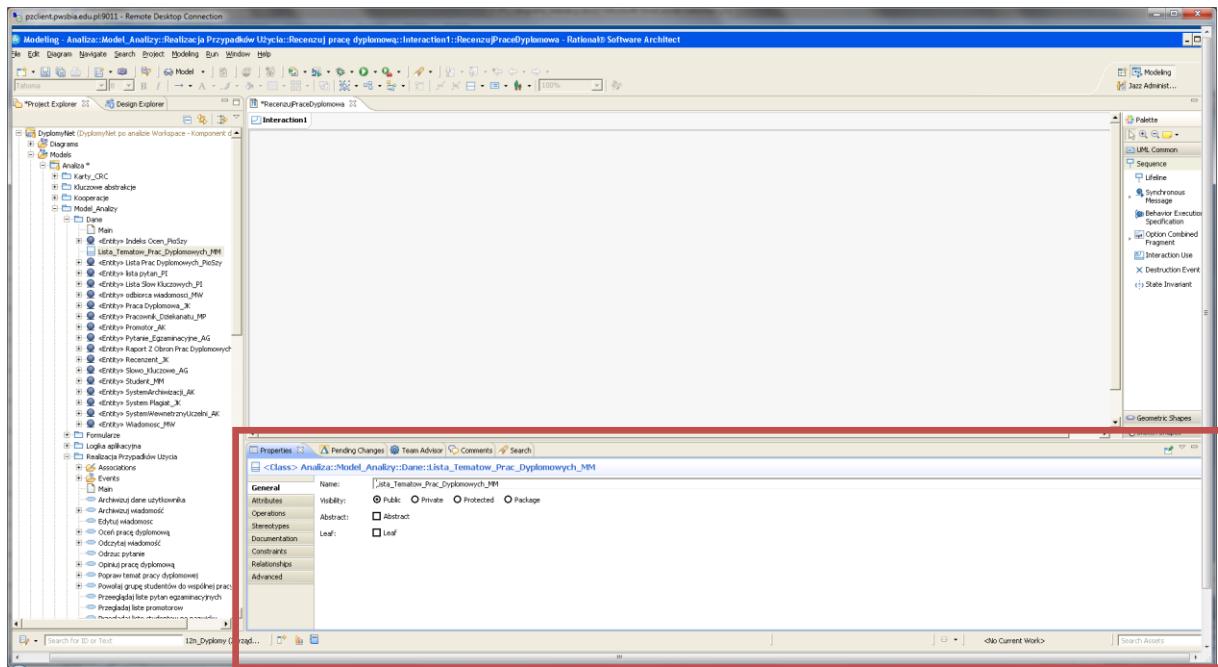
Dodawanie klas abstrakcyjnych do Modelu_Analizy

Dodawanie klas <>entity> danych

- ✓ Dodaj UML Class do folderu dane

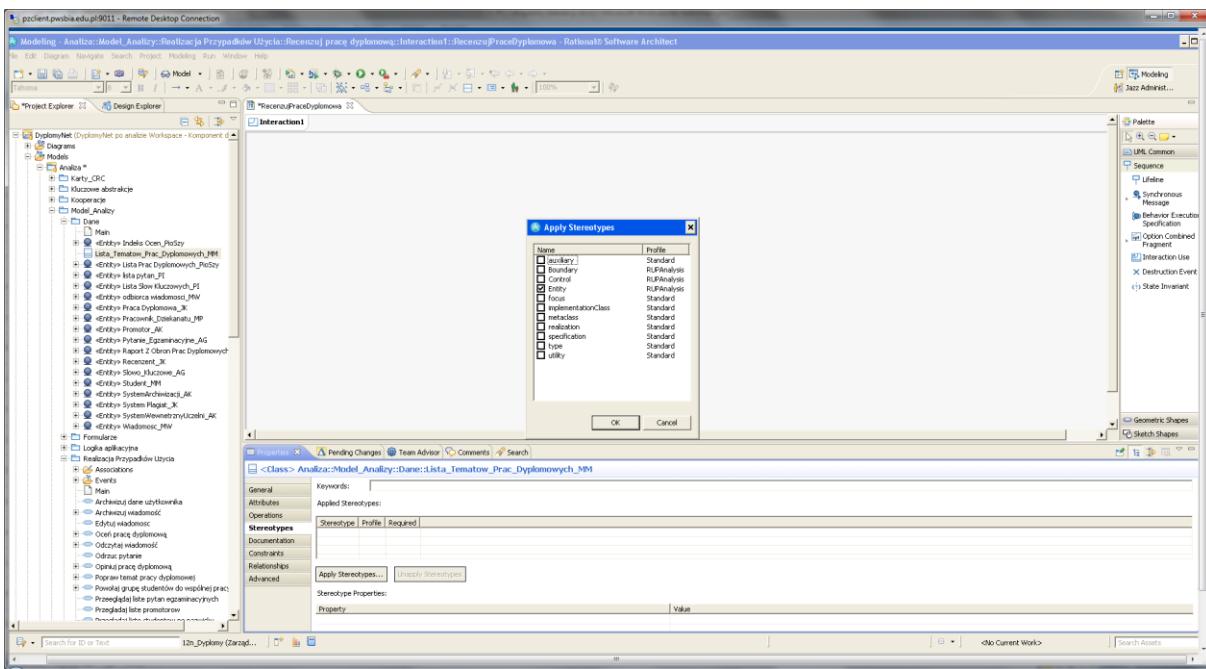


- ✓ Nadaj odpowiednią nazwę odnoszącą się do przypadku użycia i zakończ swoimi inicjałami.
- ✓ Zaznacz utworzoną klasę i przejdź do właściwości Properties

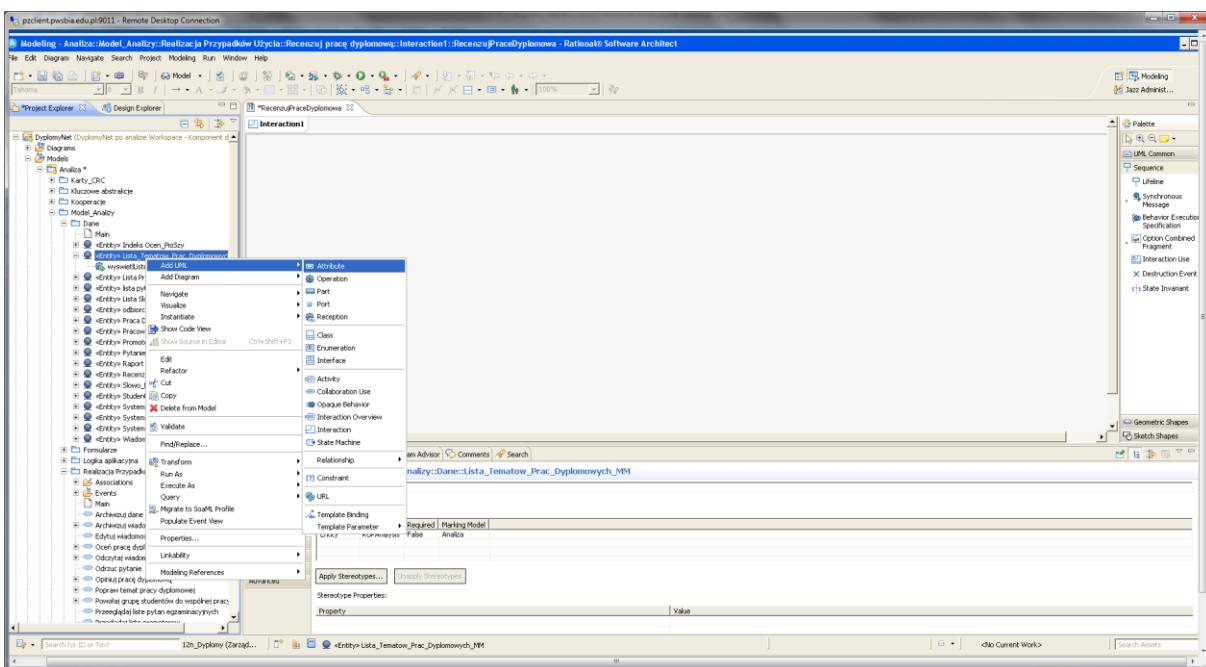


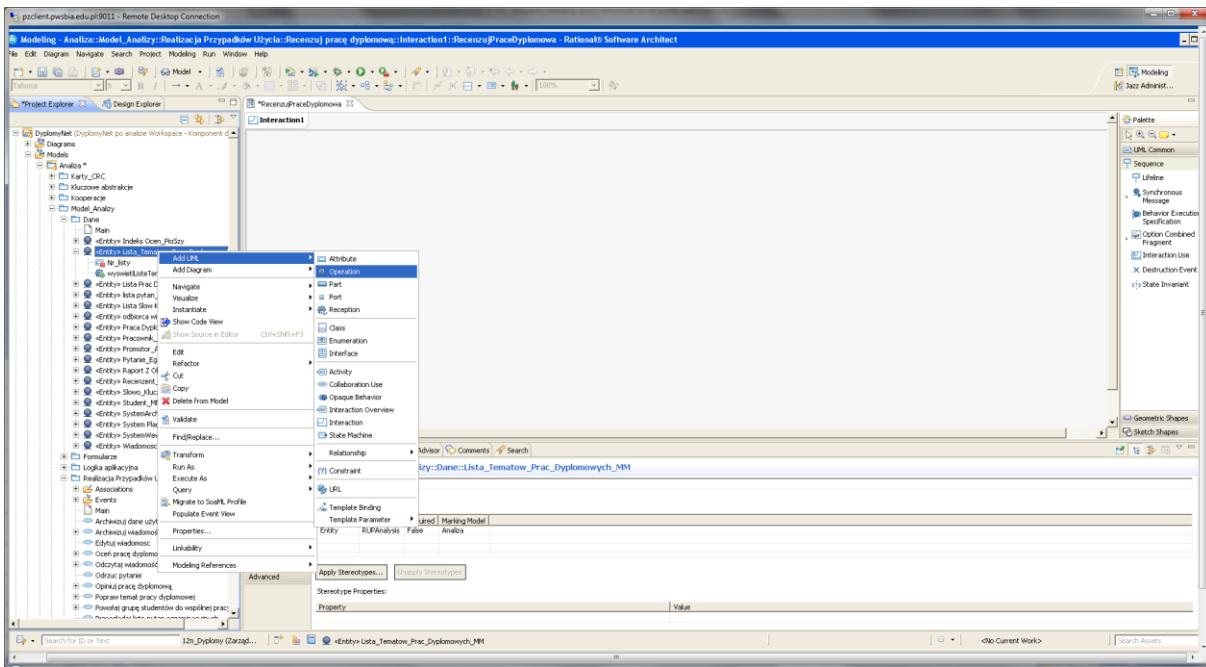
Przejdź we właściwościach do zakładki **Stereotypes**, wybierz **APPLY STEREOTYPES...**

- ✓ w okienku, które się otworzyło zaznacz Entity.

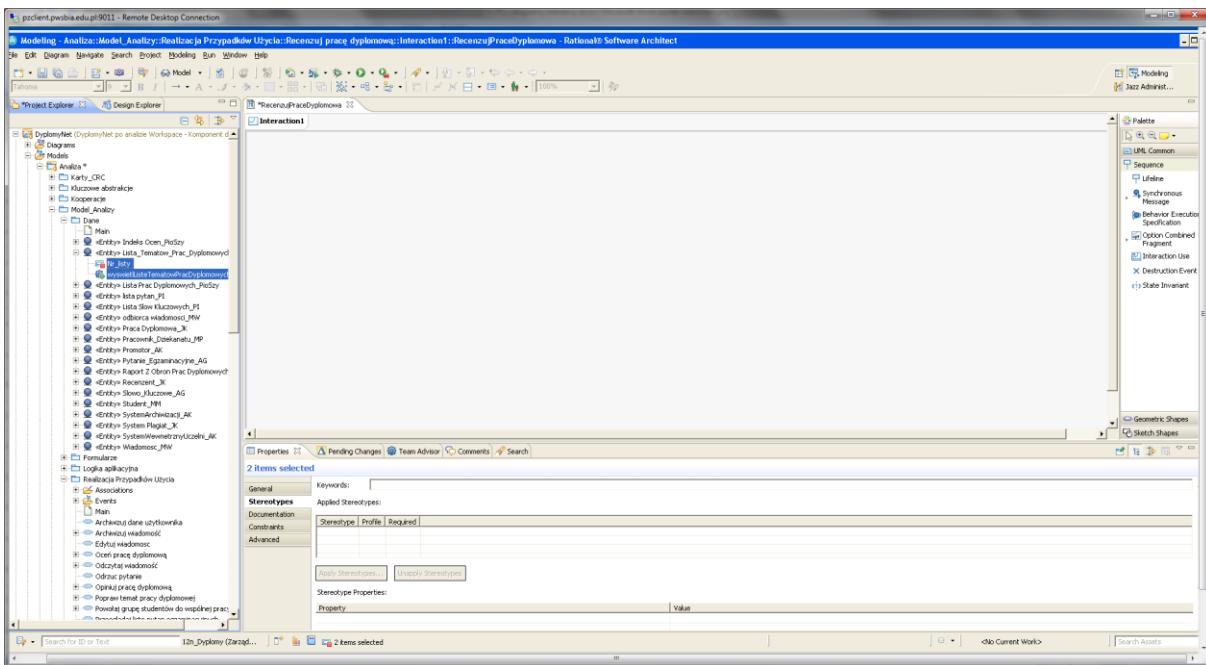


- ✓ Zatwierdź ok.
- ✓ Symbol klasy został zmieniony na klasę danych.
- ✓ Dodaj Atrybuty i operacje przewidziane dla tej klasy:

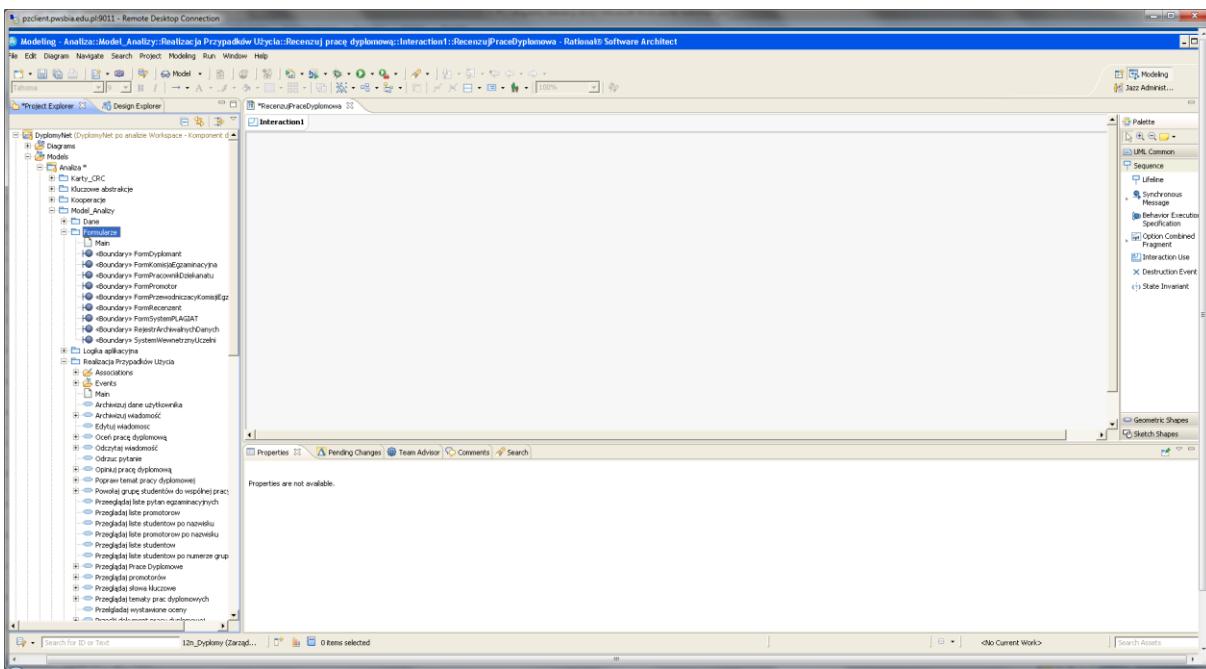




Atrybut i operacja zostały dodane.

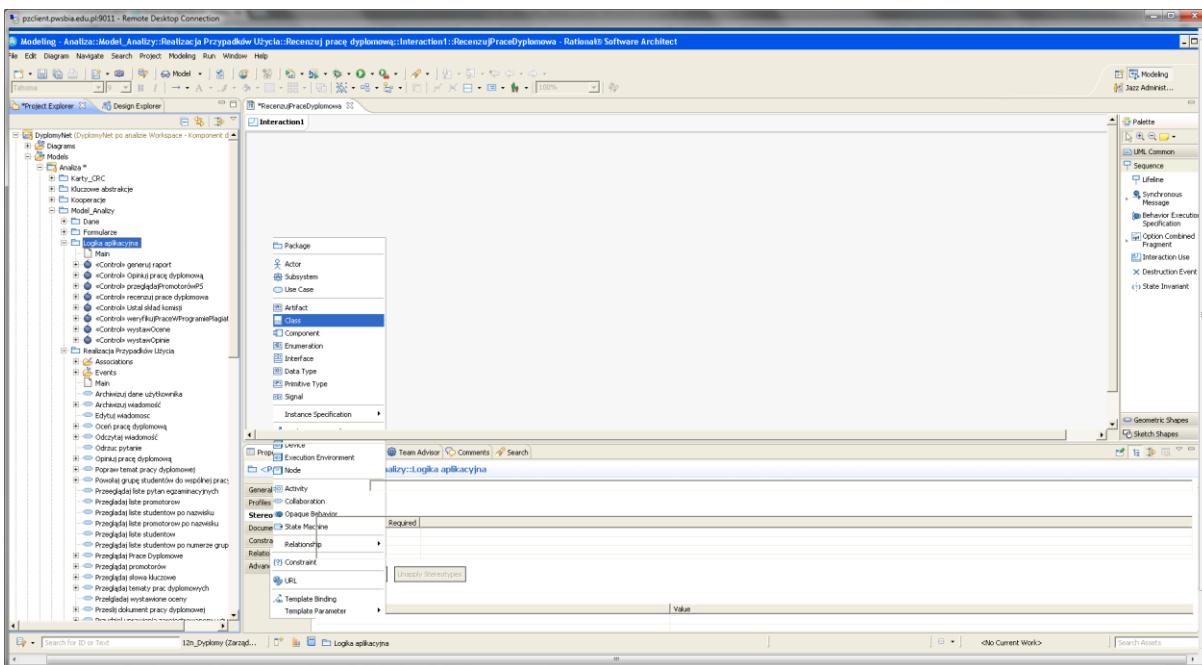


Dodawanie klasy granicznej:

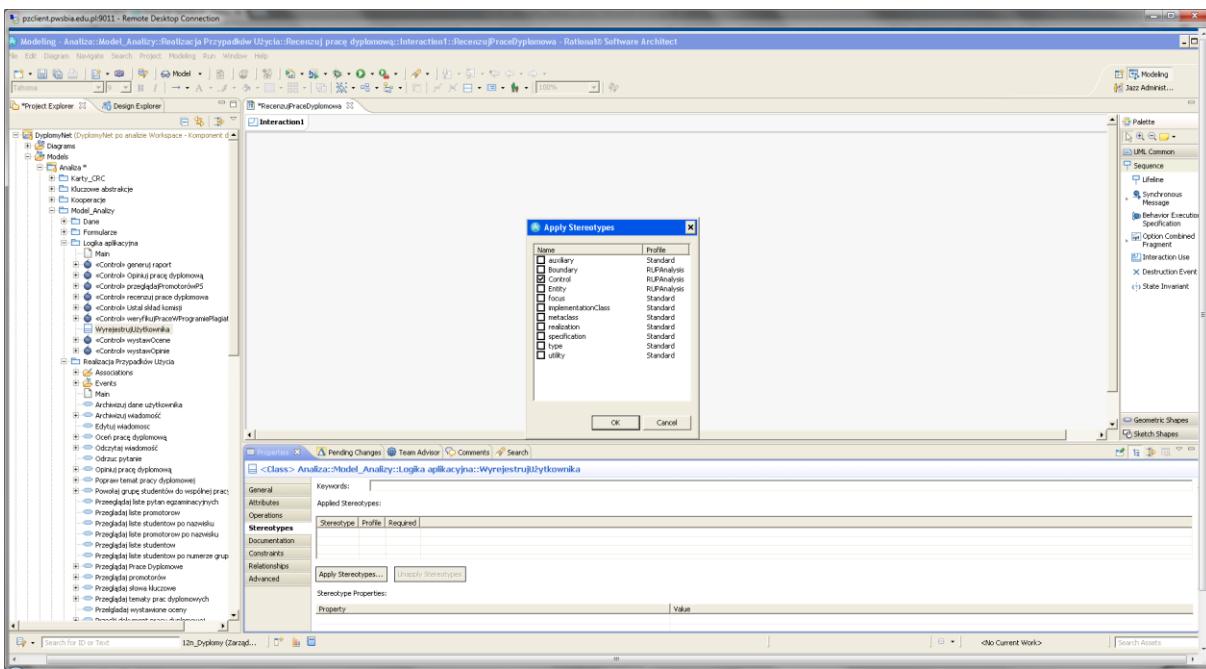


Wszystkie klasy graniczne powinny być już utworzone, jeśli jednak jakieś brakuje postępuj jak przy dodawaniu klasy danych, z tym, że w propertise wybierz stereotyp **Boundary** zamiast Entity. Nie będą potrzebne atrybuty i operacje na razie.

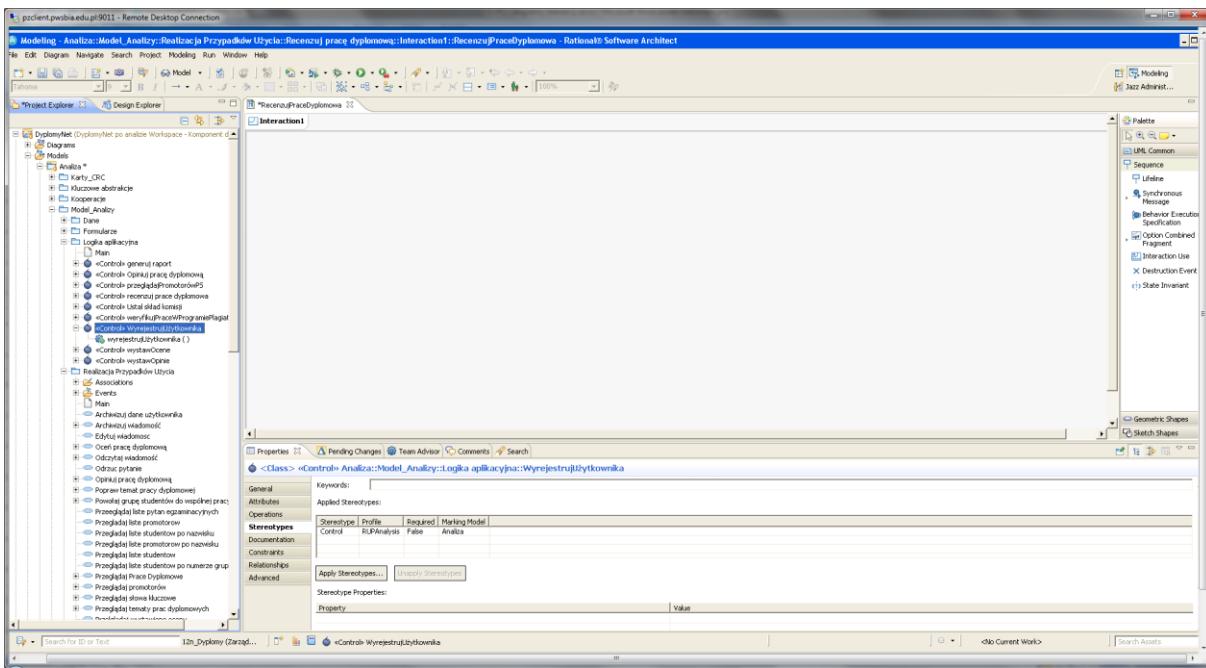
Dodawanie klasy sterującej:



- ✓ Przypisz stereotyp Control

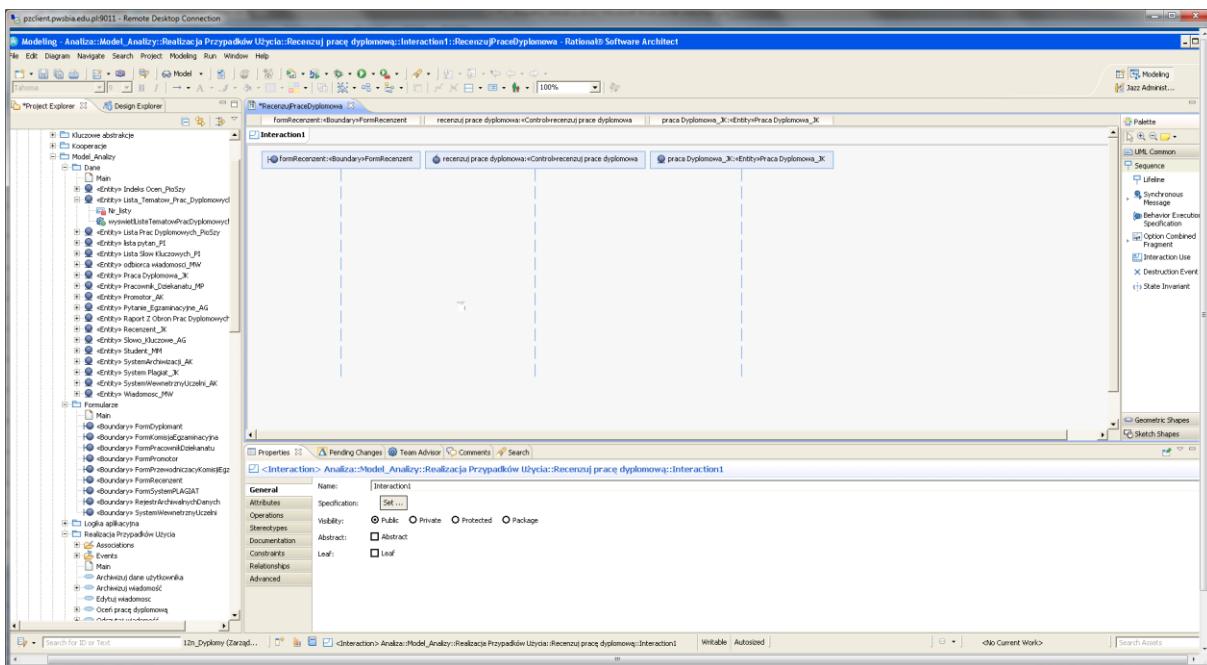
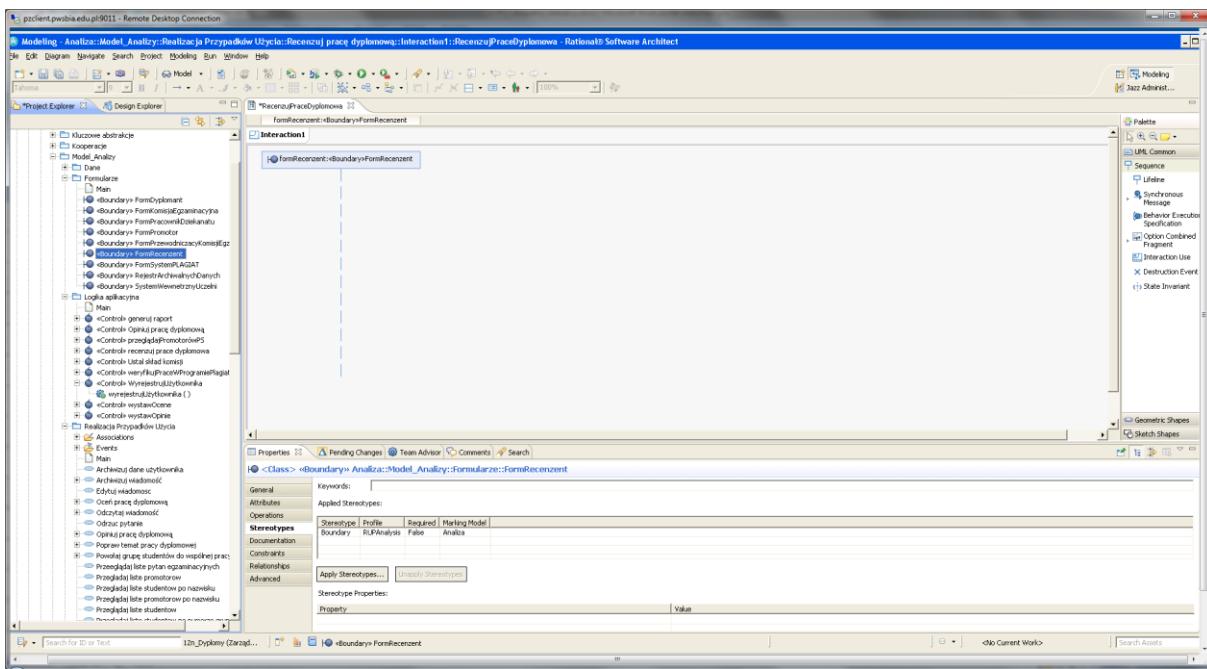


✓ Dodaj operację:

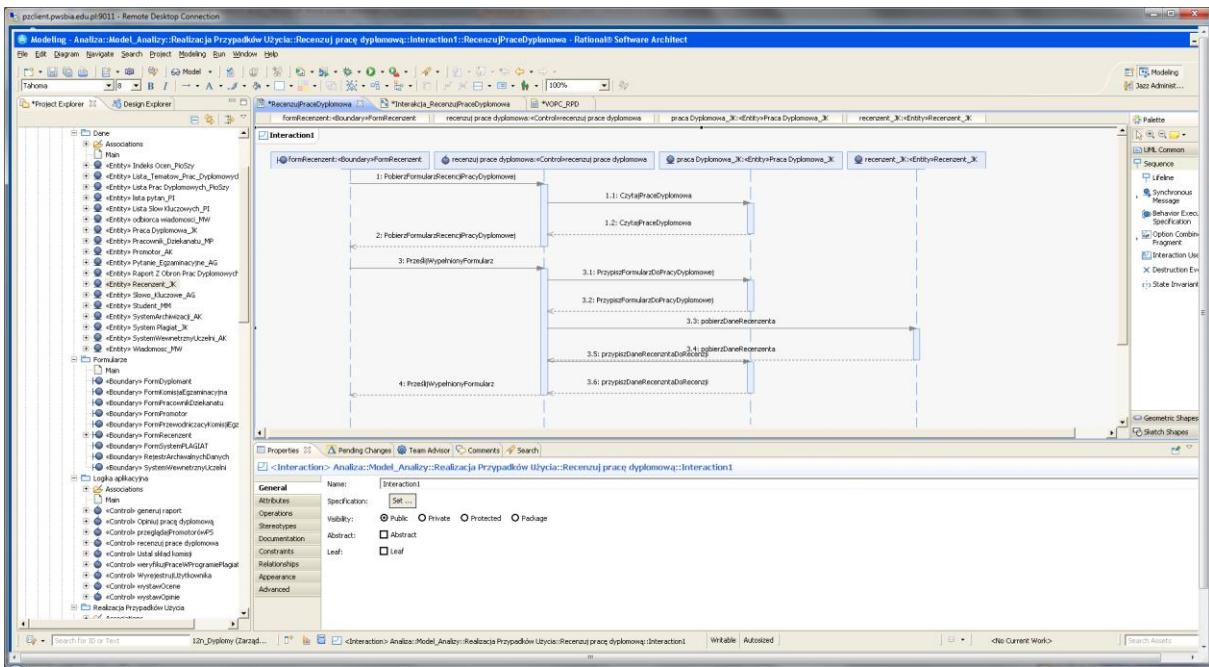


Wracamy do Tworzenia **diagramu Interakcji**.

W oknie edycji otworzyła się karta Twojego diagramu dodaj na nim odpowiednie klasy zgodnie z kolejnością zapisaną powyżej przeciagając klasę z jednego z folderów na diagram:

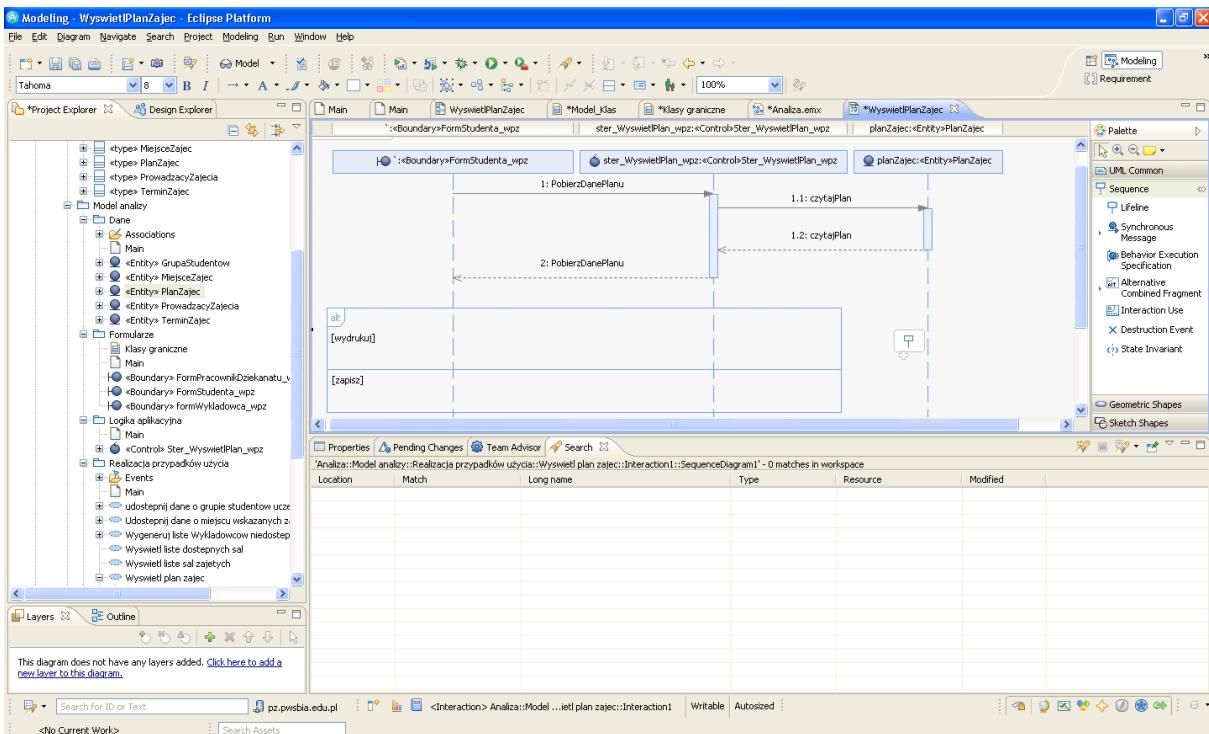


- ✓ Dodaj komunikaty synchroniczne

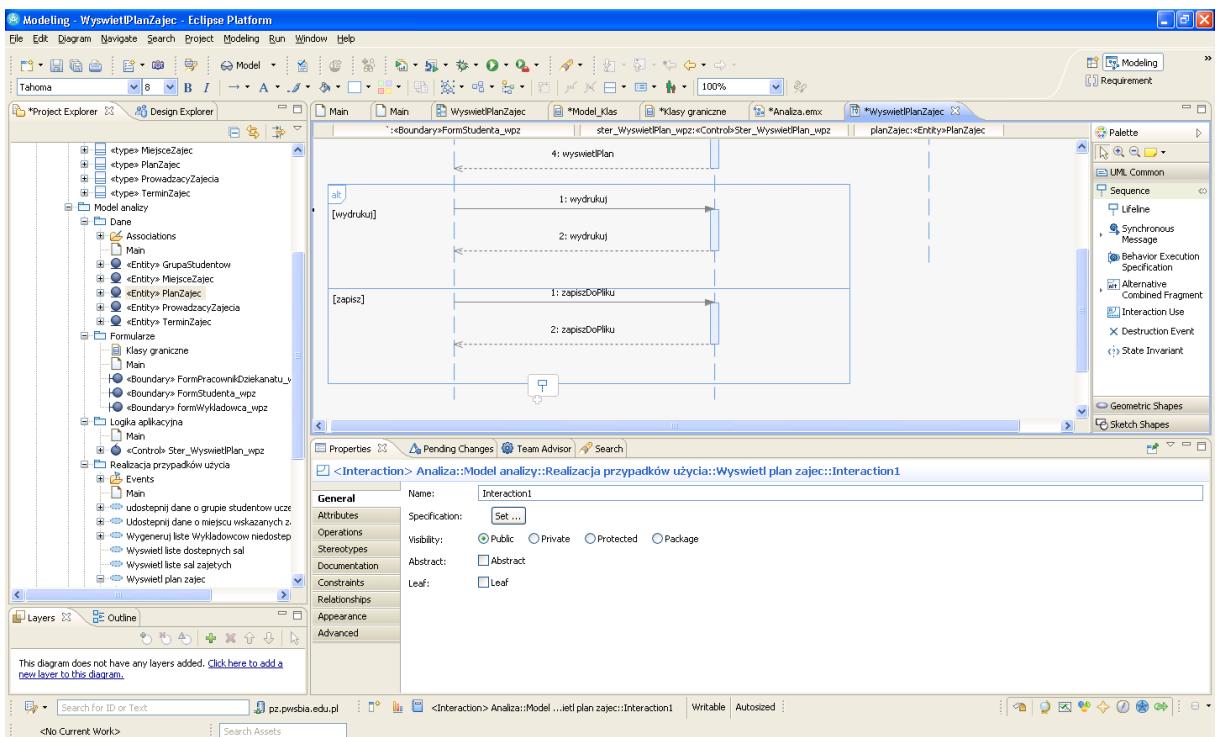


Dodaj alternative CombinedFragment (jeśli potrzebny)

Wpisz opcje alternatywy

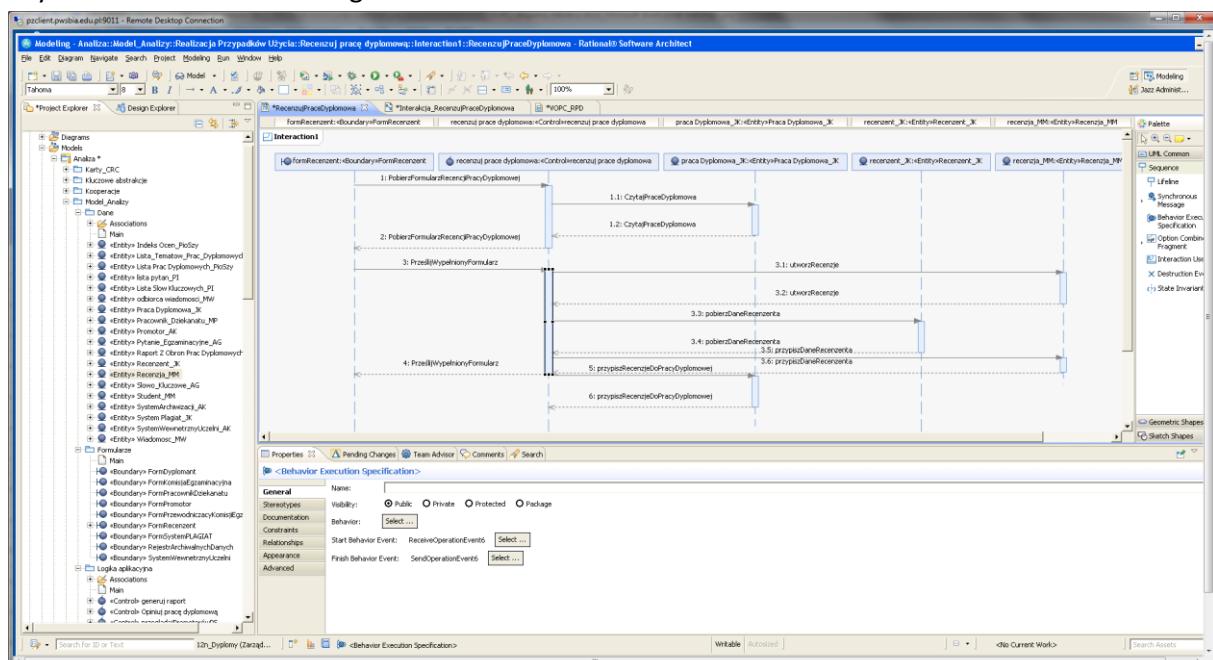


Dodaj operacje synchroniczne:

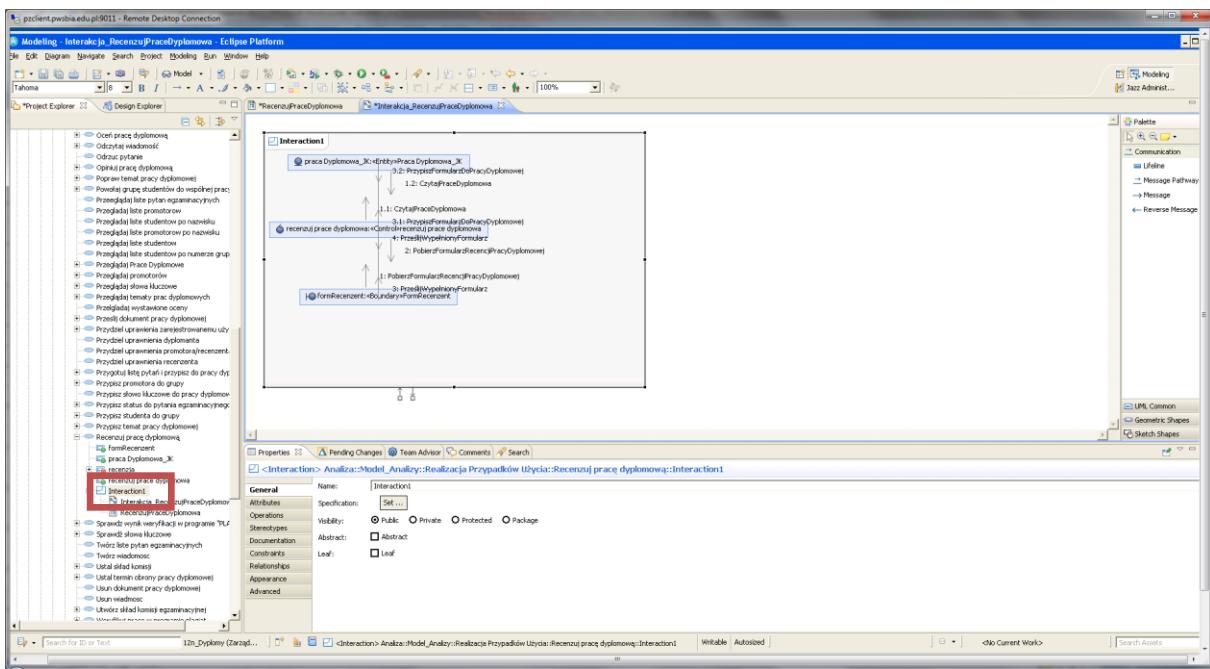


31. Przekształć model sekwencji w model interakcji

- ✓ Ustaw się na elemencie INTERACTION Twojego przypadku użycia (Utworzył się po dodaniu diagramu komunikacji)
- ✓ Wybierz prawym ADD UML DIAGRAM (do wyboru będą tylko dwa)
- ✓ Wybierz Communication Diagram

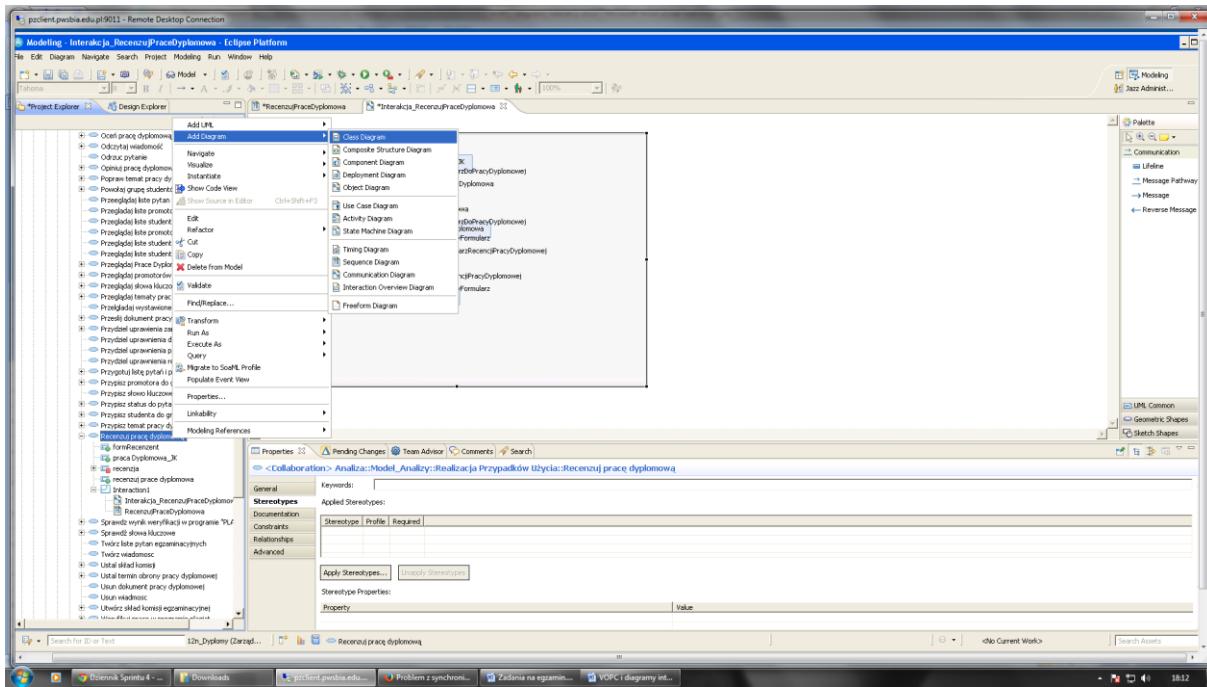


- ✓ Model utworzy się samoczynnie, na podstawie modelu sekwencji.

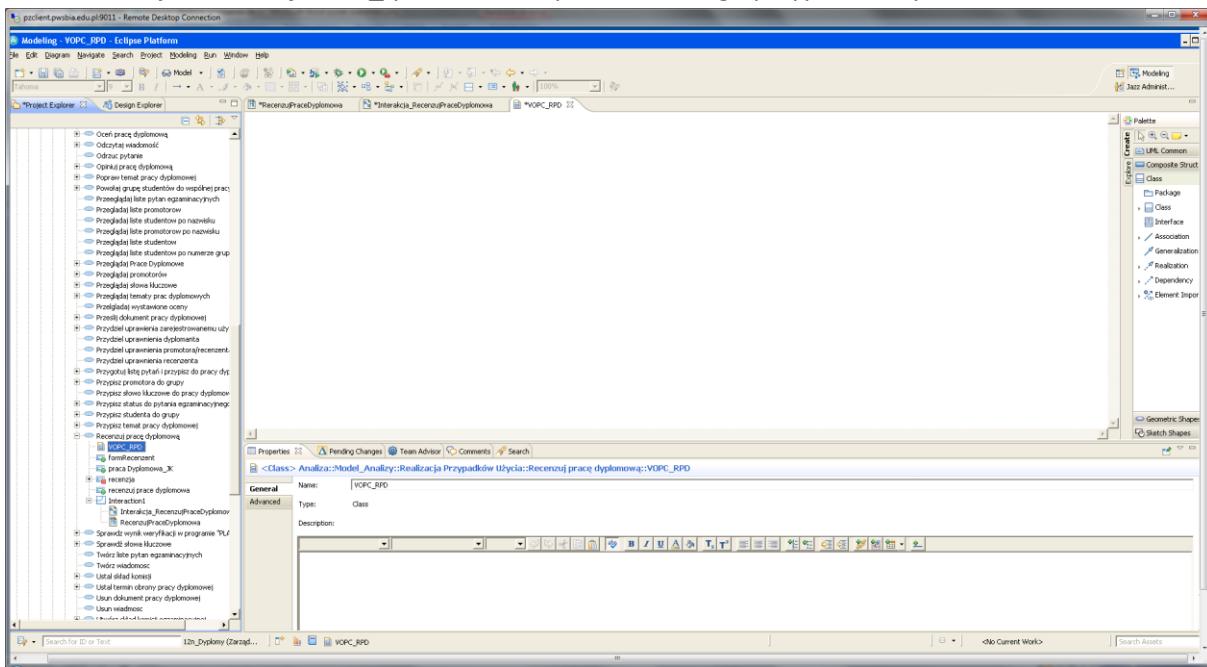


32. Tworzenie diagramu VOPC

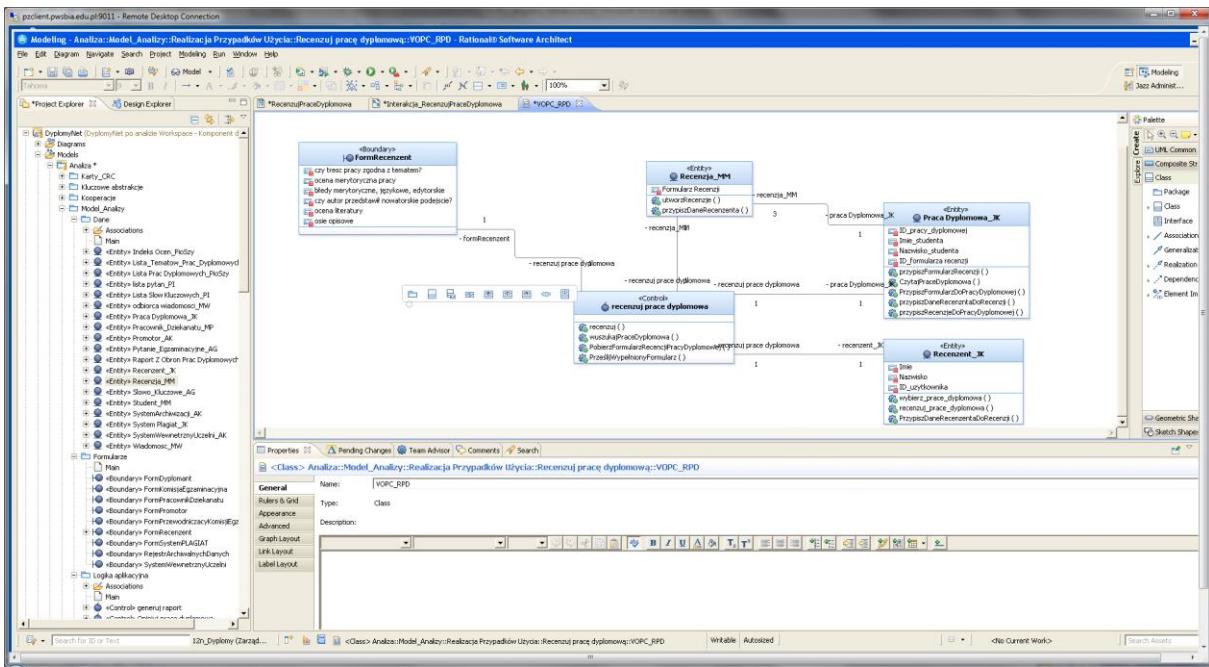
- ✓ Dodaj diagram klas do wybranego przypadku użycia:



- ✓ Nadaj mu nazwę VOPC_[pierwsze litery realizowanego przypadku użycia]



- ✓ Pobierz komplet klas wykorzystanych do reprezentowania przypadku użycia (te same klasy, których używałeś do realizacji diagramu interakcji) i przeciągnij je na nowopowstałą kartę.



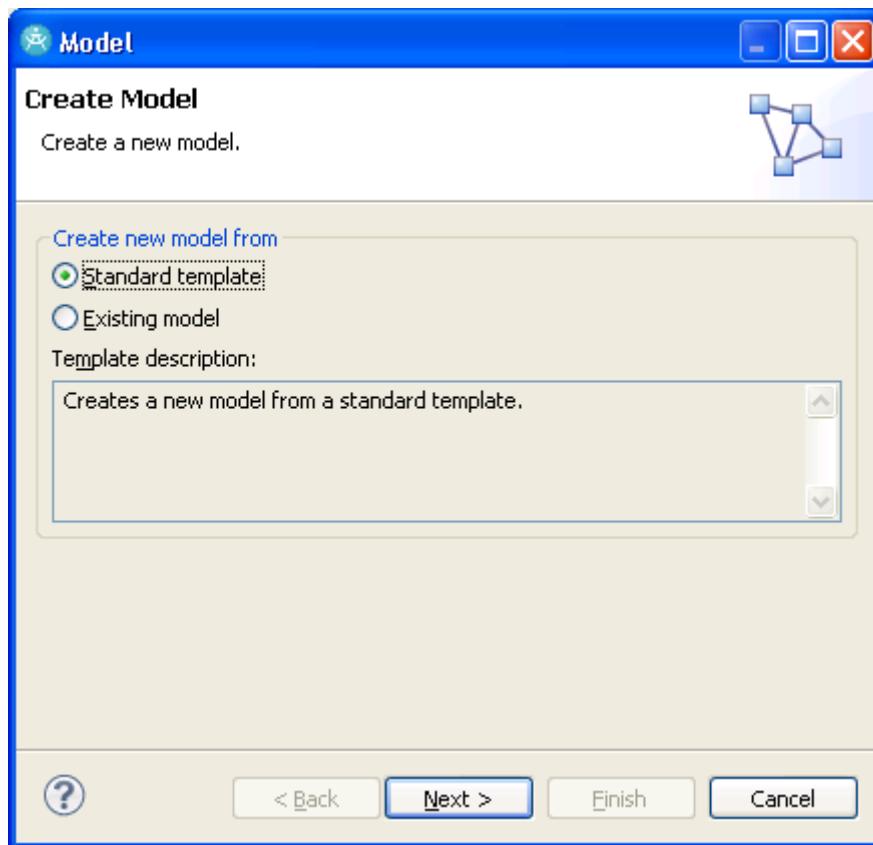
- ✓ dodaj asocjacje pomiędzy klasami
- ✓ ustal krotności
- ✓ W klasie Boundary, wpisz atrybuty poprzez ADD Atribute- powinny być to dane, które będą wprowadzane przez użytkownika.
- ✓ W klasie entity dodaj atrybuty- to elementy, które miałyby być przechowywane w bazie danych dla tego elementu.

Diagram VOPC GOTOWY! XD

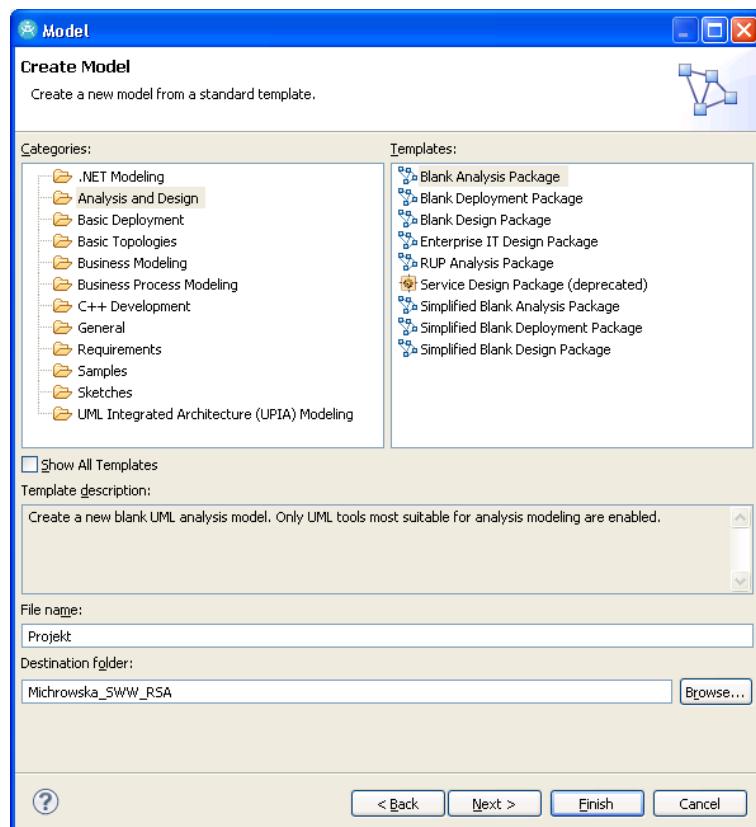
33. Diagram komponentów

W celu przejścia do tworzenia diagramu należy utworzyć nowy model.

- ✓ Utwórz nowy model w pakiecie Modele, nadaj mu nazwę Projekt.

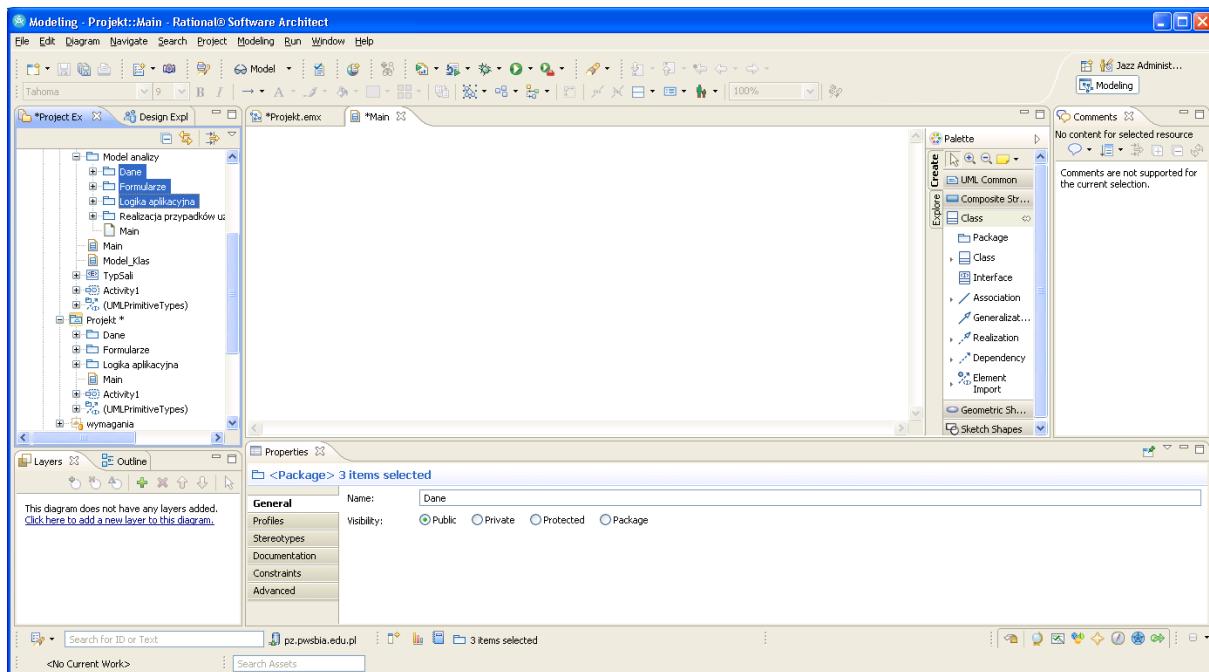


- ✓ Wybierz Analysis and Design
- ✓ Blank Analysis Package

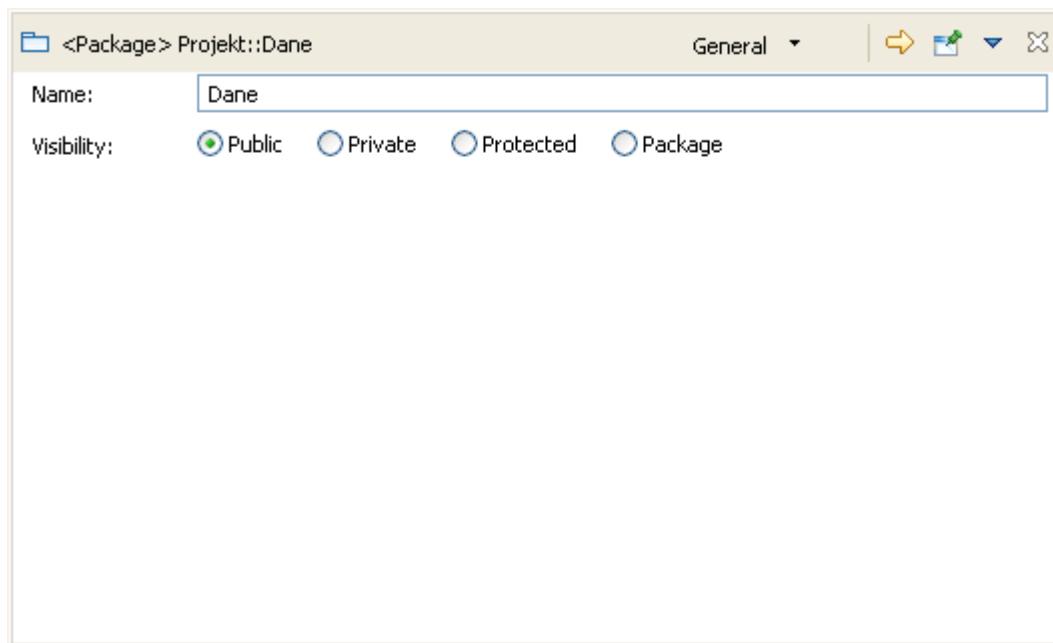


- ✓ Next
- ✓ Wybierz modele UML
- ✓ Przenieś pakiety **dane, formularze i logika aplikacyjna** do pakietu z Projektem.

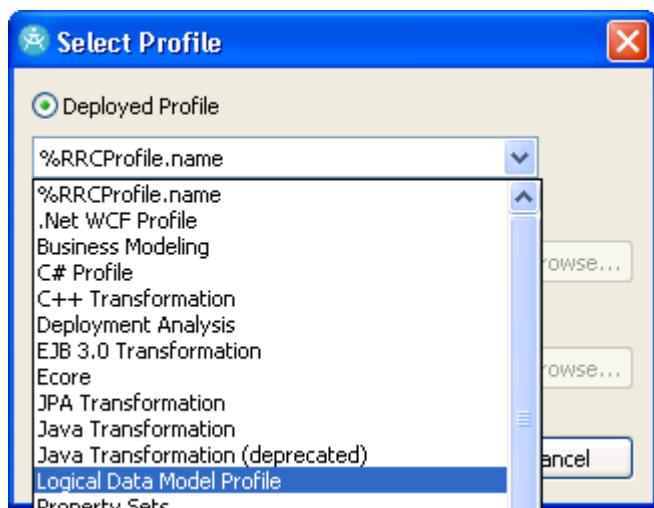
Pamiętaj! Wybieramy zawsze typy danych UML primitive.



We właściwościach folderu **Dane** wybierz **General>> profiles**



✓ Add Profile



Dodaj logika data model profile:

<Package> Projekt::Dane

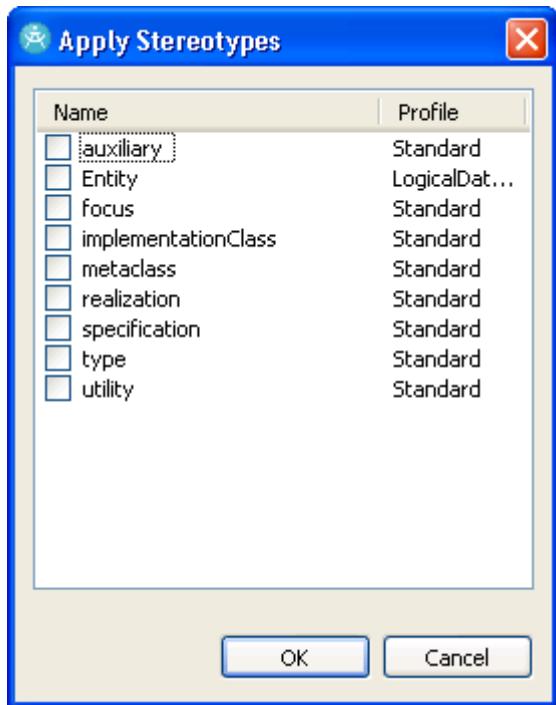
Profiles ▾

Applied Profiles:

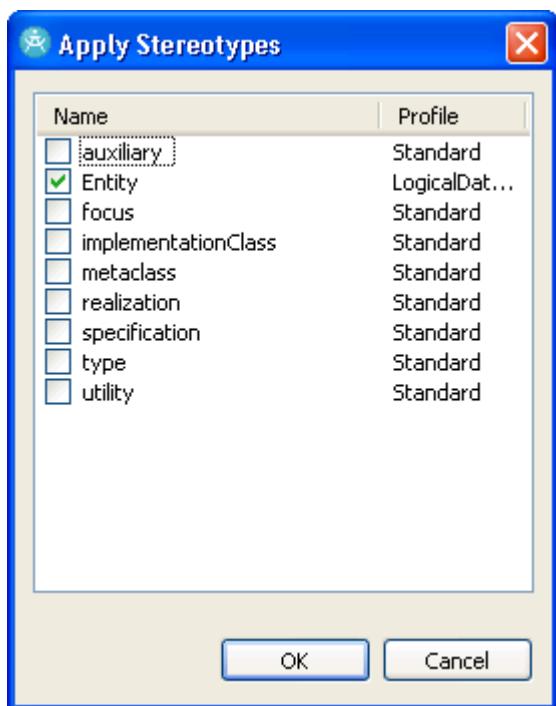
Name	Version	Release Label	Location
LogicalDataModel	93	1.1.8	C:\Program Files\IBM\SDPShared\plugins\com.ibm.xtools.t

Add Profile... Remove Profile

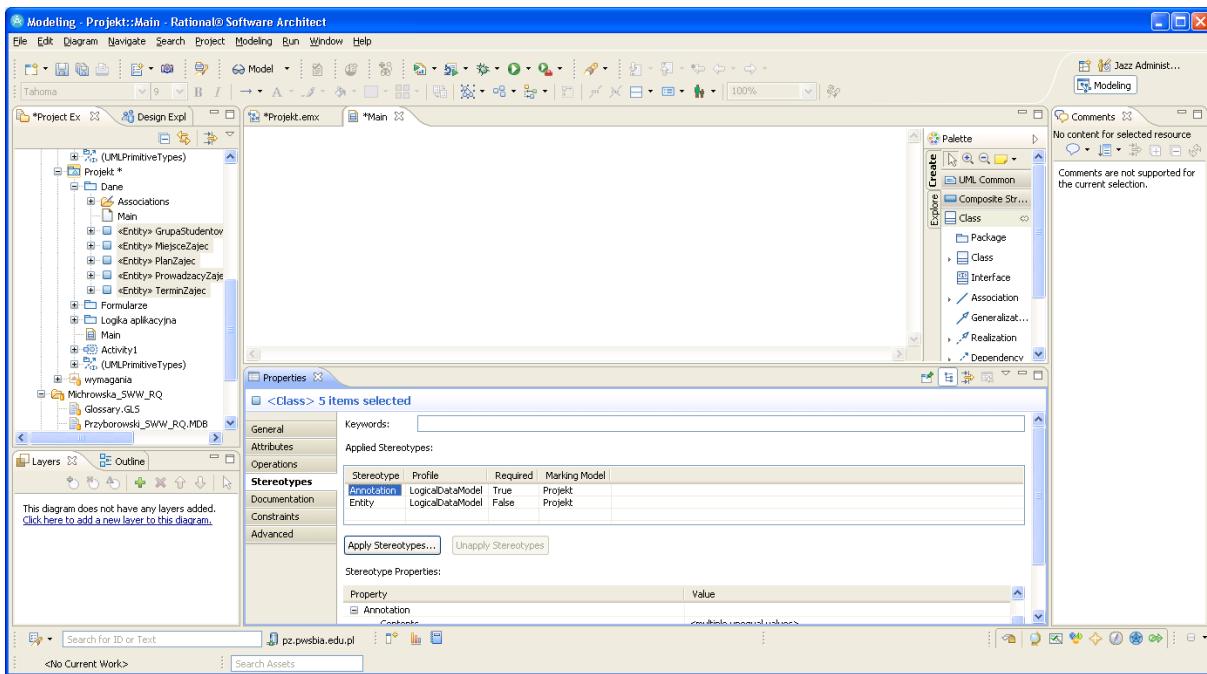
- ✓ Zmień stereotyp na wszystkich klasach w folderze dane w pakiecie Pojekt.



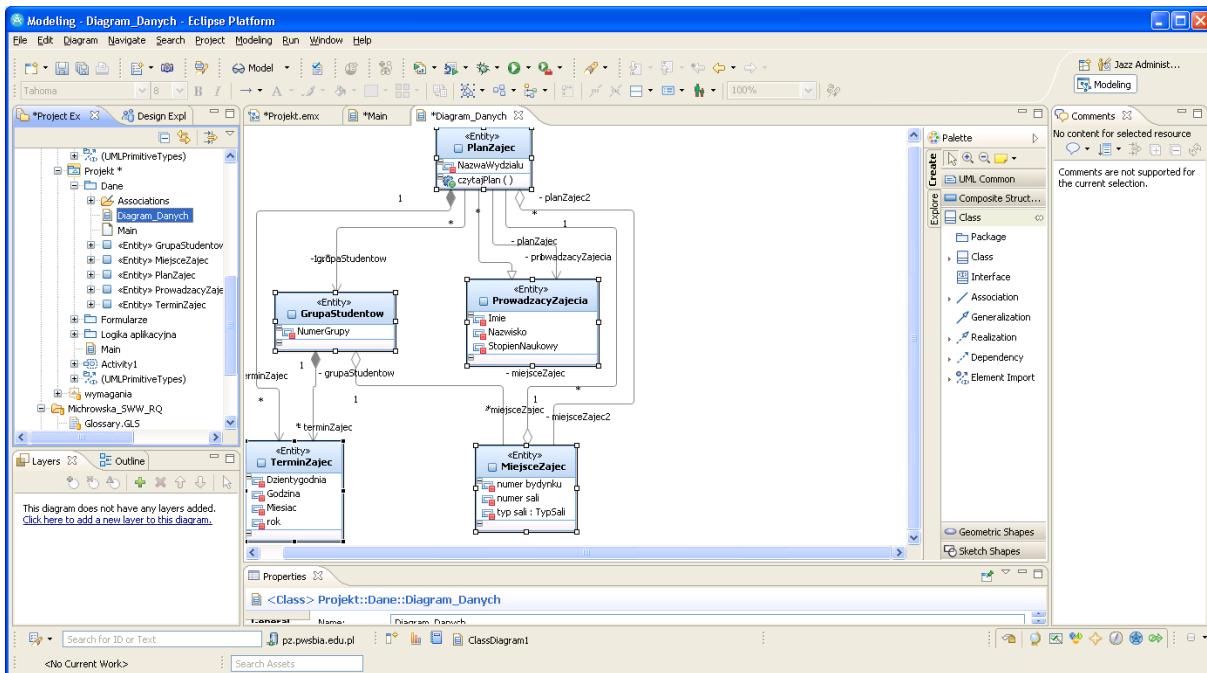
- ✓ na Entity (to nie są te same entity z modelowania klas)



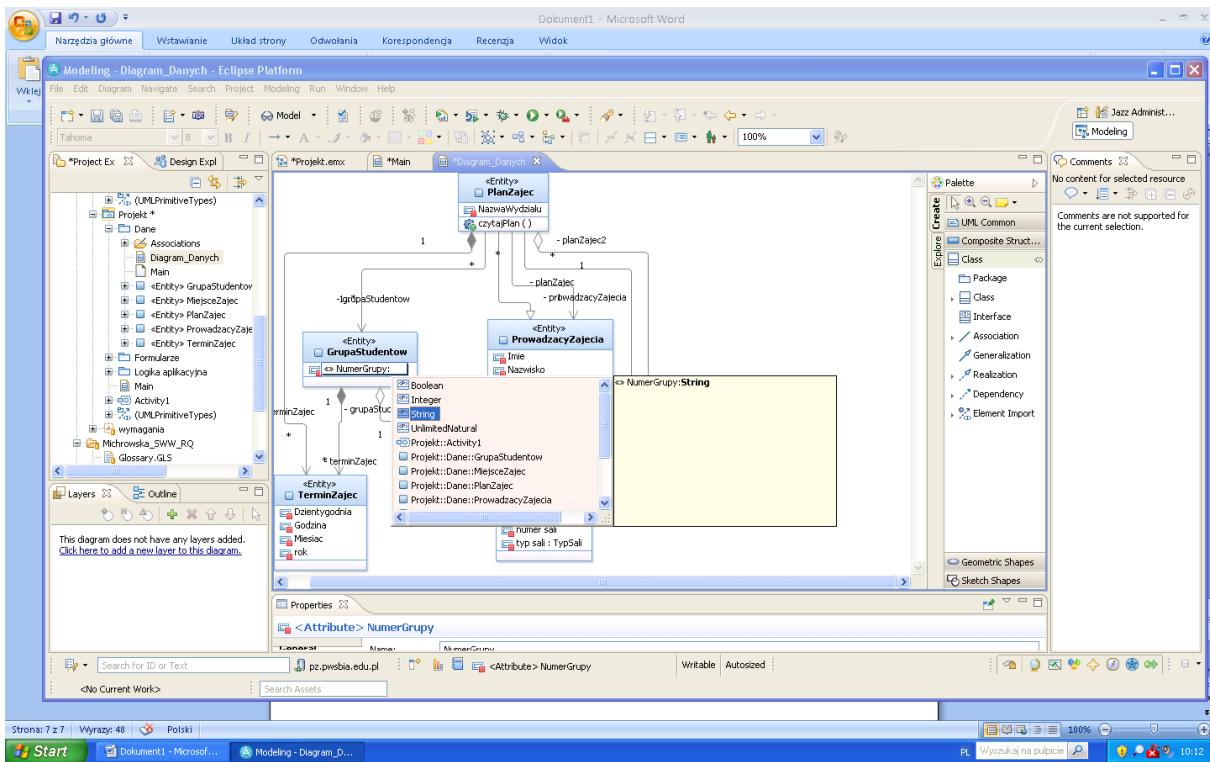
- ✓ Zostaw stereotypy jak są:



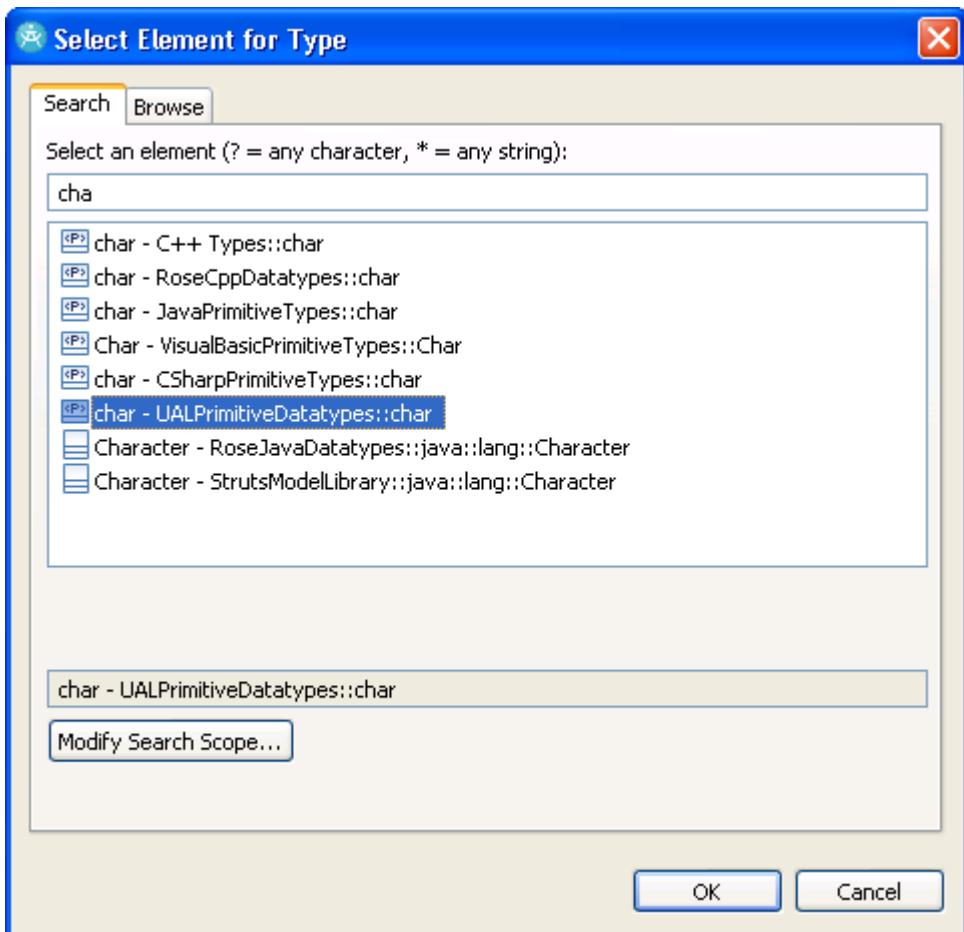
- ✓ Twórz diagram klas dla tego modelu. Nadaj mu nazwę Diagram_Danych
- ✓ Przeciagnij wszystkie dodane klasy na diagram. Jeśli wcześniej nie utworzyłeś relacji dla klas danych, to po przeciągnięciu nie będziesz miał relacji, utwórz je.



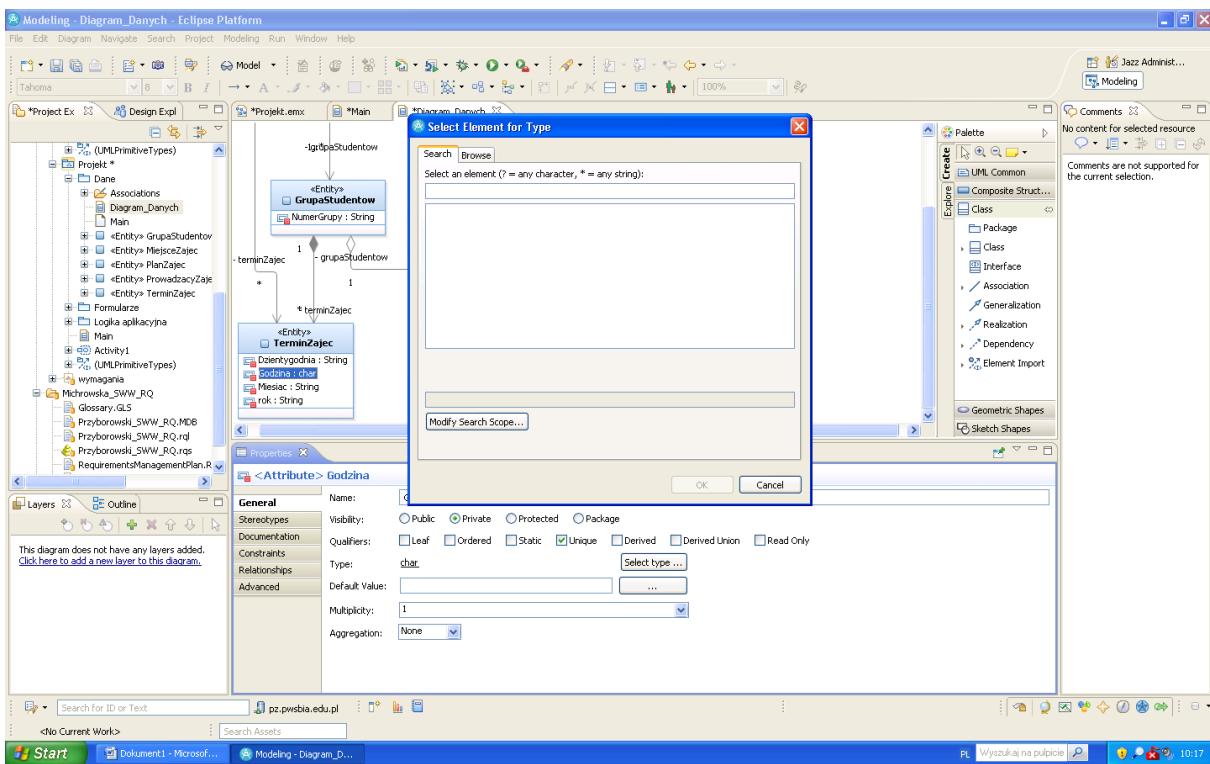
- ✓ Dodaj typy danych na diagramie, poprzez dodanie ':' po nazwie elementu, lub w propertiesie.



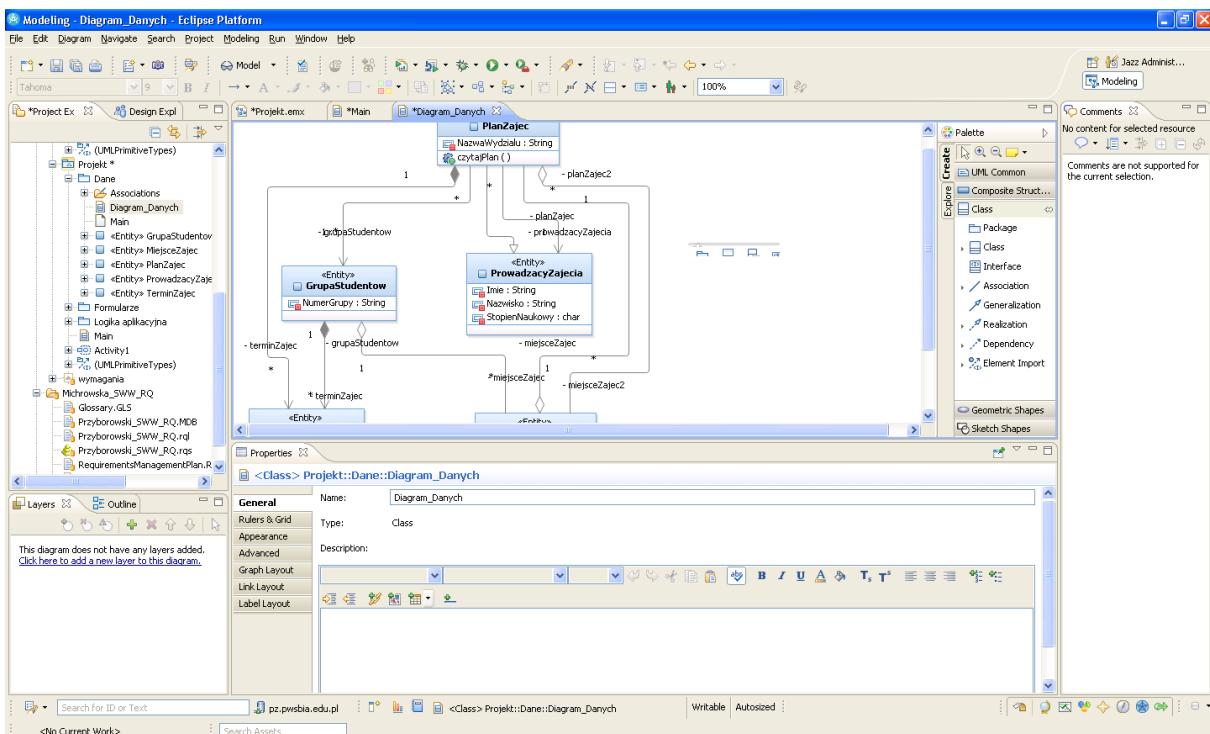
- ✓ Zawsze wybieramy UML primitive



- ✓ W Properties General>>> Select Type.



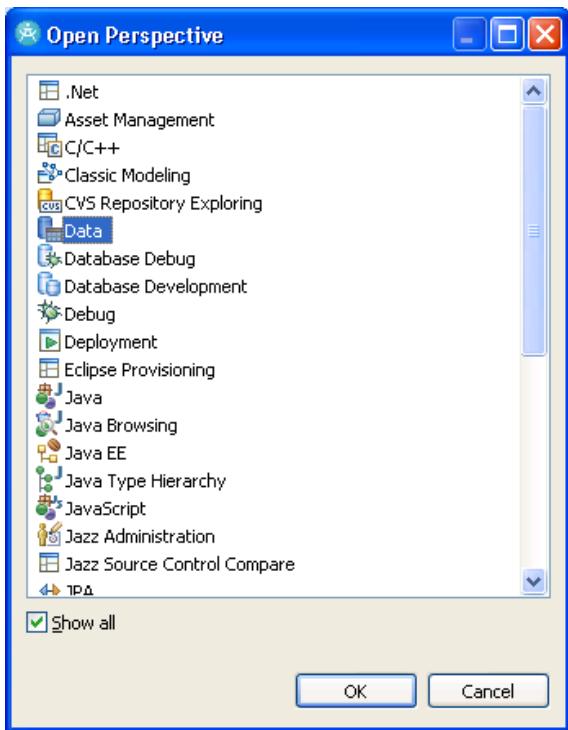
Wszystkie typy danych zostały uzupełnione.



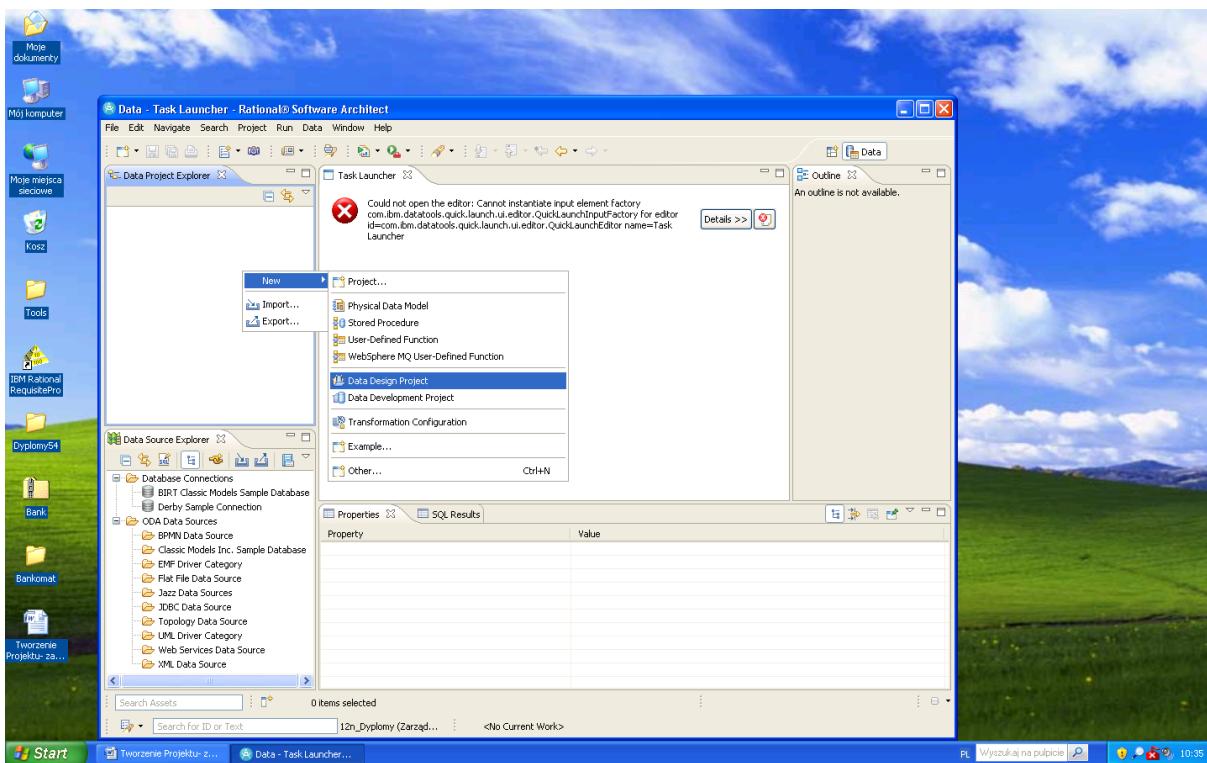
Przed zmianą perspektywy koniecznie wyślij zmiany na serwer. Nie wysłanie ich może spowodować utratę wszystkich wprowadzonych elementów przez dodanie modelu danych-taką sytuację mieliśmy na zajęciach, więc lepiej się zabezpieczyć.

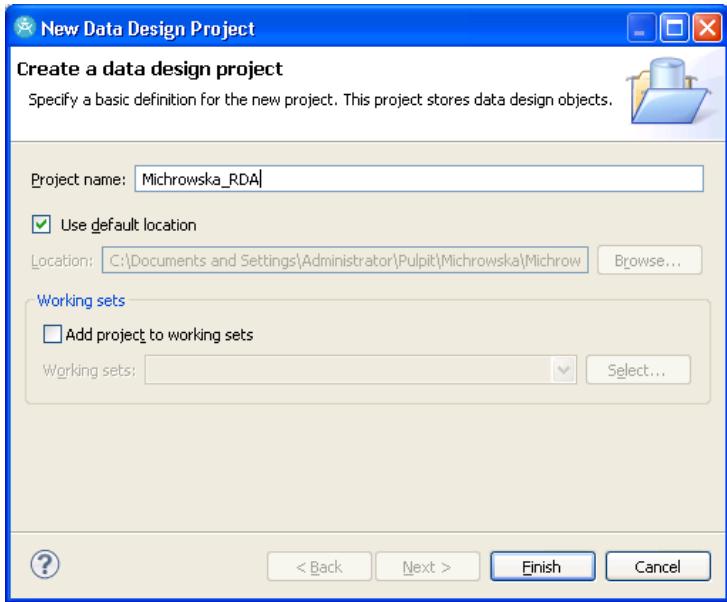
34. Komponent bazy danych

- ✓ Zmień perspektywę na Data:



- ✓ W oknie **Data Project Explorer** twórz nowy Project na bazie szablonu i nadaj mu określoną nazwę.
- ✓ Wybierz Data Design Project.



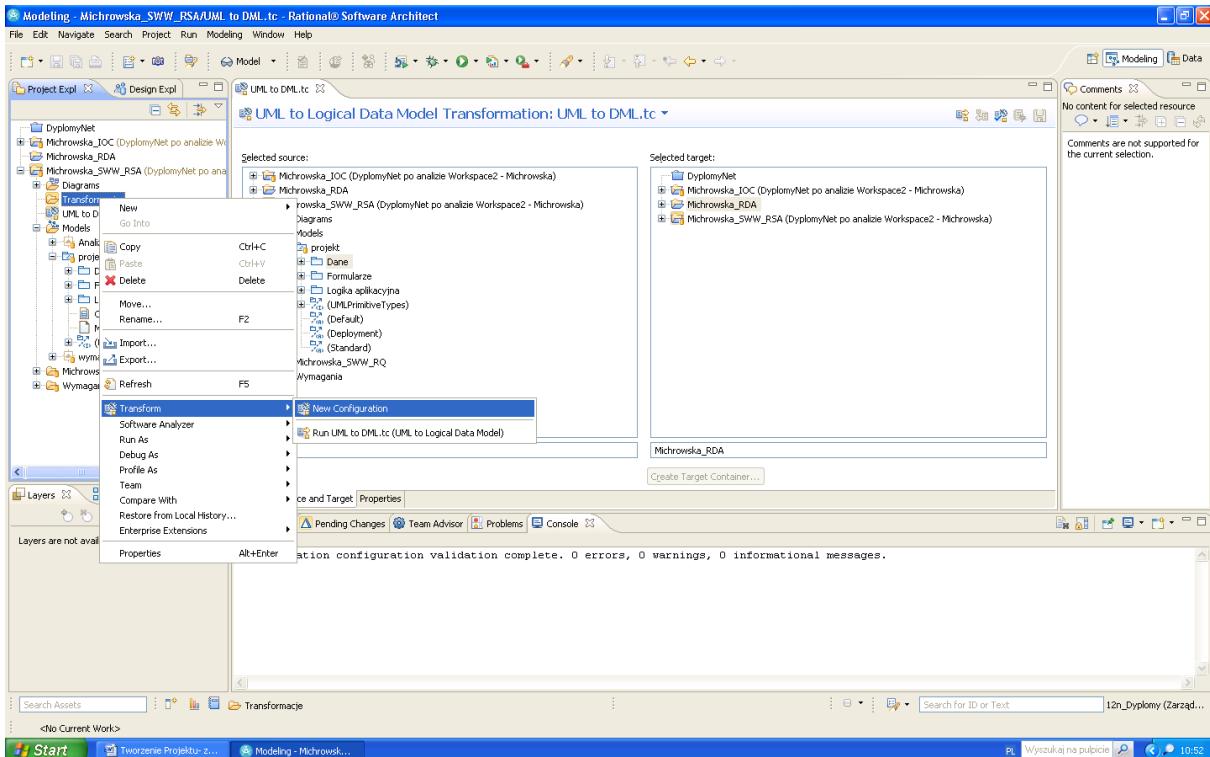


✓ Finish.

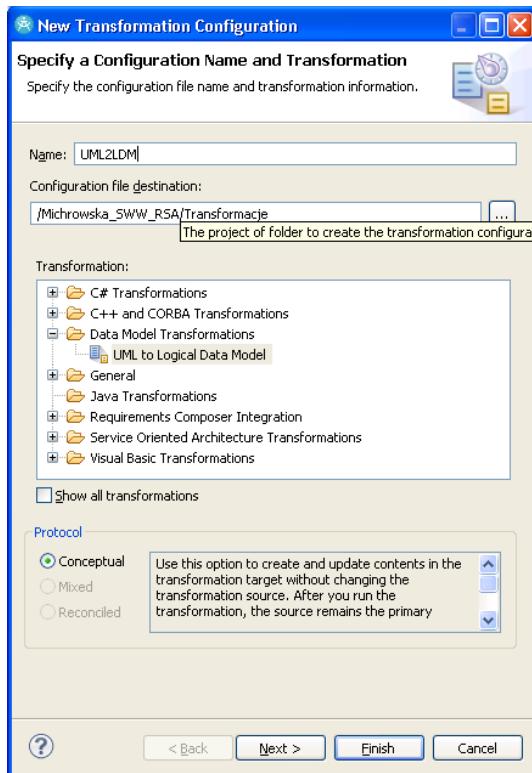
- ✓ Zmień perspektywę na **modeling**.
- ✓ W pakiecie projektu Utwórz nowy folder i nadaj mu nazwę **Transformacje**.



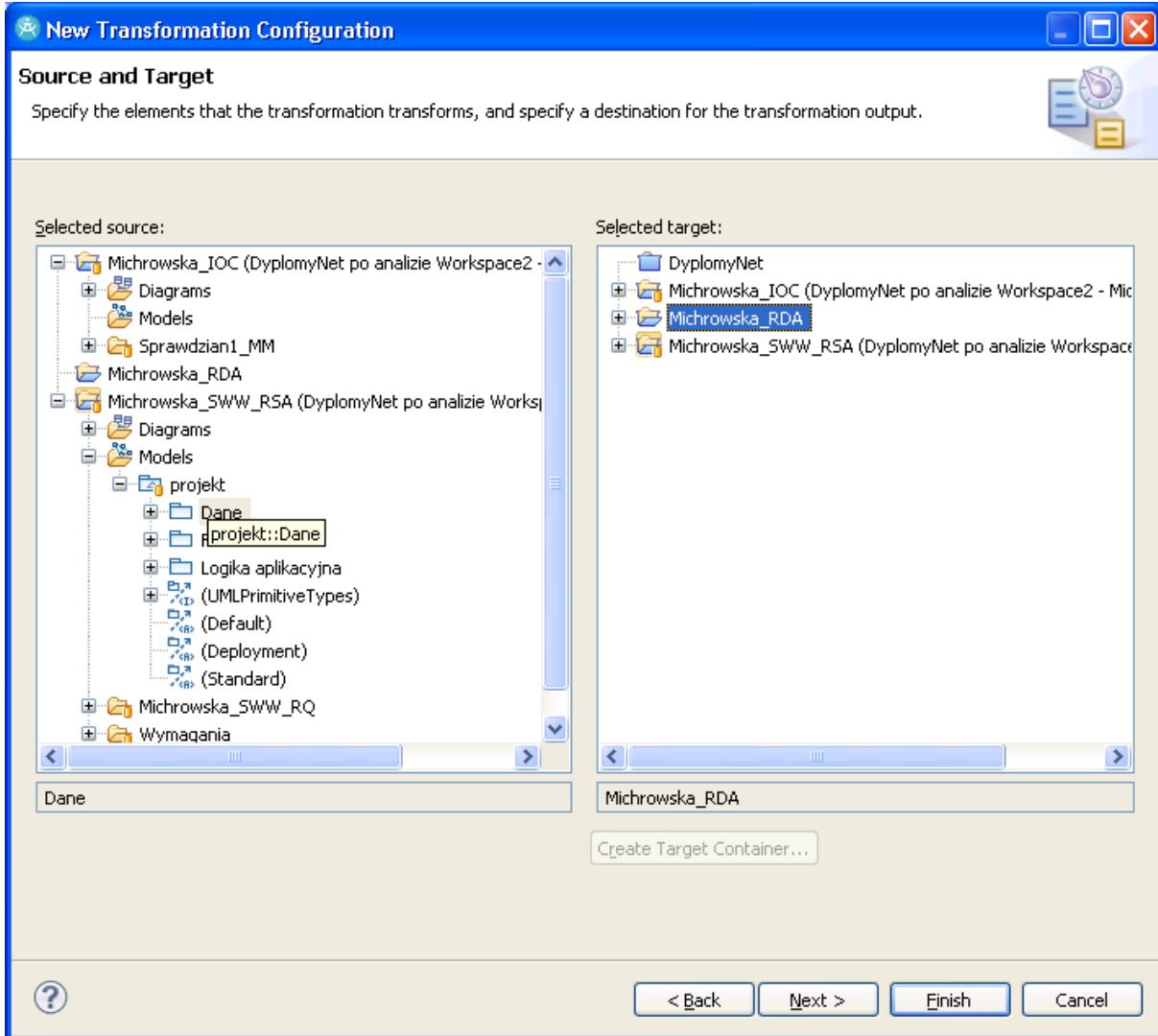
- ✓ Na folderze Transformacje wybierz opcje **Transform >> New configuration**

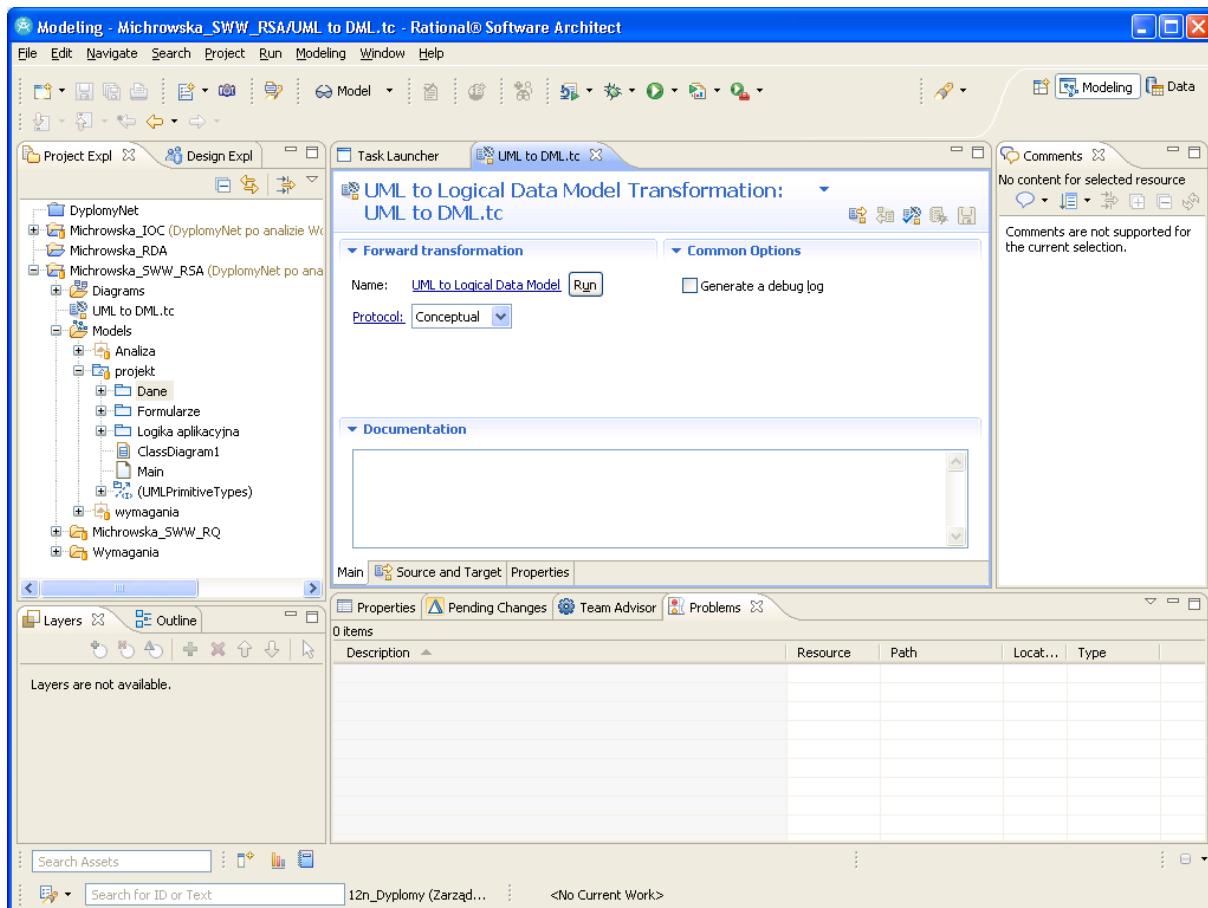


- ✓ Nadaj tej transformacji nazwę UML2LDM (Logical Data Model)



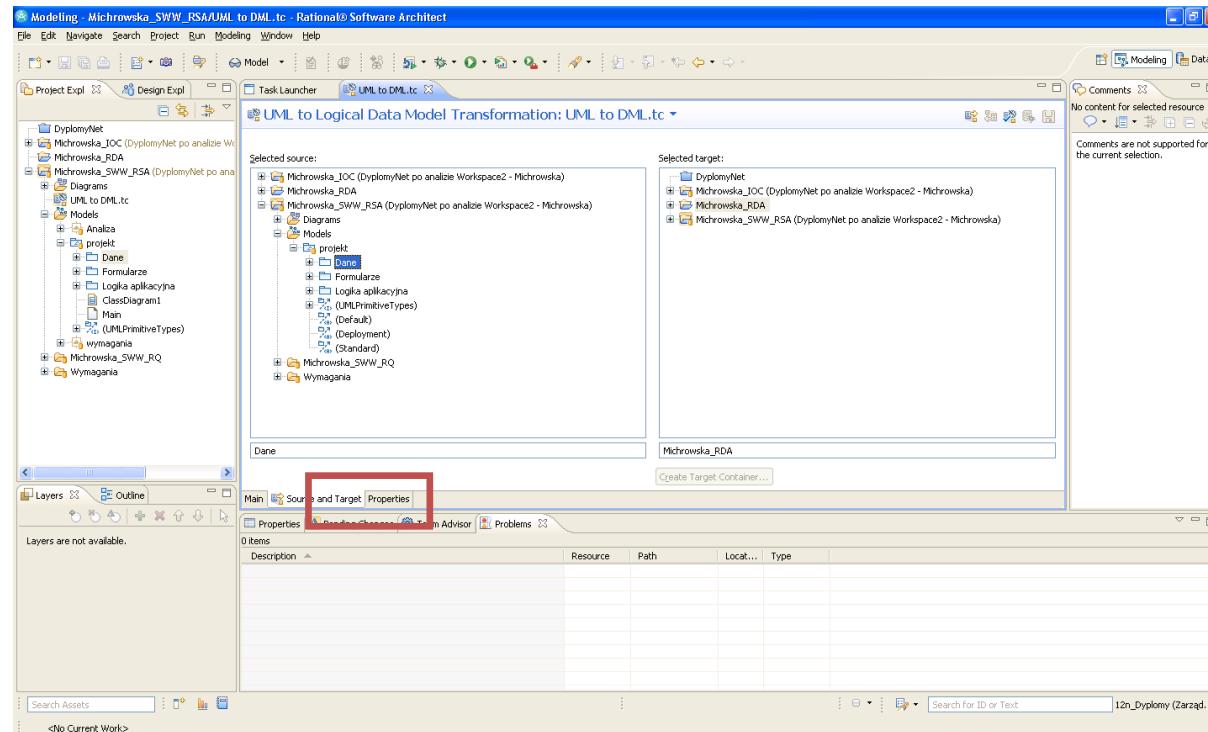
- ✓ Wskaż folder dane z paczki projekt jako source i target jako utworzony wcześniej model DataBase.



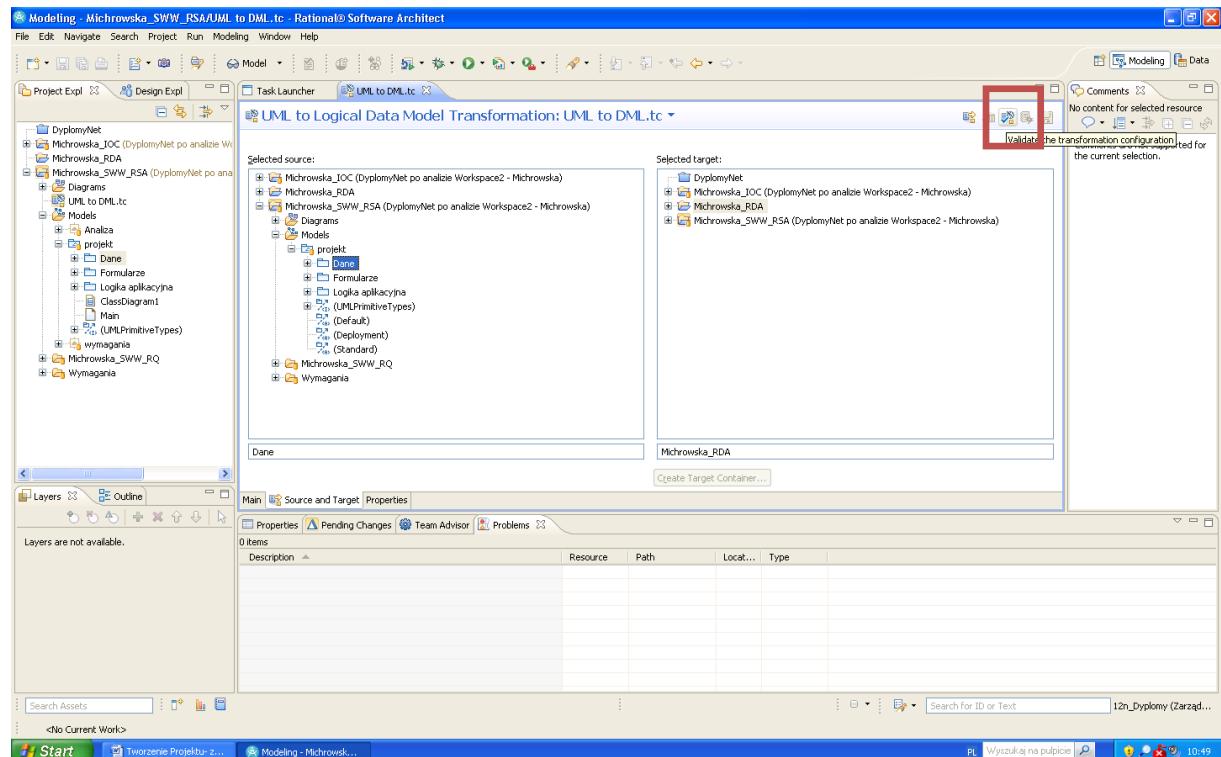


Transformacja została rozpoczęta.

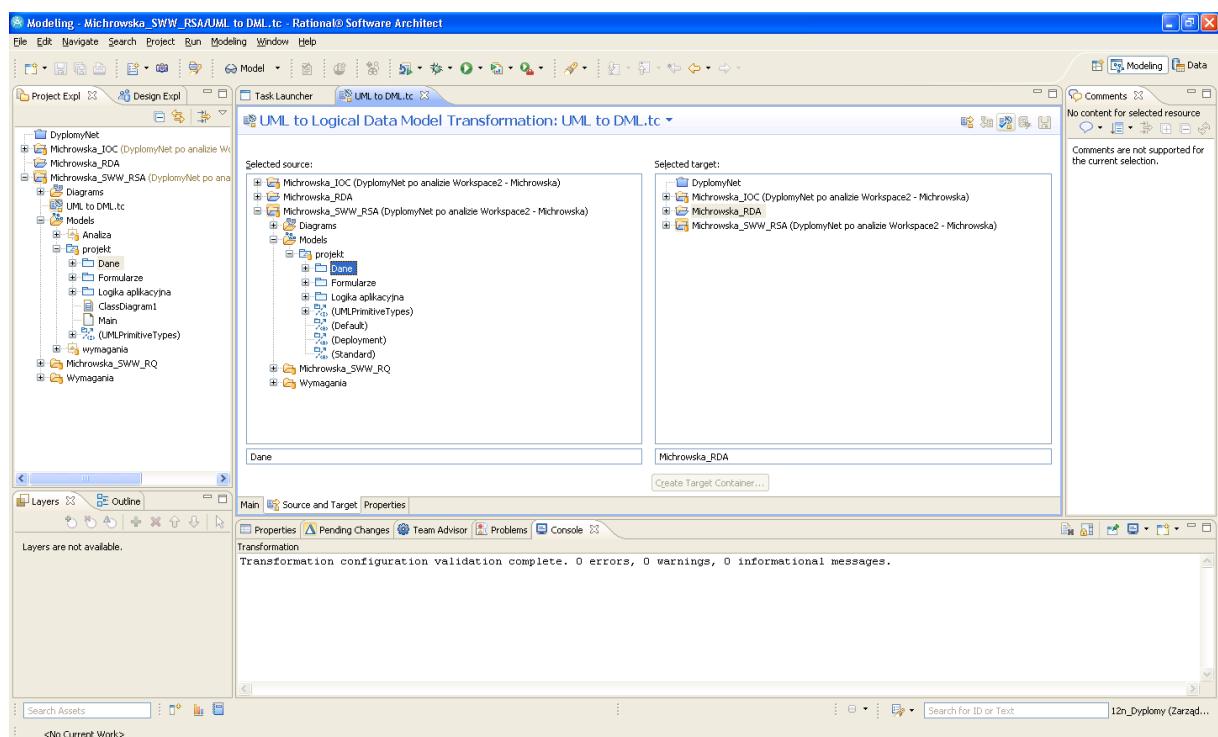
- ✓ Zmień kartę na dole na Source and target.



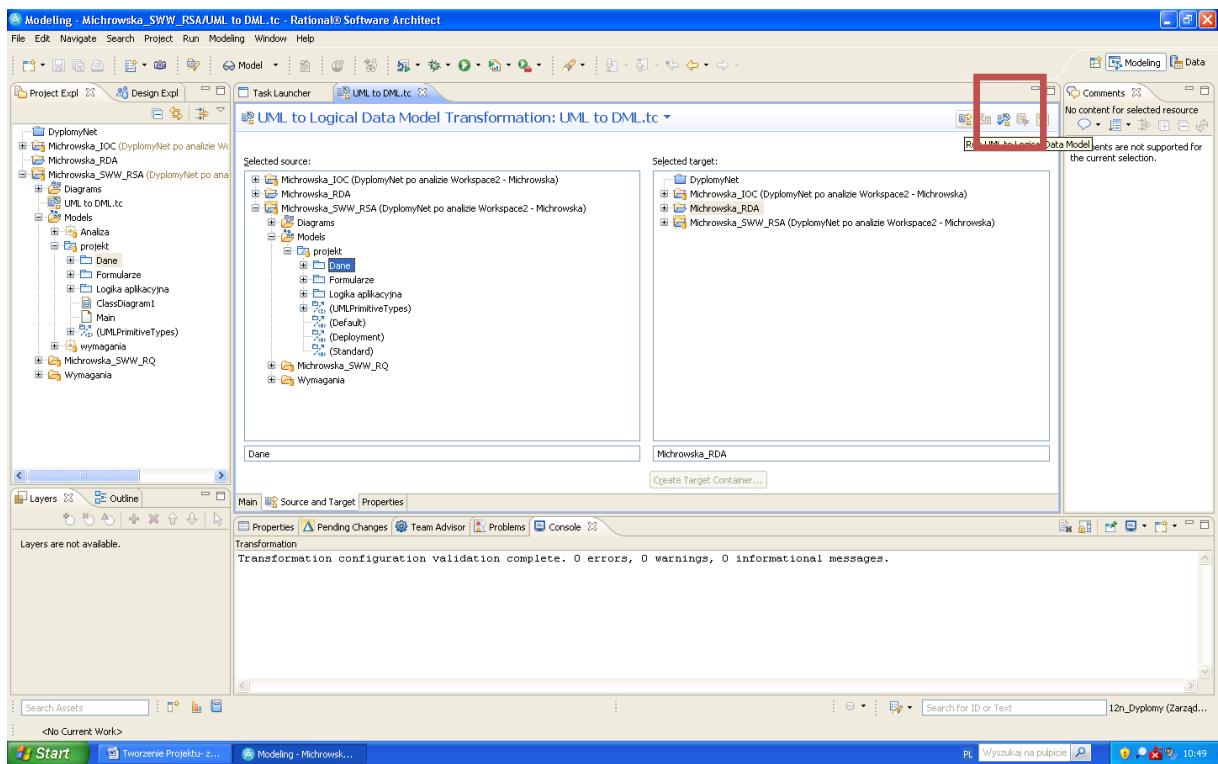
✓ Waliduj transformacje



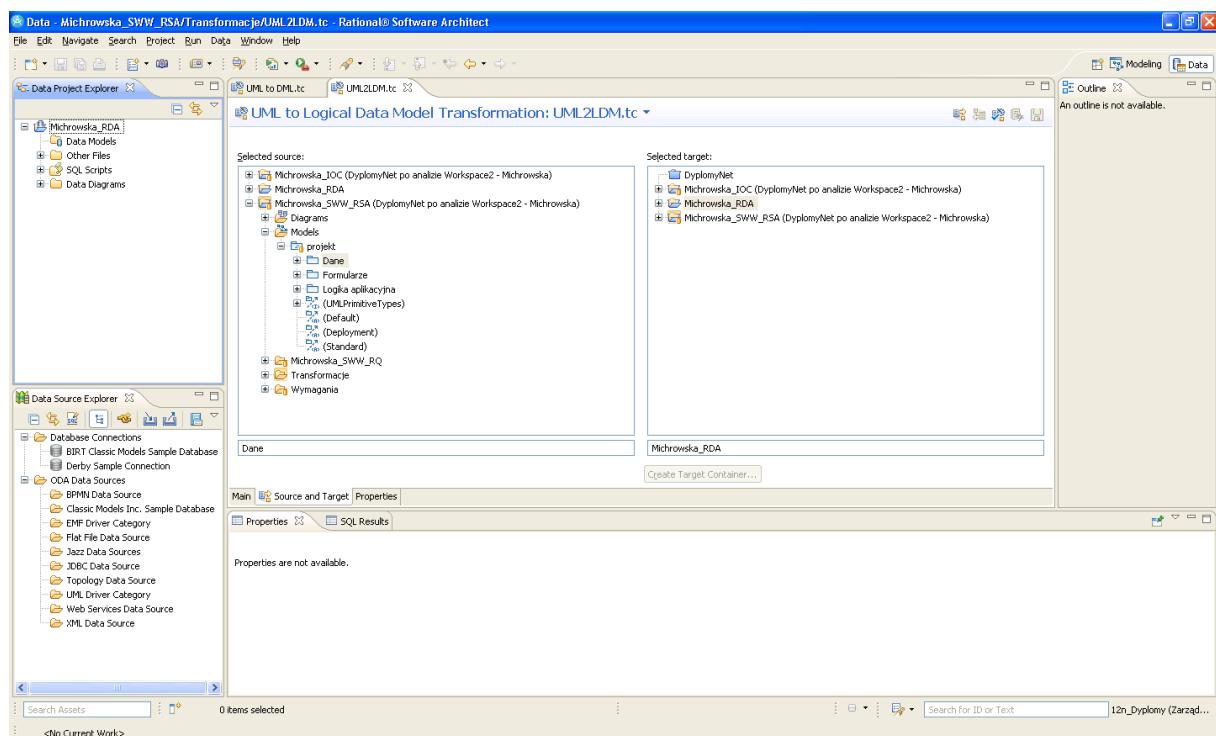
Wynik zobaczysz w okienku na dole, jeśli nie ma błędów przejdź do następnego kroku, jeśli masz błędy wróć do diagramu danych w pakiecie projekt i sprawdź czy we wszystkich nazwach nie masz gdzieś spacji lub polskiego znaku, to najczęstsze źródło błędu.



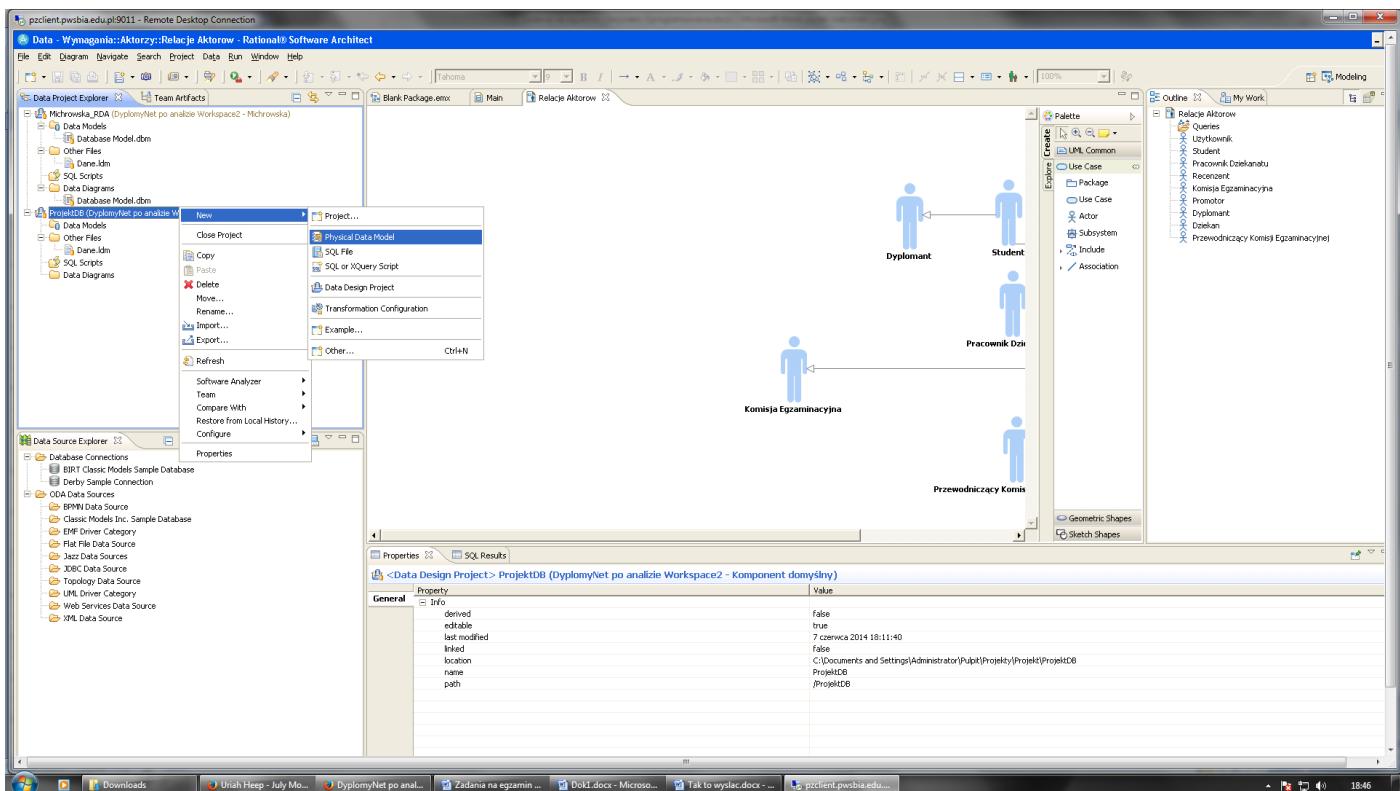
✓ Teraz uruchom transformację



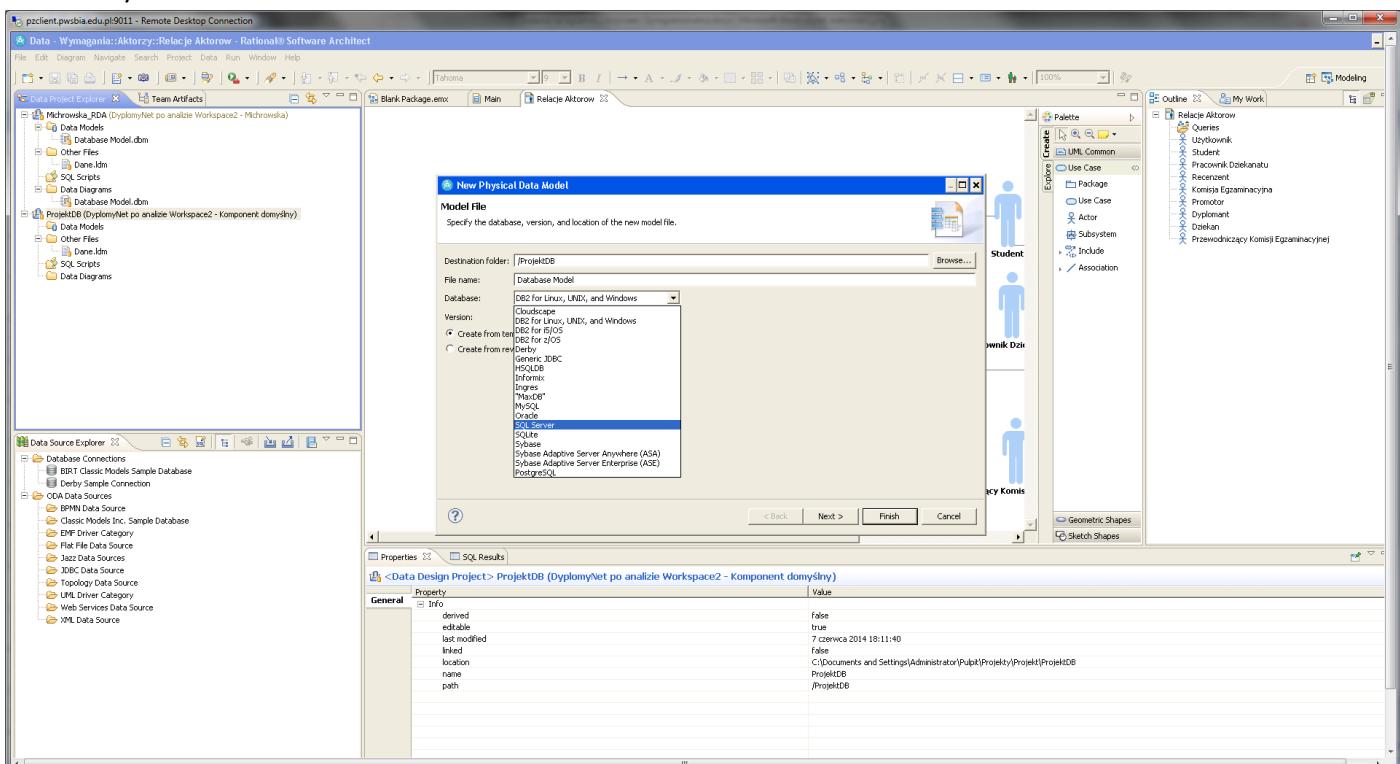
- ✓ Zmień perspektywę na DATA i sprawdź czy projekt został utworzony:



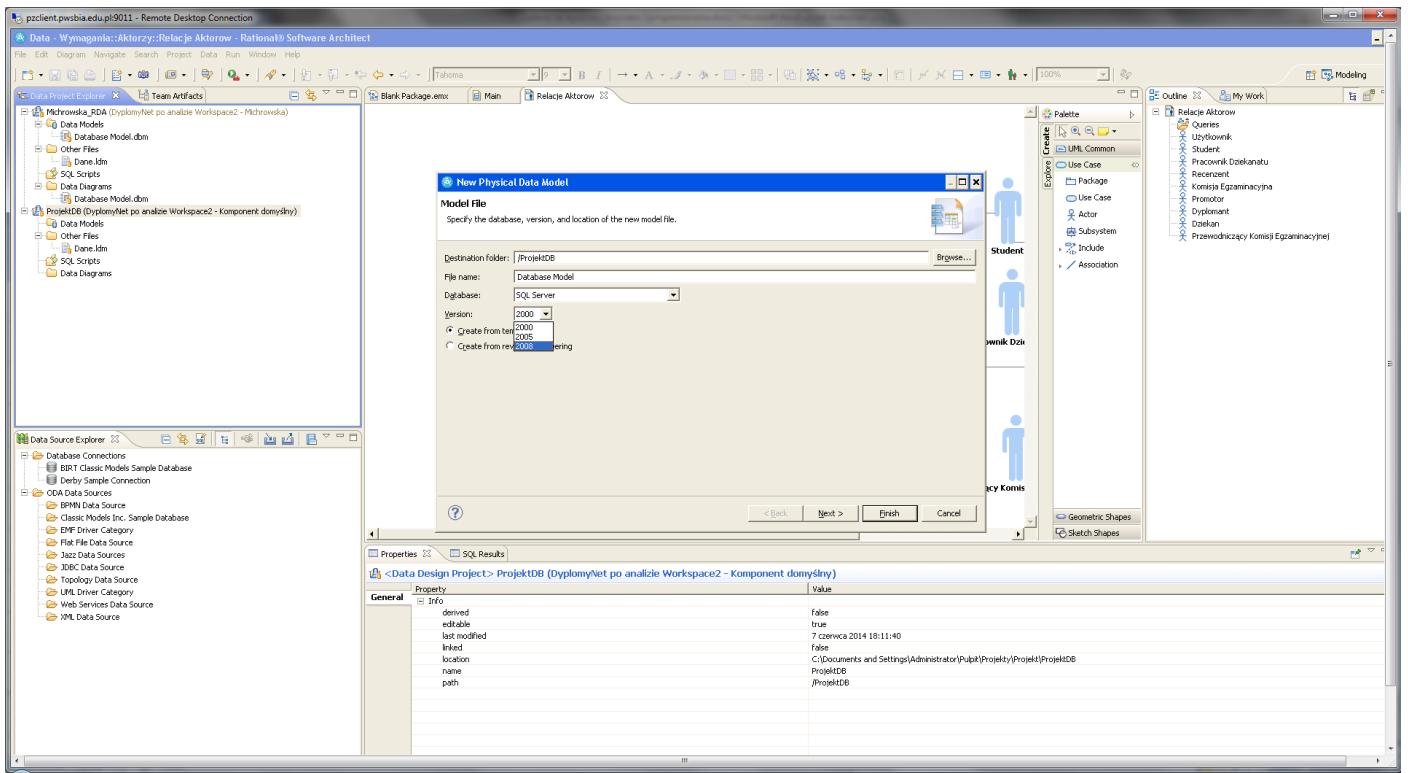
- ✓ Następnie na utworzonym projekcie wybierz prawym New >> Physical Data Model



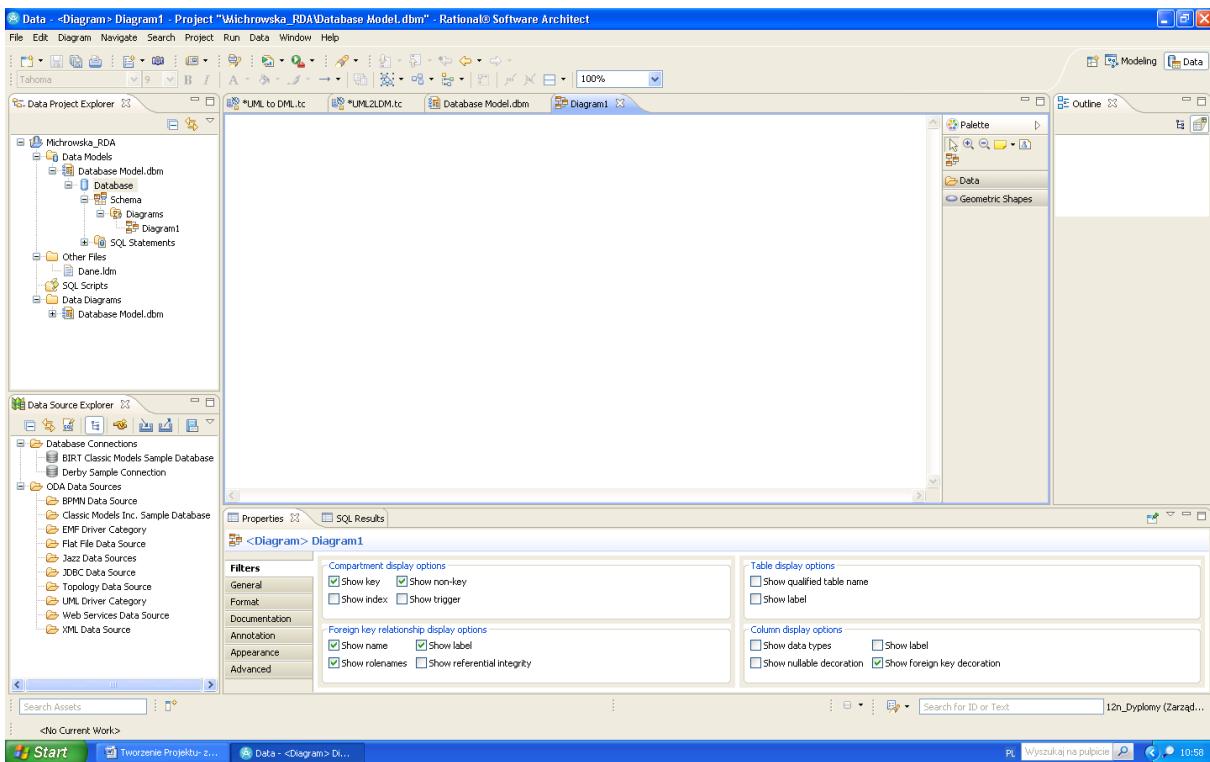
- ✓ Wybierz SQL Server



- ✓ oraz Version



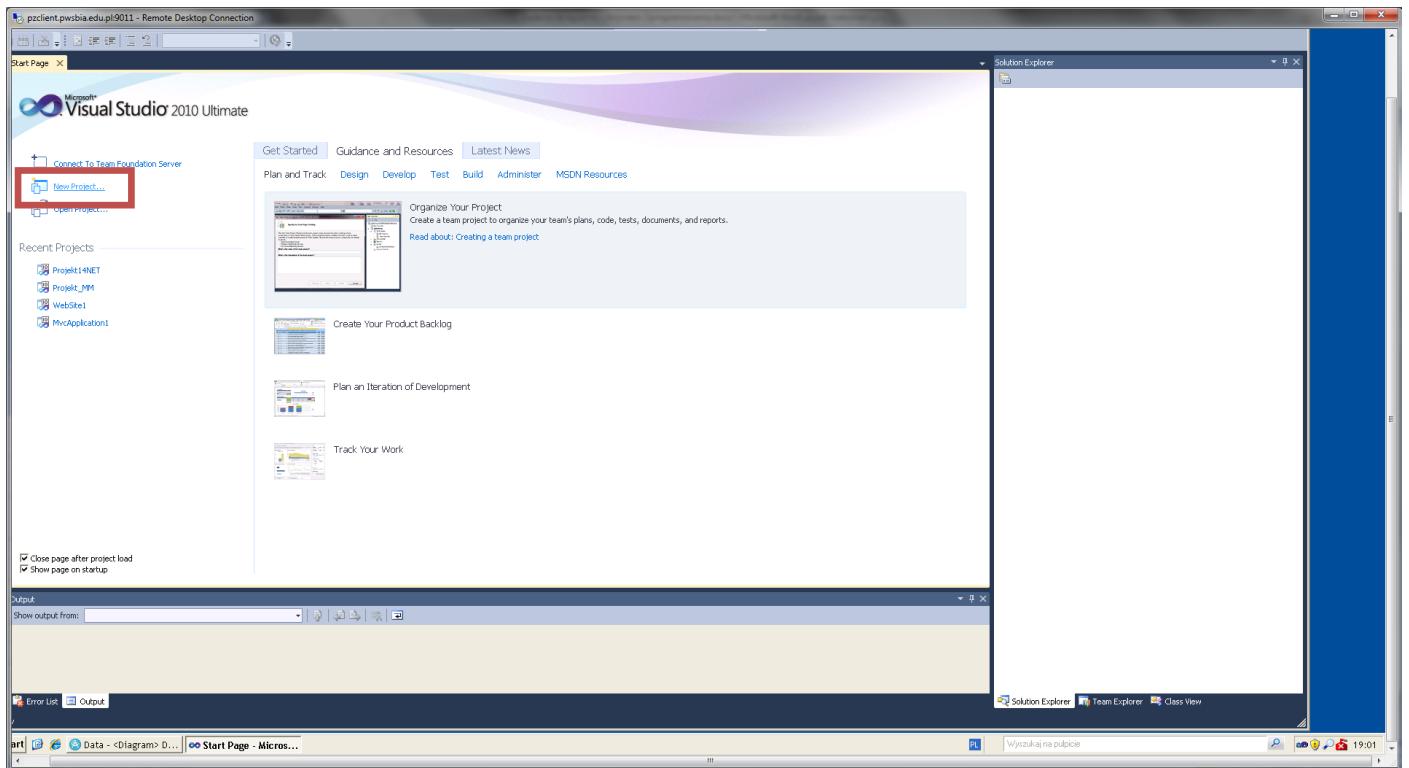
- ✓ Finish
- ✓ W ten sposób otrzymałeś nowy model.



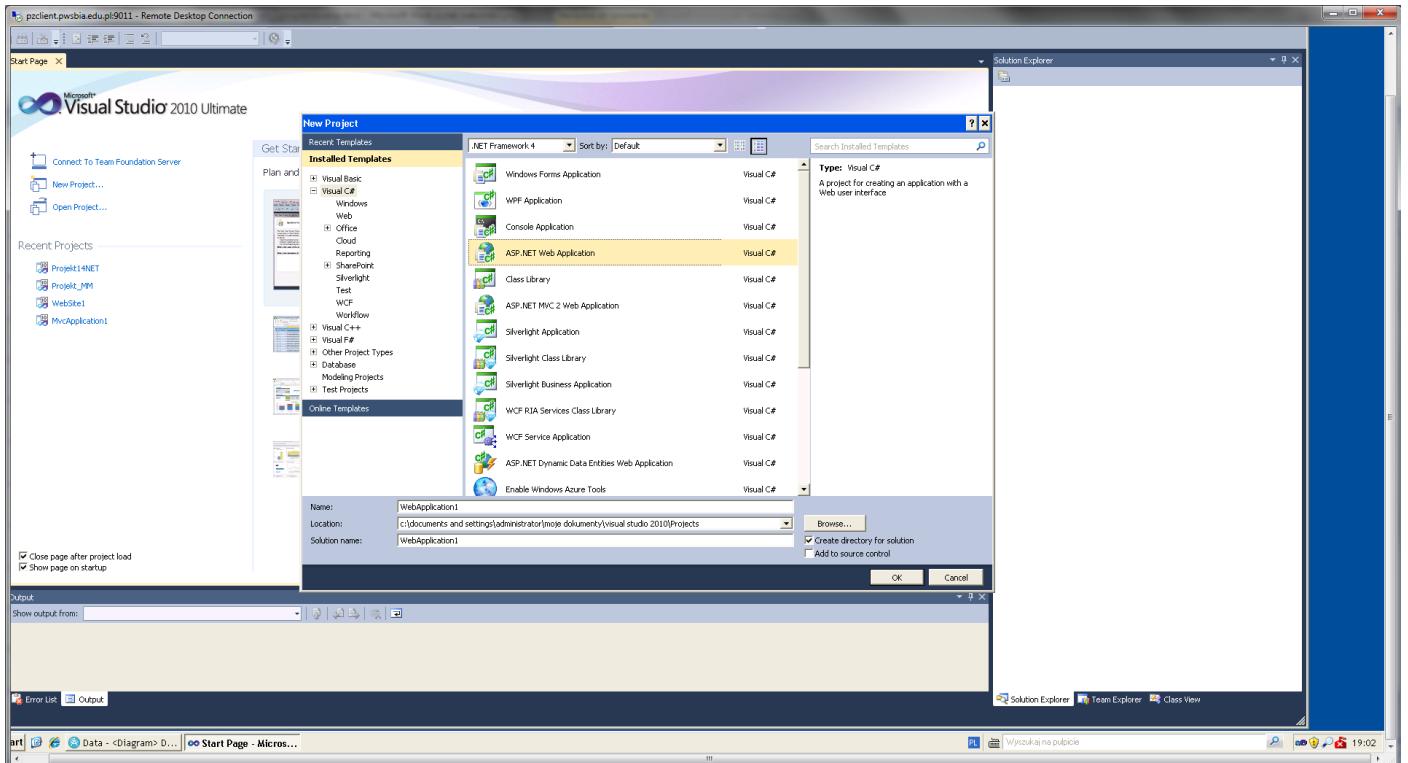
Jeśli nie udało się prawidłowo wykonać tego kroku, nic nie stoi na przeszkodzie, aby przejść do generowania kodu.

35. Mapowanie klas i wygenerowanie skryptu SQL

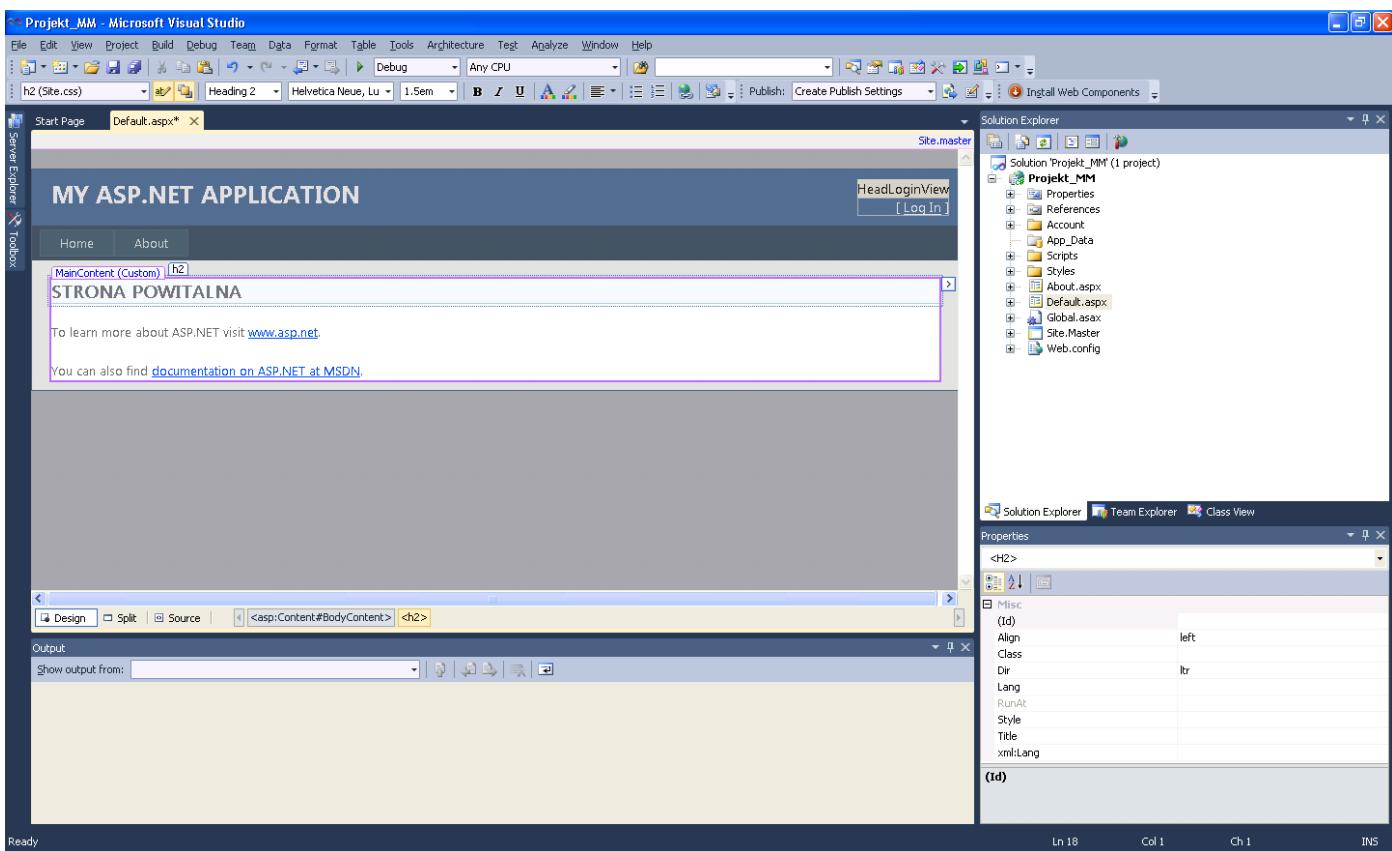
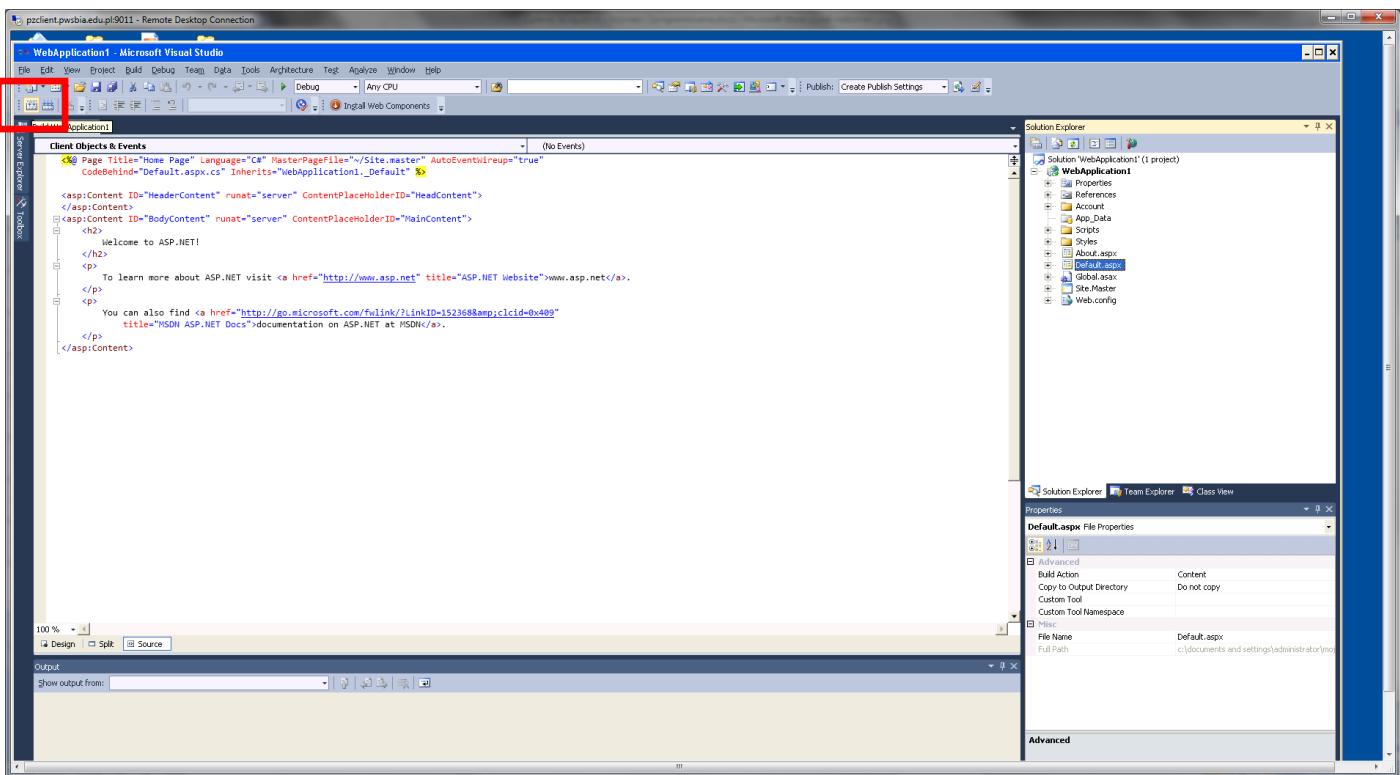
- ✓ Utwórz projekt w Visual Studio



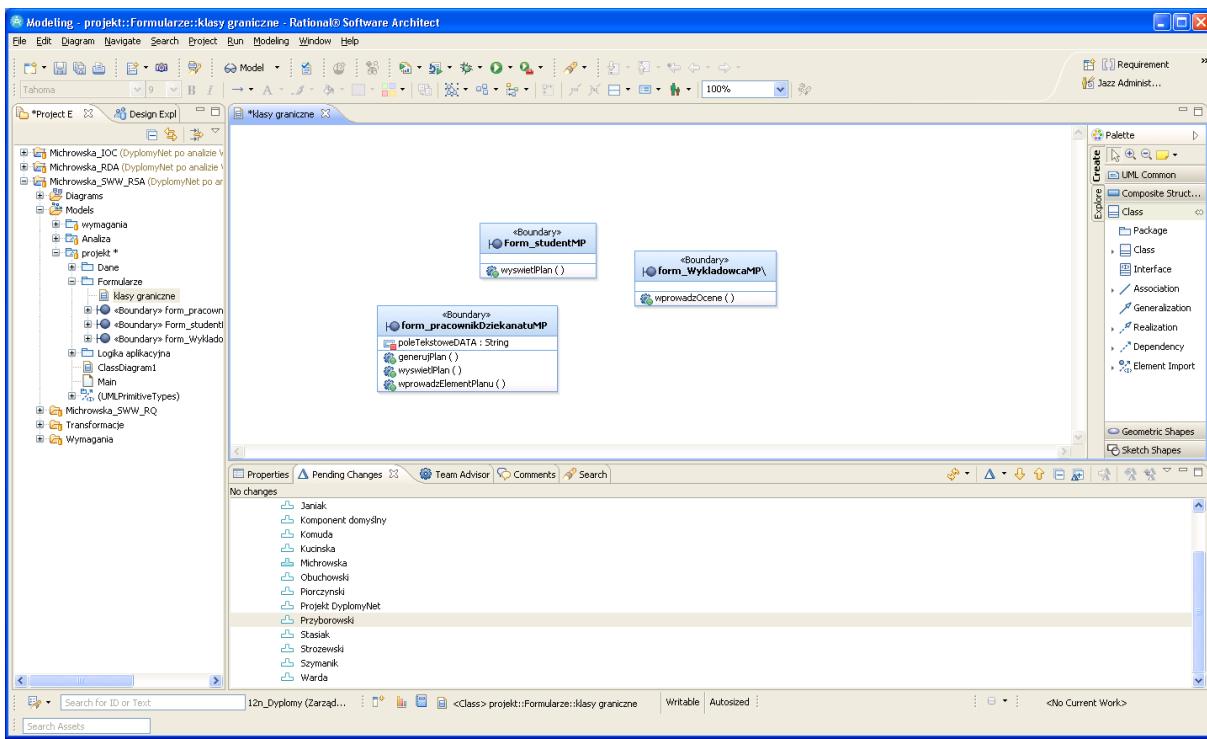
- ✓ Zaznacz Visual C# i ASP .NET Web Application nadaj nazwę i podaj lokalizację, gdzie ma być tworzony project, żebyś mógł go łatwo znaleźć.



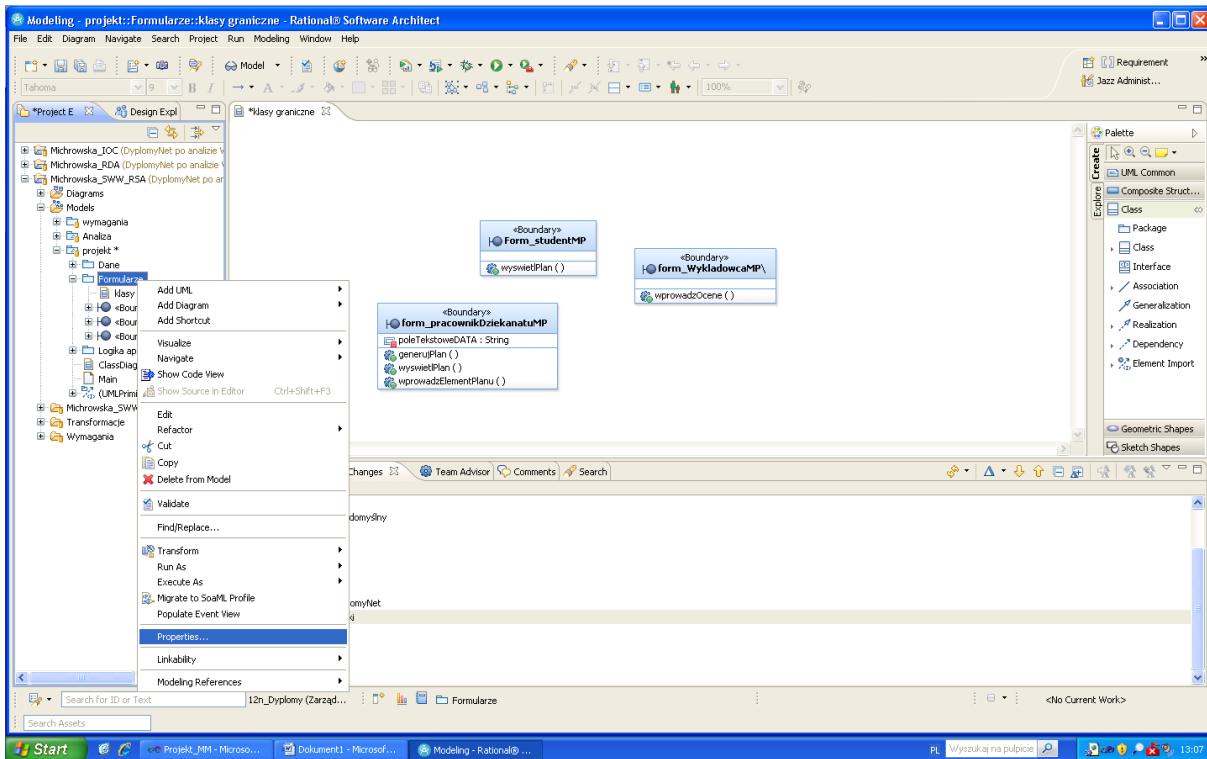
- ✓ OK.
- ✓ Zbuduj Aplikację i Solucję



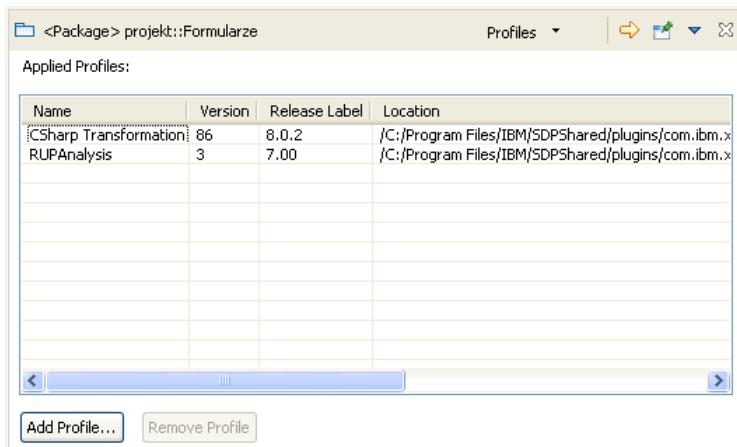
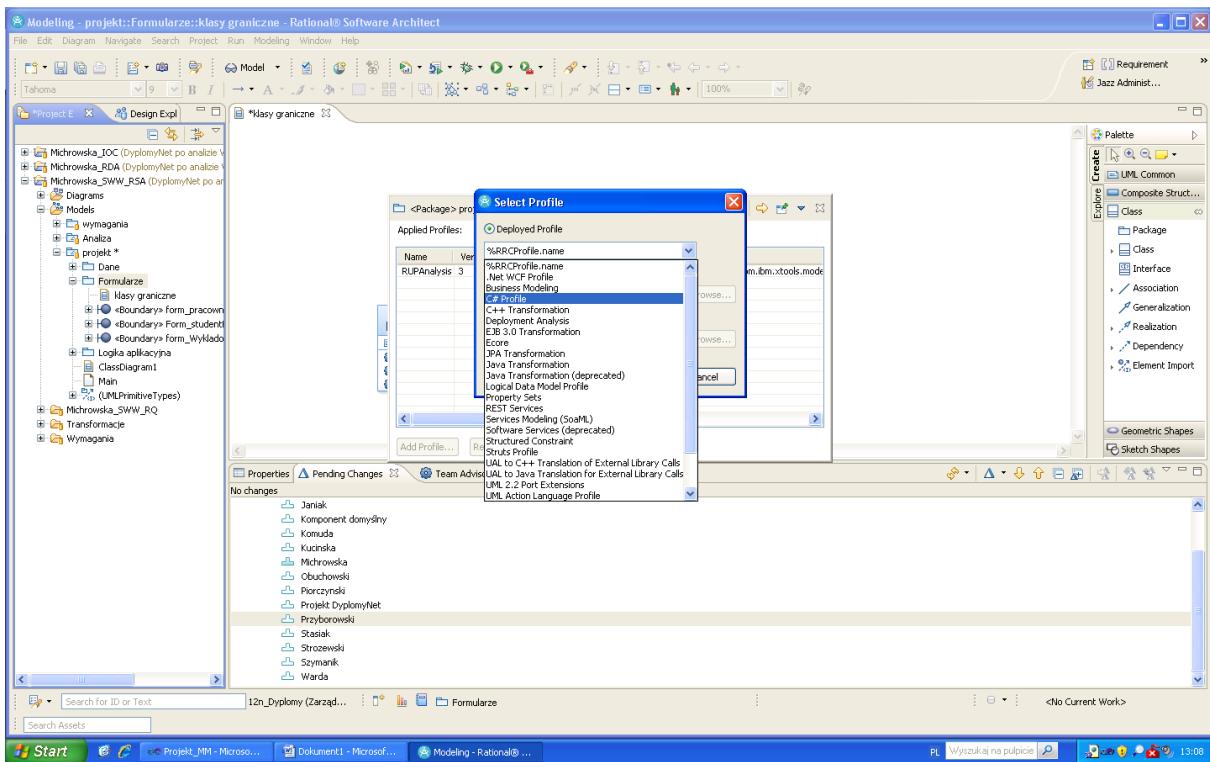
- ✓ Przejdź do RSA nie zamkaj Visual Studio.
- ✓ W RSA przekształć warstwę formularzy, tworząc diagram klas granicznych.
- ✓ Dodaj typy danych do atrybutów.



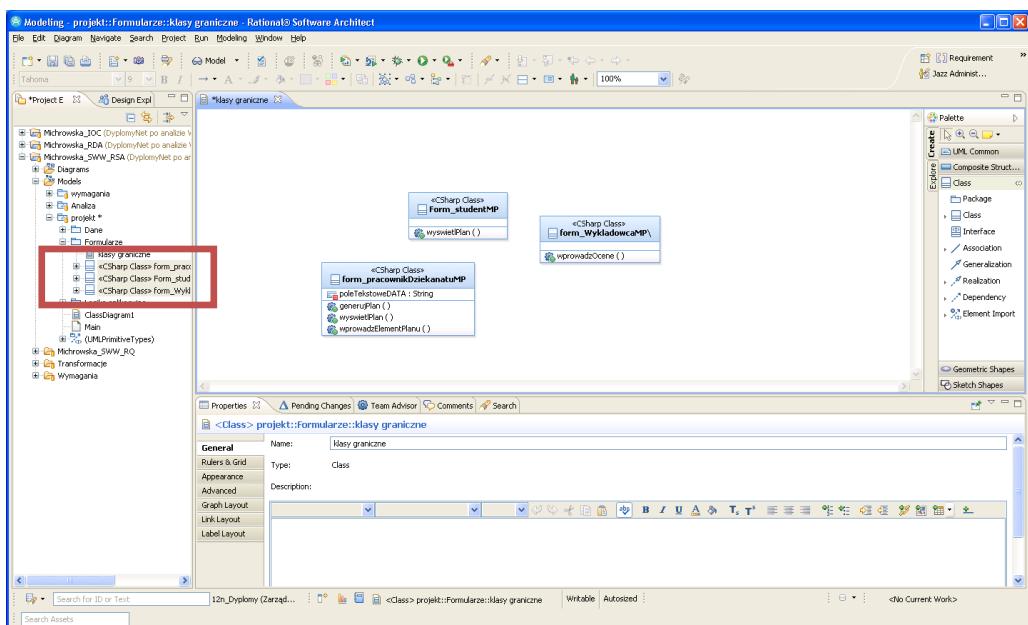
- ✓ Przejdź do Properties pakietu Formularze.



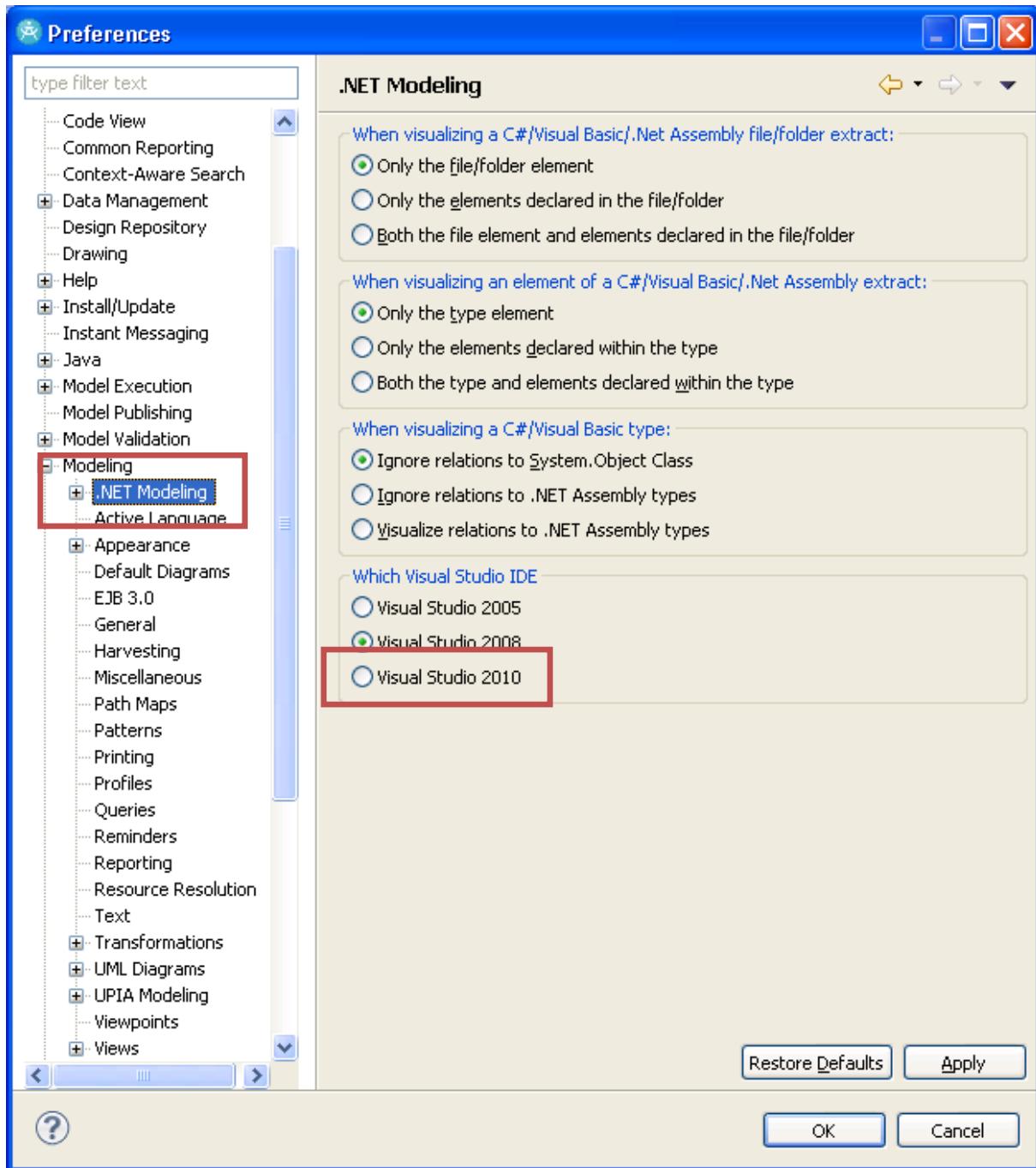
- ✓ Dodaj profil C#



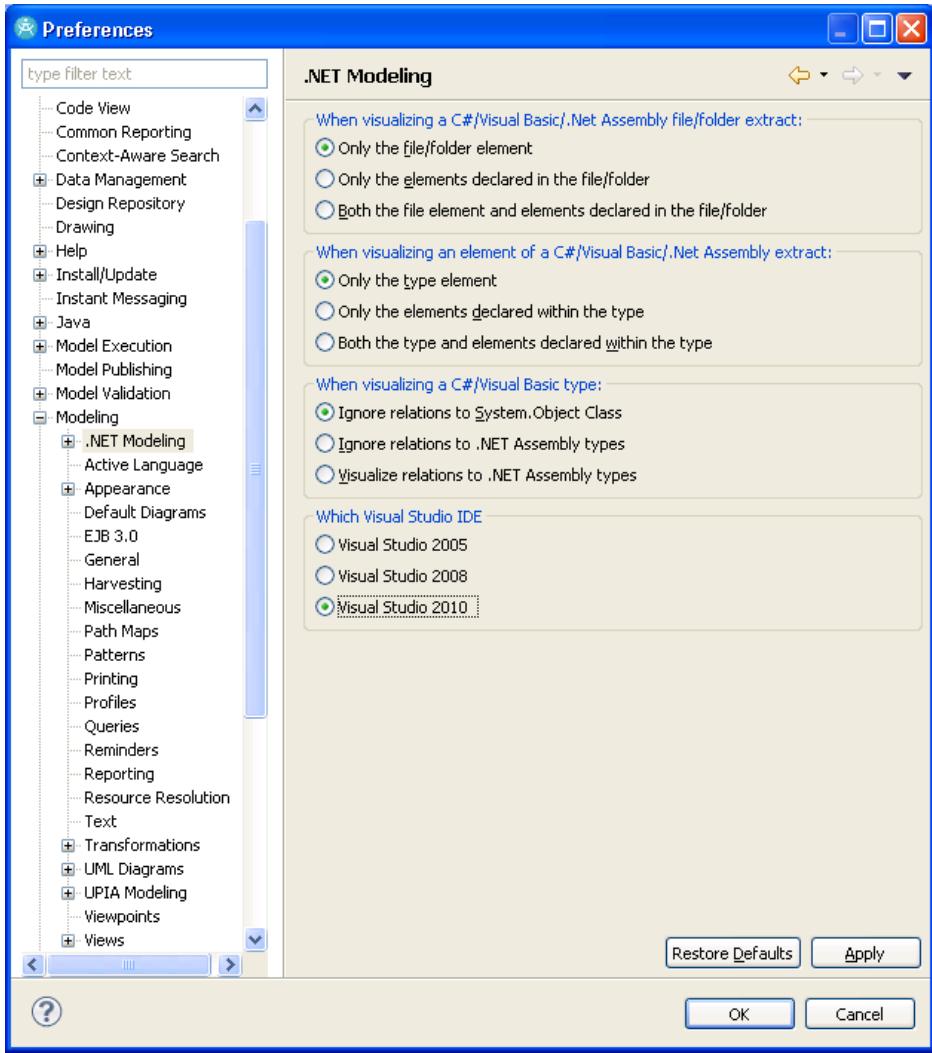
- ✓ Zmień stereotyp klas granicznych na CSharp Class.



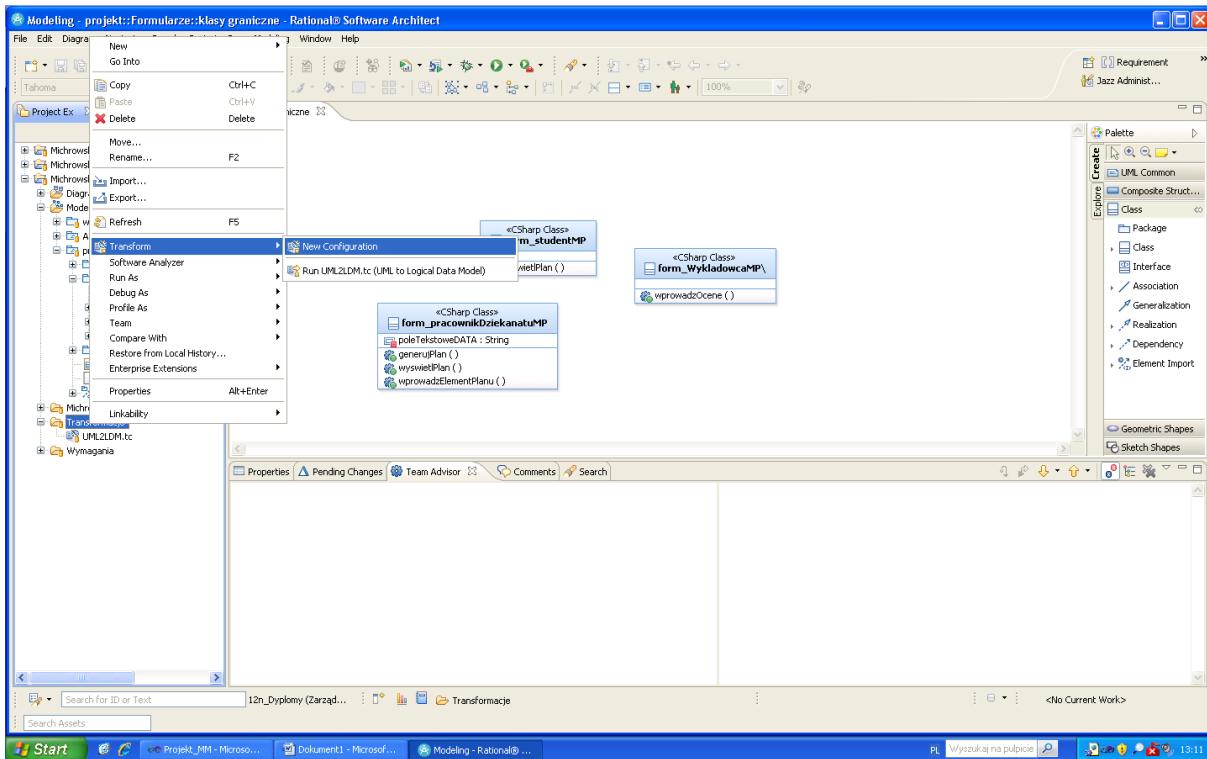
- ✓ Wejdź w window >>> preferences i ustaw Visual Studio jako aktywne.



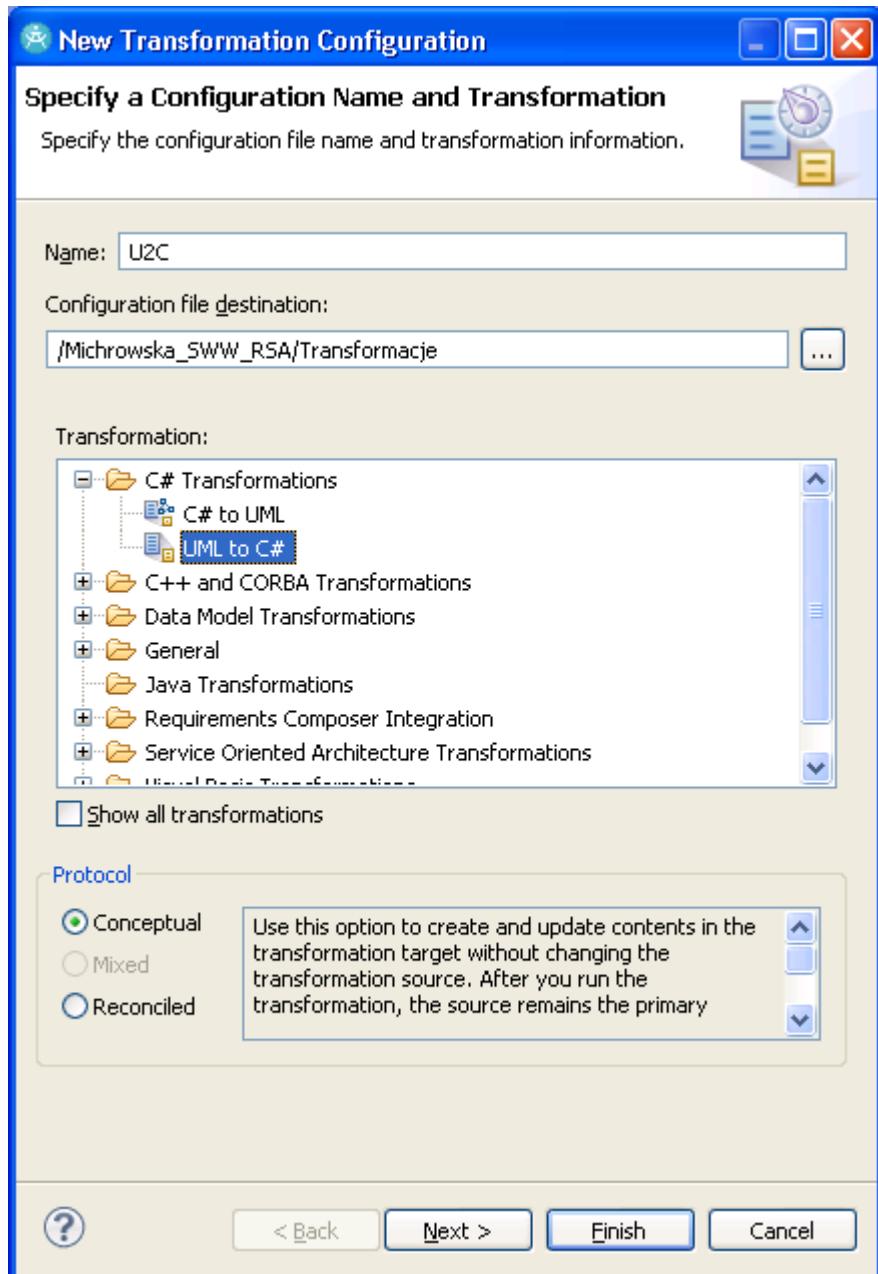
- ✓ Wybierz 2010



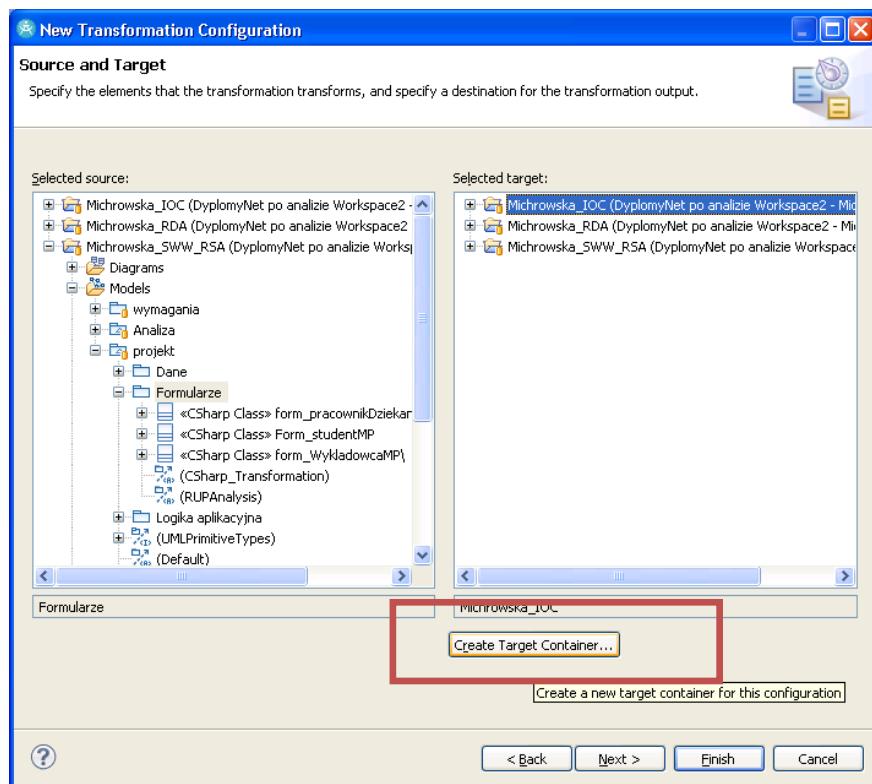
- ✓ OK
- ✓ W folderze transformacje wybierz **Transform >> New Configuration**



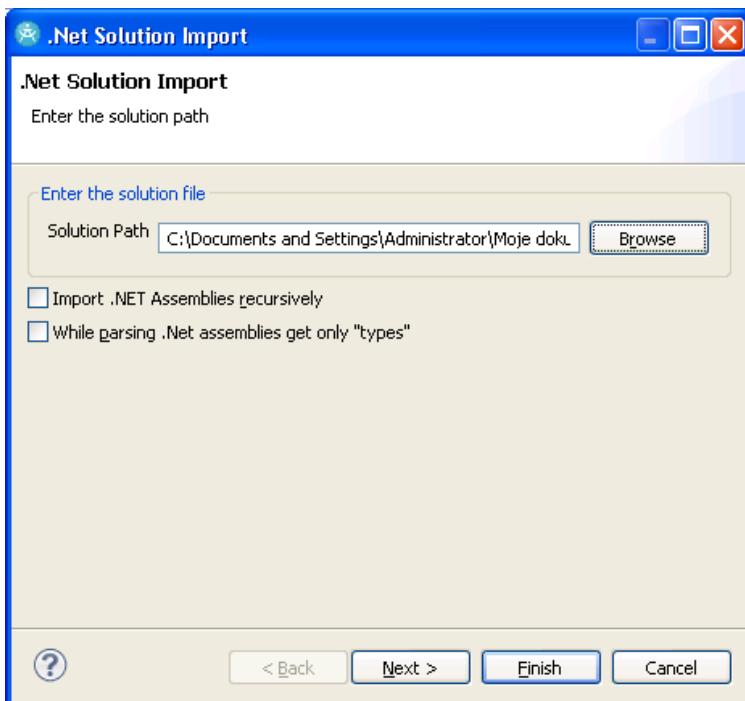
- ✓ Nadaj nazwę
- ✓ Wybierz UML to C#



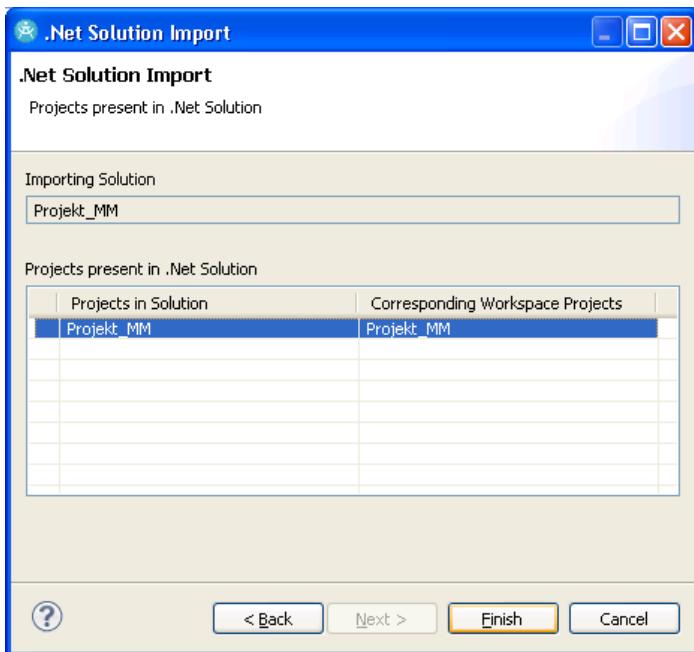
- ✓ Twórz target container, do którego będzie generowany kod.



- ✓ Podaj lokalizację projektu

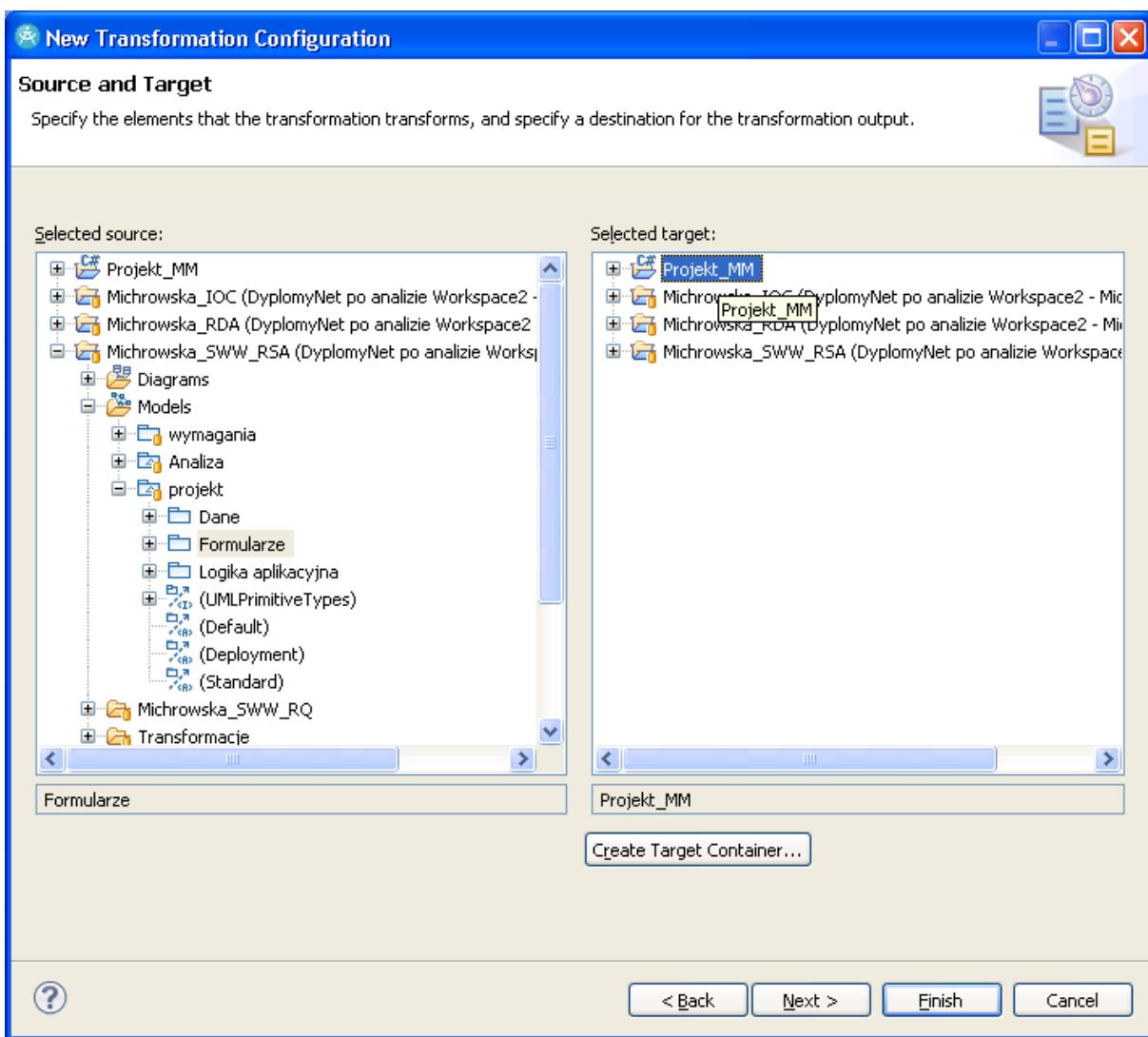


- ✓ Finish

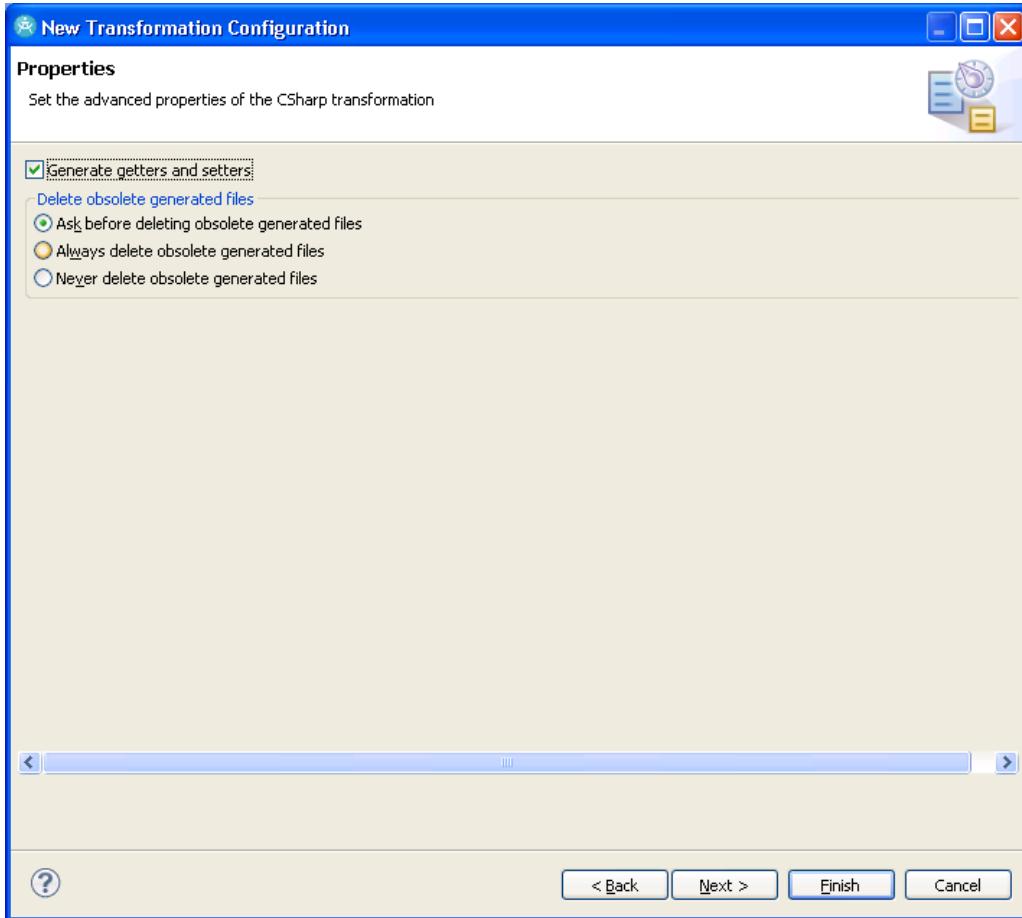


Będzie się robiło jakiś czas.

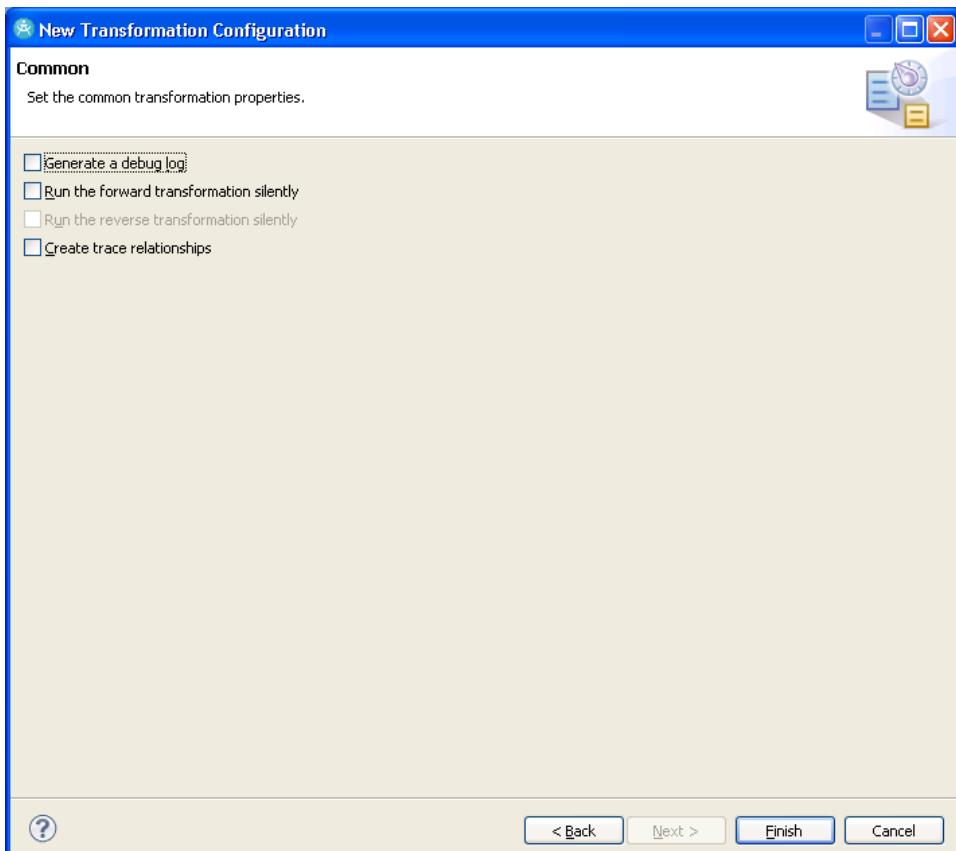
- ✓ Potem wskaż foldery Formularze i Nowopowstały kontener C#



- ✓ Wybierz Next
- ✓ Zaznacz getters i setters

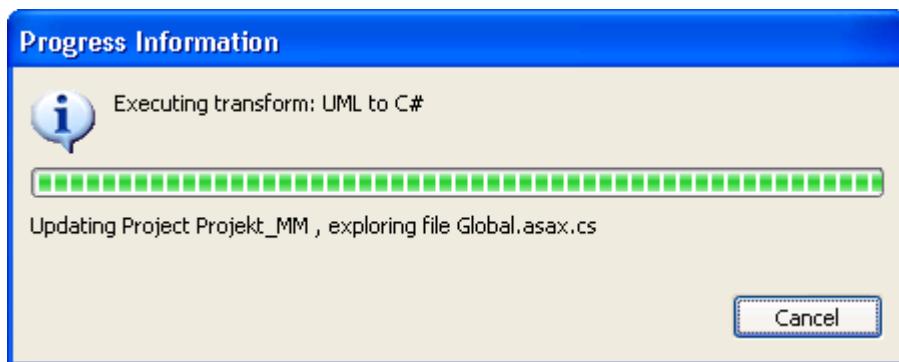


- ✓ Next

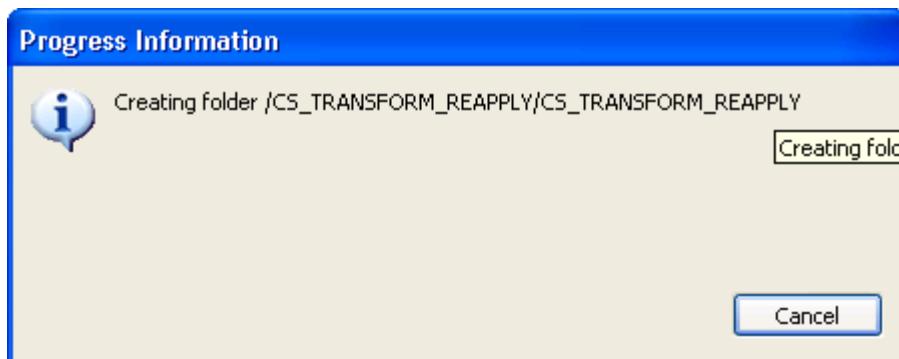


- ✓ Finish

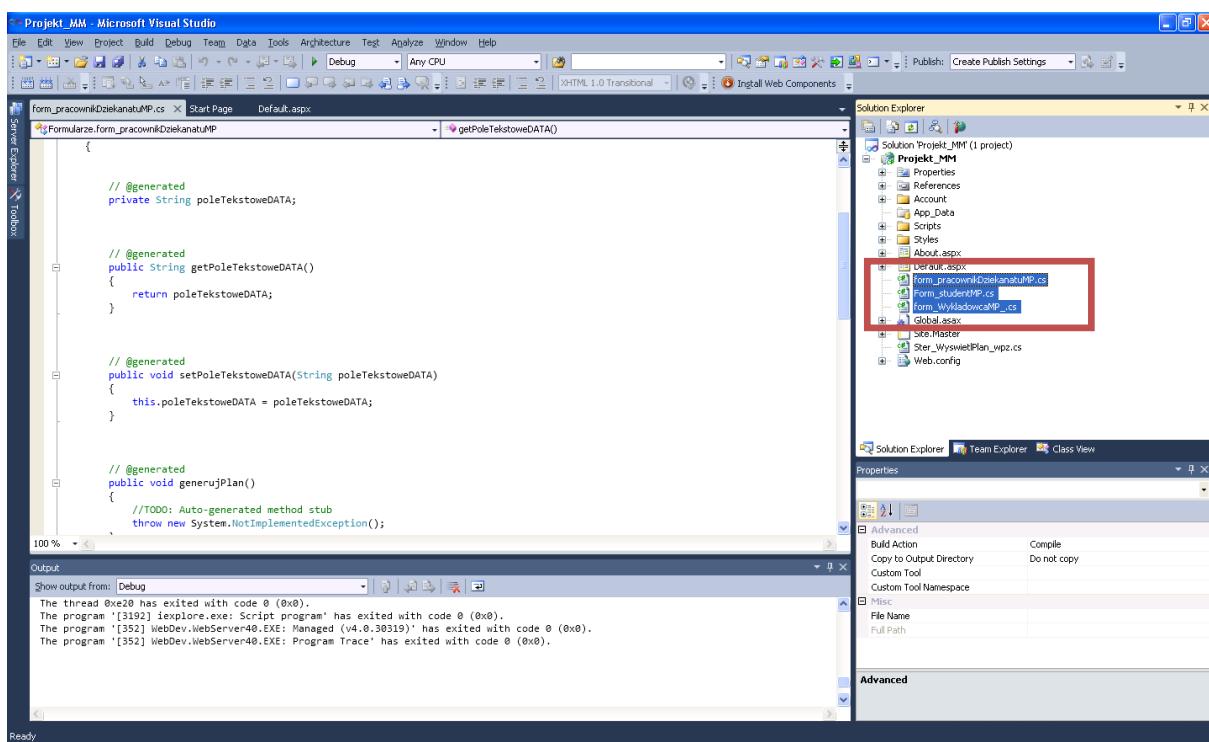
- ✓ Zmień kartę w transformacji na **Source i target** wybierz **validate**, a następnie run



Pojawi się komunikat o taki:

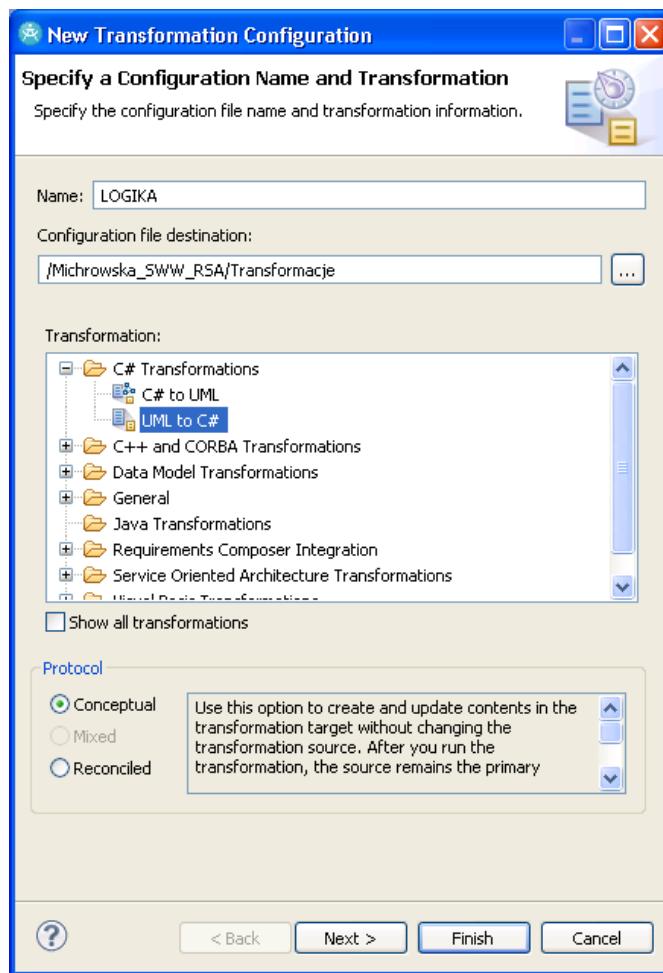


A potem będziesz miał wygenerowany kod w plikach o nazwach klas:



- ✓ Teraz powtarzamy wszystko dla klas z logiką aplikacyjną.

- ✓ Twórz transformację na folderze Transformacje



Wskaż foldery Logika aplikacyjna jako Source i Kontener jako Target- nie trzeba już tworzyć nowego kontenera.

Przy ponownej generacji klas, pojawi się komunikat, czy chcesz usunąć poprzednio generowane klasy odznacz w oknie wszystkie i wybierz No, żeby nie zostały skasowane.

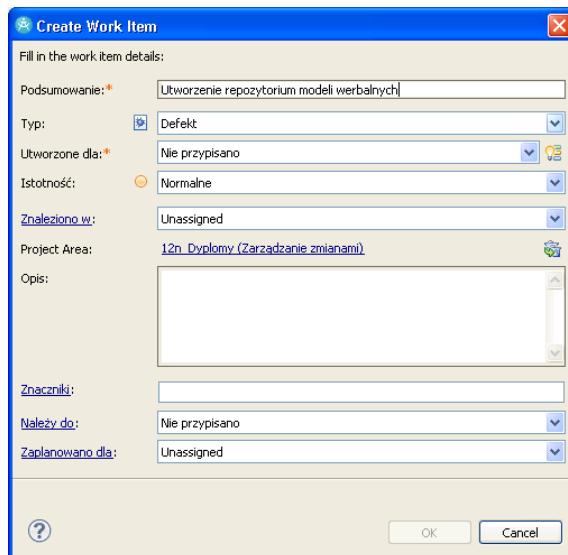
Teraz możesz przesyłać wszystkie zmiany na serwer i cieszyć się zwycięstwem! XD

36. Dodatki

Tworzenie Work Item

Przesyłając zmiany na serwer możesz Tworzyć odpowiednie Work Items, które będą dotyczyły zrealizowanej zmiany.

Zamiast wybierać Assign Comment, wybierz opcje Create Work Item i scustomizuj go, zgodnie z zapotrzebowaniem.



Utwórz model aktywności dla wybranej usługi.

