

Niektóre praktyczne elementy analizy wymagań

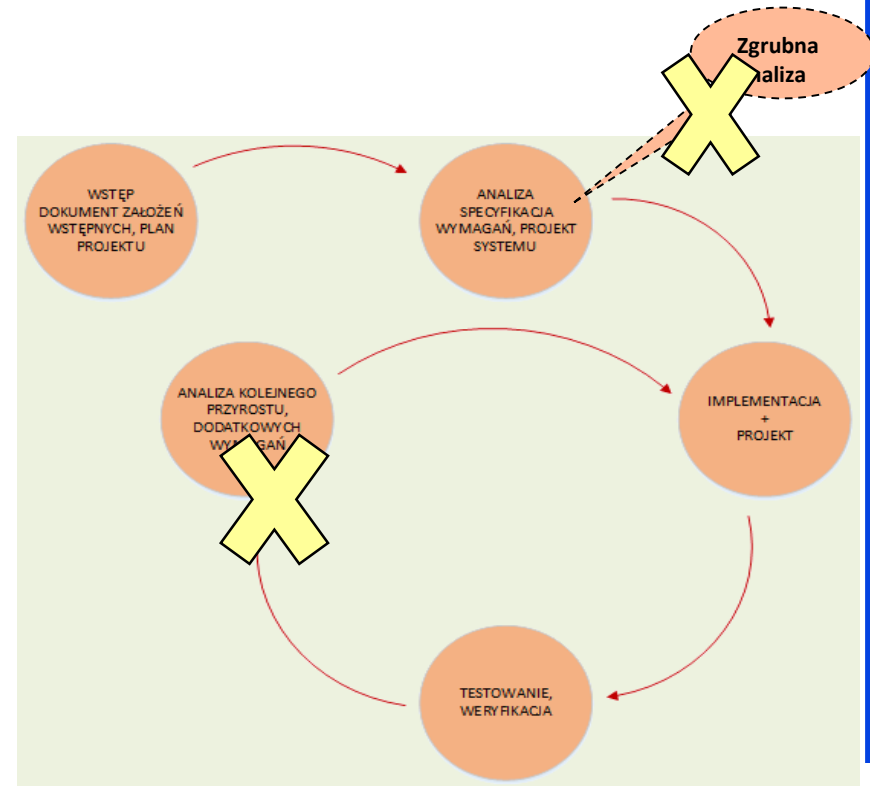


Układ prezentacji

- ☐ Jak zrobić specyfikację wymagań gdy w projekcie planujemy przyrostową realizację.
- ☐ Przypadki użycia
 - Jak dobrze zrobić specyfikację aktorów
 - Jak podejść do problemu granulacji
- ☐ Wzorce analizy - przykłady

❑ Jak zrobić specyfikację wymagań gdy w projekcie planujemy realizację przyrostową

- Zgrubna „standardowa” plus specyfikacja w przyrostach
- Całościowa „standardowa”
 - SWS + Przyp. Użycia + model logiczny
- Specyfikacja agile’owa
 - Wersje jak wyżej
 - Uwaga: w PJATK domagamy się modelowania przypadków użycia



Analiza agilowa (SCRUM) scenariusze (historijki) definiowane w sprintach

➤ Najłatwiej – w powiązaniu z kolejnymi przyrostami odpowiadającymi sprintom realizującym produkty rejestru zaległości (ang. *Product Backlog*)

Profiles

•As a site member, I want to describe myself on my own page in a semi-structured way. That is, I can fill in predefined fields, but also have room for a free-text field or two. (It would be nice to let this free text be HTML or similar.)

•.....

News

•As a site visitor, I can read current news on the home page.

•As a site visitor, I can access old news that is no longer on the home page.

•.....

Courses and Events

•As a site visitor, I can see a list of all upcoming "Certification Courses." I can page through them if there are a lot.

•As a site visitor, I can see a list of all upcoming "Other Courses" (non-certification courses). I can page through them if necessary.

•.....

<http://www.mountaingoatsoftware.com/agile/scrum/product-backlog/example>

□ Praktyka modelowania PU

Przypadki użycia – wsparcie specyfikacji wymagań funkcjonalnych

- ⇓ Specyfikacja aktorów
- ⇓ Diagram (diagramy) PU
 - *biznesowe vs. systemu informatycznego*
- ⇓ Ustrukturalizowany opis (specyfikacja) poszczególnych PU
- ⇓ Zachowanie – przebieg wykonania najważniejszych PU (lub ich sekwencji), udział aktorów i wyróżnionych obiektów systemu
 - diagramy sekwencji, diagramy komunikacji (kolaboracji)
 - diagramy czynności

Przypadki użycia – przykład specyfikacji aktorów, bardzo enigmatyczny(1)

Kierownik projektu	
Opis	Osoba odpowiedzialna za przebieg projektu i identyfikację zagrożeń projektu. Osoba ta przeprowadza analizę, ma szerokie spojrzenie na projekt. Kierownik projektu posiada uprawnienia administratora (zarządzanie pytaniami, udziałowcami, cyklami wytwarzania).
Przypadki użycia	Autoryzacja, Założenie projektu, Przypisanie cyklu wytwarzania do projektu, Zarządzanie cyklami wytwarzania, Zarządzanie iteracjami, Zarządzanie udziałowcami, Zarządzanie pytaniami, Odpowiadanie na pytania, Analiza odpowiedzi, Edycja harmonogramu, Import/export harmonogramu

Odbiorca	
Opis	Jest to kierownik projektu po stronie odbiorcy
Przypadki użycia	Dziedziczy z Kierownik projektu

Dostawca	
Opis	Jest to kierownik projektu po stronie dostawcy
Przypadki użycia	Dziedziczy z Kierownik projektu

*tylko usytuowanie
i przypisanie PU ?*

Przypadki użycia – przykład specyfikacji aktorów (2)

• **Obserwator problemu** – również reprezentowany przez pracowników wydziału oraz studentów. Reprezentowany jest przez osoby posiadające jedną z poniższych ról:

- Osoba zainteresowana problemami z danej kategorii
- Osoba zainteresowana problemem

Obserwator problemu posiada uprawnienia zwykłego użytkownika poszerzone o możliwość wglądu w cykl życia problemów poufnych, których jest obserwatorem. Nie bierze jednak bezpośredniego udziału w cyklu życia problemu.

• **Udziałowiec problemu** – użytkownik będący bezpośrednim udziałowcem problemu (patrz kryterium II w punkcie 1.4.3.5). Reprezentowany jest przez studentów i pracowników wydziału posiadających jedną z poniższych ról:

- Właściciel problemu
- Zgłaszający problem
- Adresat problemu
- Wykonawca problemu

*opis roli i kogo reprezentuje
(odzwierciedla)*

Udziałowiec problemu posiada uprawnienia obserwatora poszerzone o możliwość edycji problemów, których jest udziałowcem. Bierze więc bezpośredni udział w cyklu życia problemu.

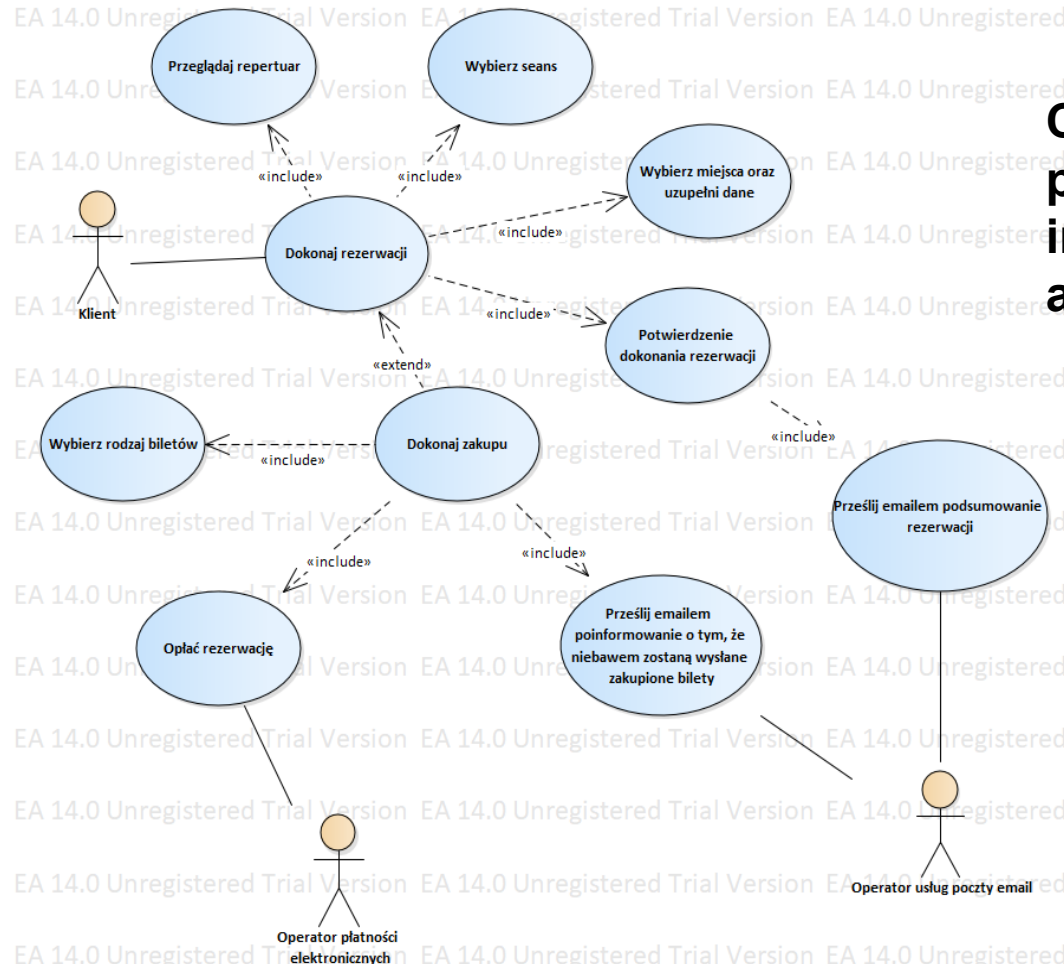
Właściciel problemu dodatkowo ma możliwość zmiany kategorii problemu, jego poufności oraz dodania i usuwania ewentualnych obserwatorów problemu.

• **Administrator** – osoba posiadająca powyższe role rozszerzone o uprawnienia administratora. Ma również możliwość wykonywania wszelkich operacji na problemach, stąd też można go jednocześnie traktować jako ukrytego udziałowca wszystkich problemów. Dodatkowo posiada również możliwość zakładania kont użytkownikom, uniewidaczniania problemów w systemie, wykonywania operacji na grupach użytkowników i kategoriach problemów oraz przydzielania użytkownikom ról w ramach poszczególnych kategorii problemów.

Każdy z powyższych aktorów posiada jednocześnie uprawnienia wynikające ze stanowisk, jakie piastują w ramach wydziału. Dotyczą one widoczności problemów w systemie i są szerzej omówione w punkcie, gdzie zostały również dokładnie określone zasady dotyczące widoczności problemów w systemie w zależności od ról, jakie użytkownik pełni w ramach różnych kryteriów.

Problem granulacji PU

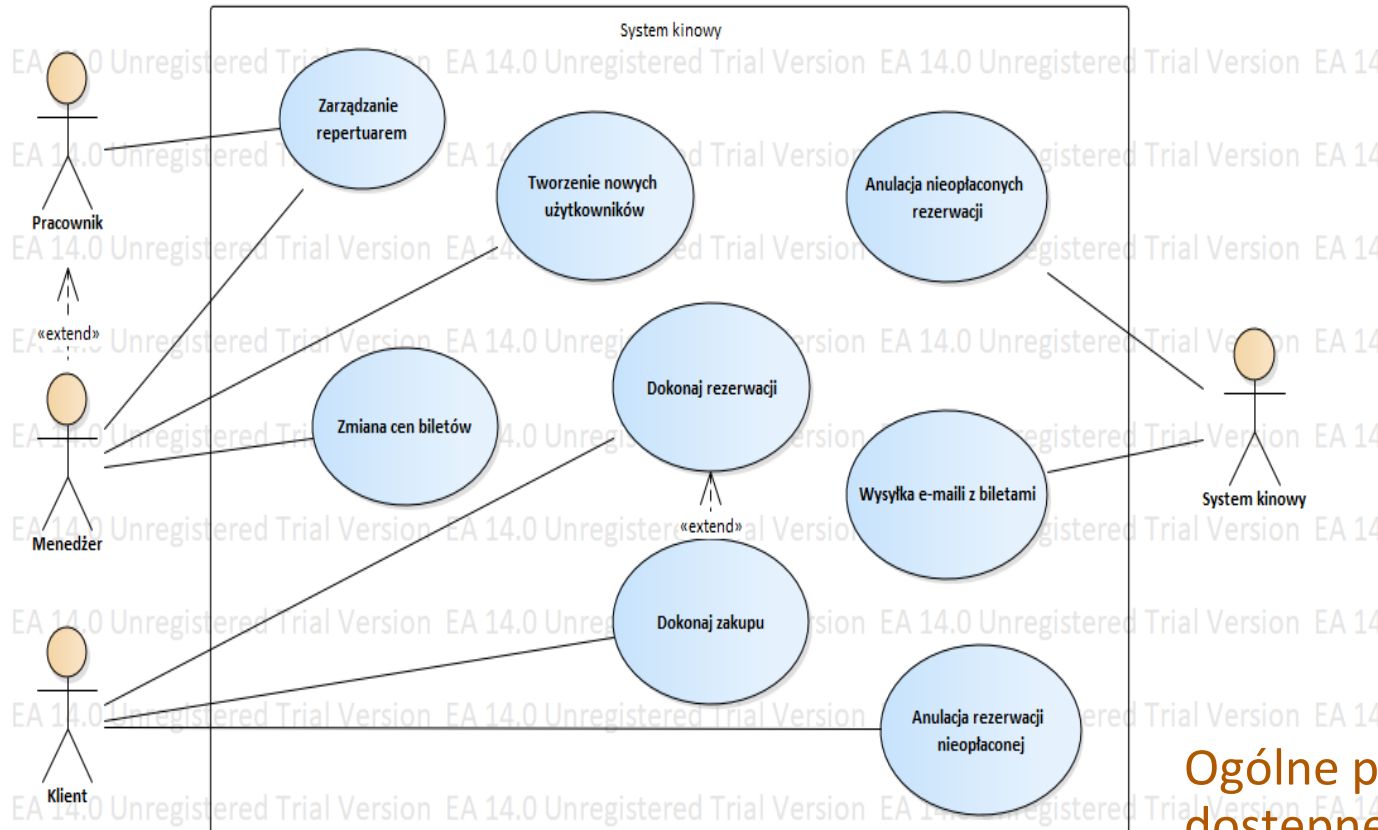
Czy (kiedy) przypadek użycia powinien reprezentować każdą interakcję użytkownika z systemem, a kiedy spełnienie celu aktora?



Całościowy przypadek użycia związany z rezerwacją biletów

M. Sielawa , M. Niewiarowski, M. Groth, S. Zblewski: SUPREME - System wspomagający pracę kina. Praca dypl. PJATK 2018

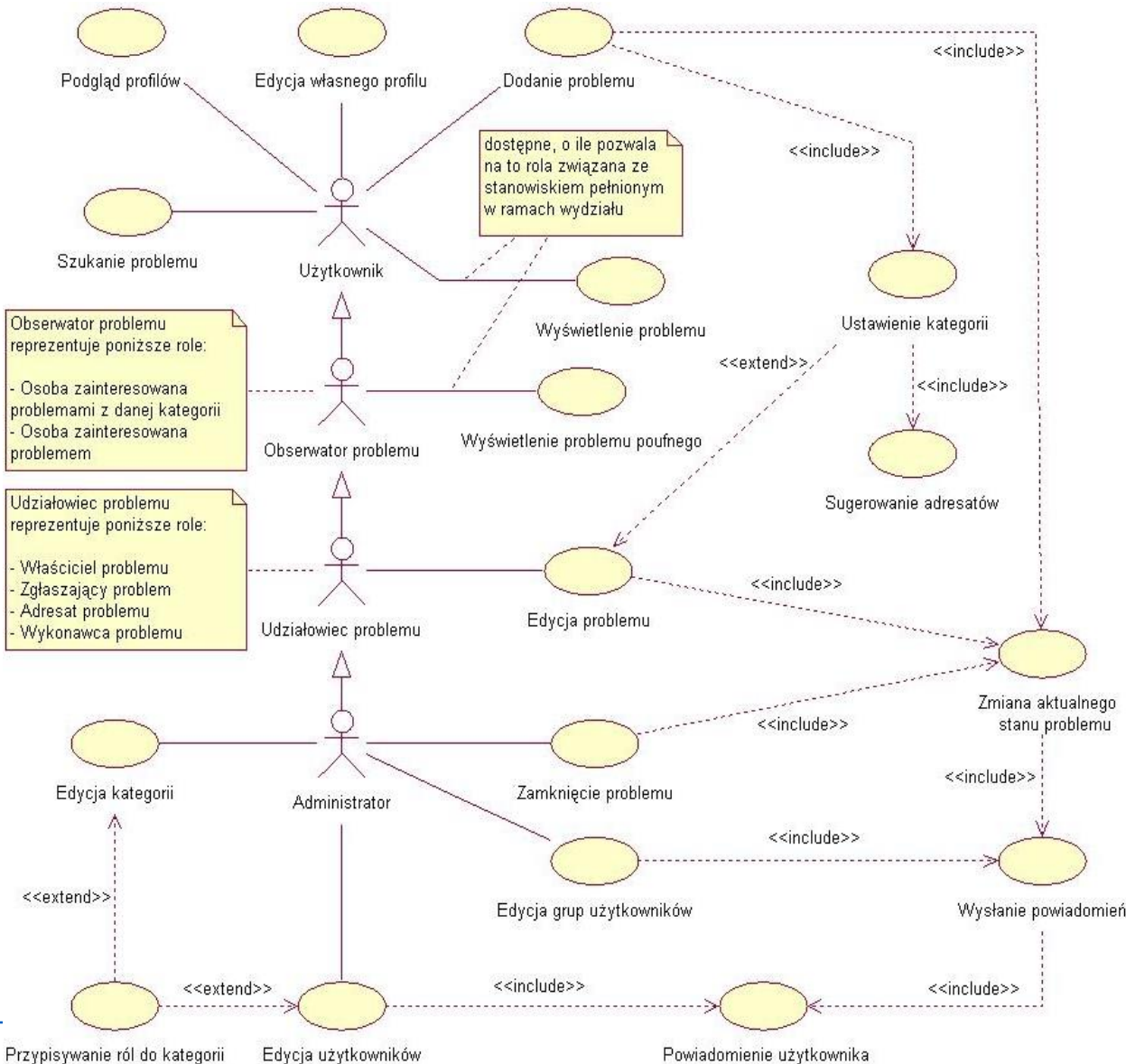
Problem granulacji PU



Ogólne przypadki użycia dostępne w systemie

M. Sielawa , M. Niewiarowski, M. Groth,
S. Zblewski: SUPREME - System wspomagający
pracę kina. Praca dypl. PJATK 2018

Przypadki użycia – diagram PU

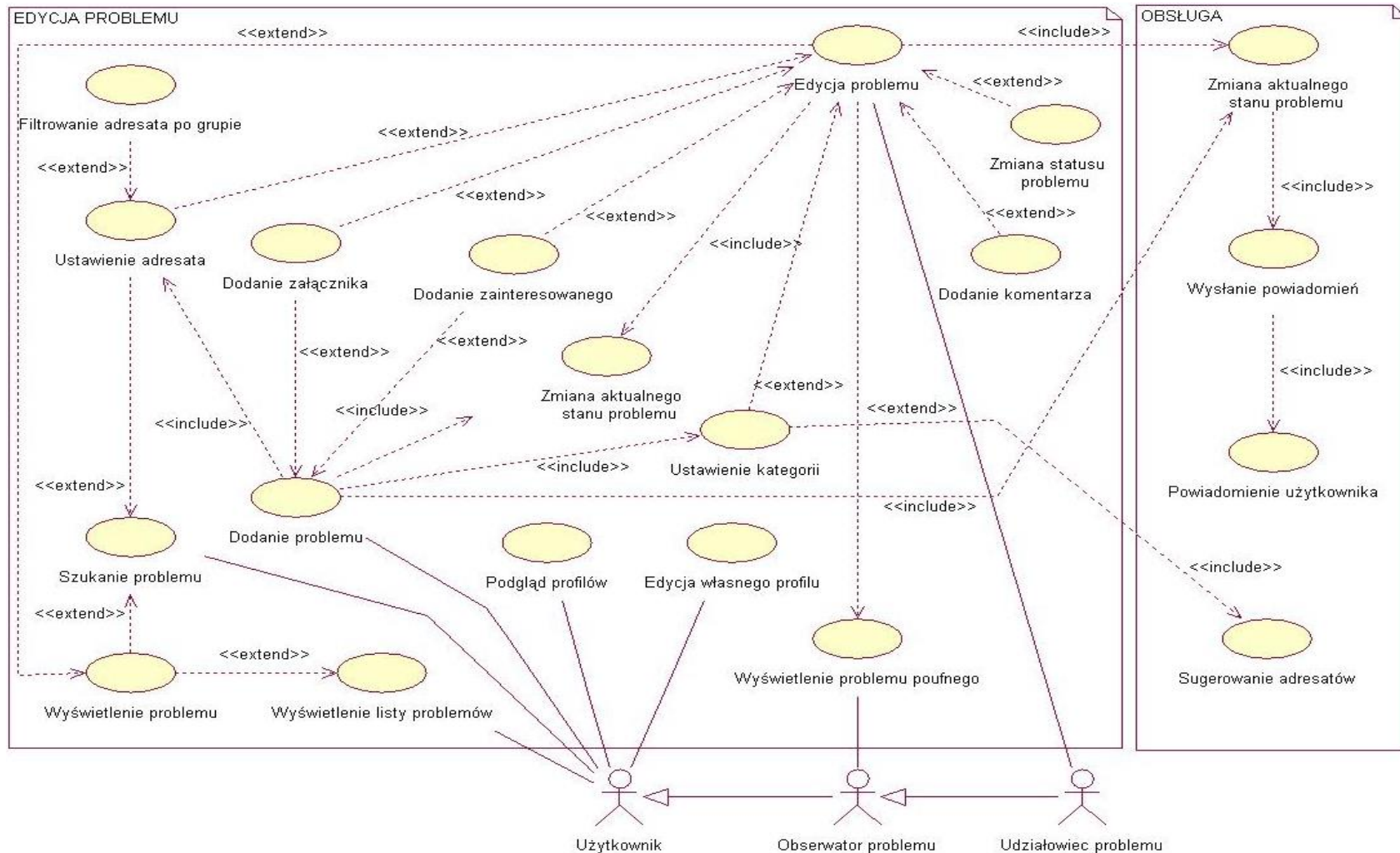


*T. Łutowicz: System
monitorowania problemów
procesu dydaktycznego*

diagram zbiorczy

Przypadki użycia – *diagramy szczegółowe*

T. Łutowicz: System monitorowania problemów procesu dydaktycznego



Co zrobić gdy diagram przypadków użycia „wychodzi” mocno rozbudowany i mało czytelny?

- **Tworzyć diagramy hierarchiczne**

np. na diagramie ogólnym umieścić PU *Administrowanie systemem*, a na kolejnych rozwinąć ten PU o przypisane mu (include, extends) PU szczegółowe, jak *Administrowanie użytkownikami*, *Zarządzanie uprawnieniami*, *Edycja itp.*

← przykłady

- **Zweryfikować możliwość uogólnienia, związki, np. pod kątem dziedziczenia funkcji przez aktorów**

- **Tworzyć diagramy dla poszczególnych podsystemów logicznych**
Wówczas nieujęte na diagramie podsystemy modelujemy jako aktorów!

- **Tworzyć diagramy dla poszczególnych aktorów**

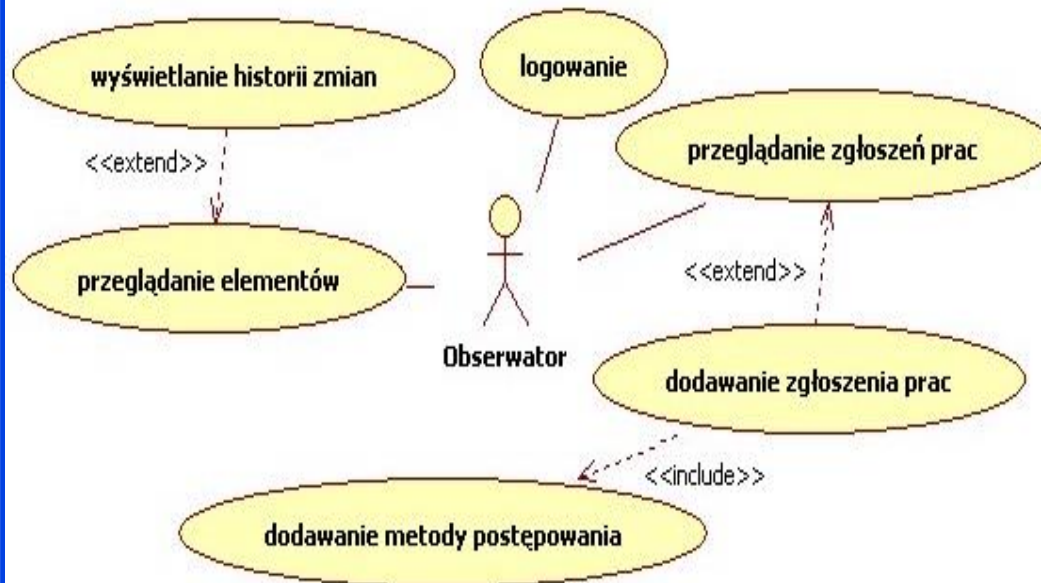
np. odrębnie dla zwykłego użytkownika, odrębnie dla administratora,...

przykład →

Przypadki użycia – *uszczegóławianie według zadań poszczególnych aktorów*

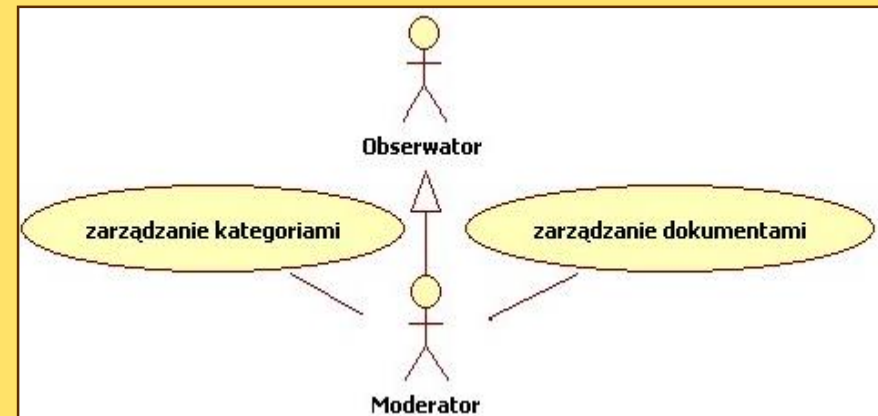
M. Janicki *System wspomagający projektowanie wag elektronicznych*

Diagram przypadków użycia Obserwatora



Głównym zainteresowaniem Obserwatora jest przeglądanie elementów wprowadzonych do systemu, ich powiązań i historii zmian. ... Aktor o najmniejszym zakresie uprawnień, którego możliwości posiadają wszyscy inni aktorzy (dziedziczą jego PU).

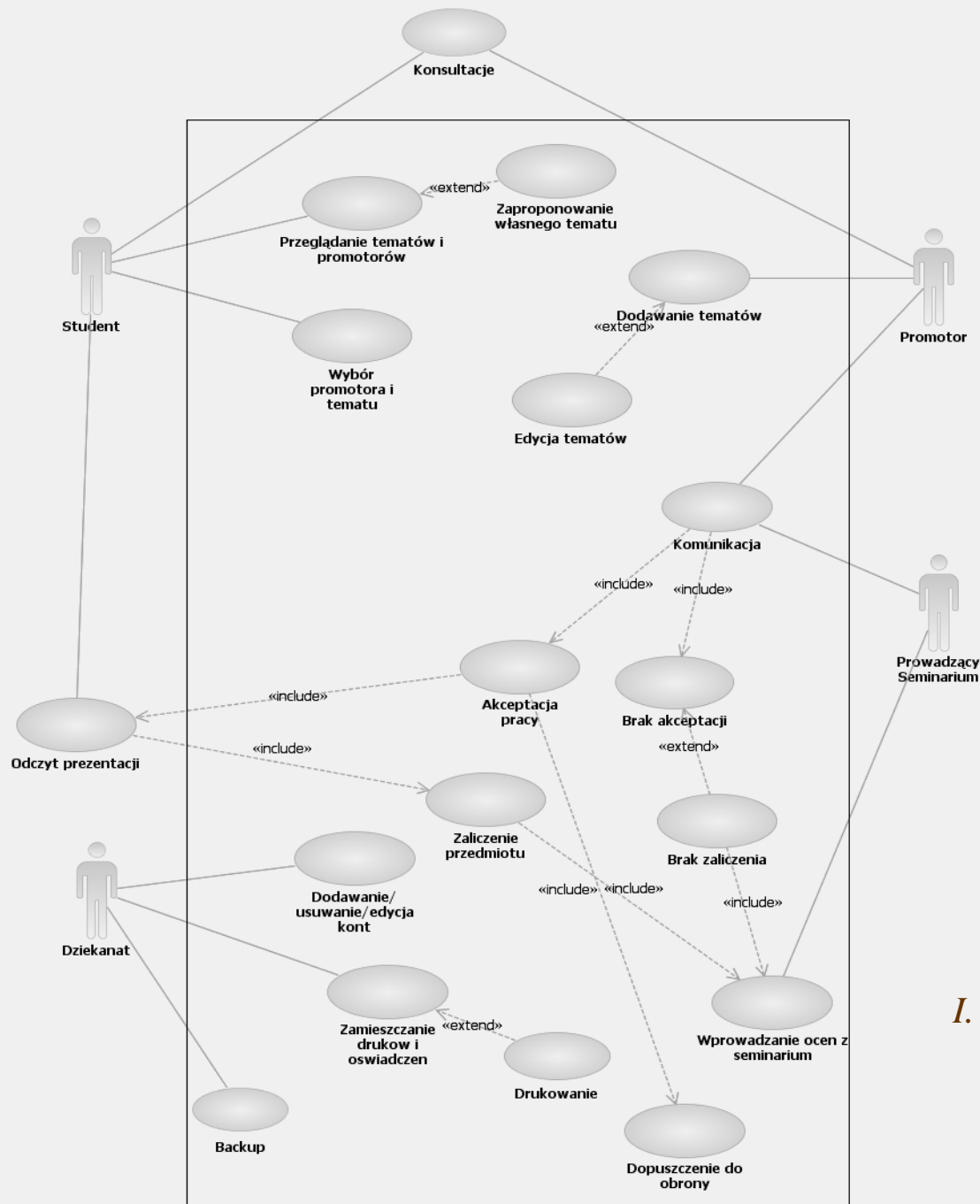
Diagram przypadków użycia Moderadora



Moderator to abstrakcyjny aktor, który zarządza dokumentami zgodnymi ze swoim profilem. ...Ma dostęp do wybranych elementów i dokumentów, jak Redaktor do instrukcji, Programista - do programów, czy Konstruktor - do części mech. i elektroniczn.

Co zrobić gdy w analizie pojawiają się zadania spoza zakresu? albo nienależące do aktualnie rozpatrywanego wydania?

- Można przedstawić je jako podsystemy (logiczne) reprezentowane przez aktorów
- Rysując pewne PU poza ramką określającą zakres systemu *przykład* →
- Wyodrębniamy – np. cieniujemy (kolorujemy) na diagramie przypadki użycia nienależące (albo należące) do zakresu *przykład* →



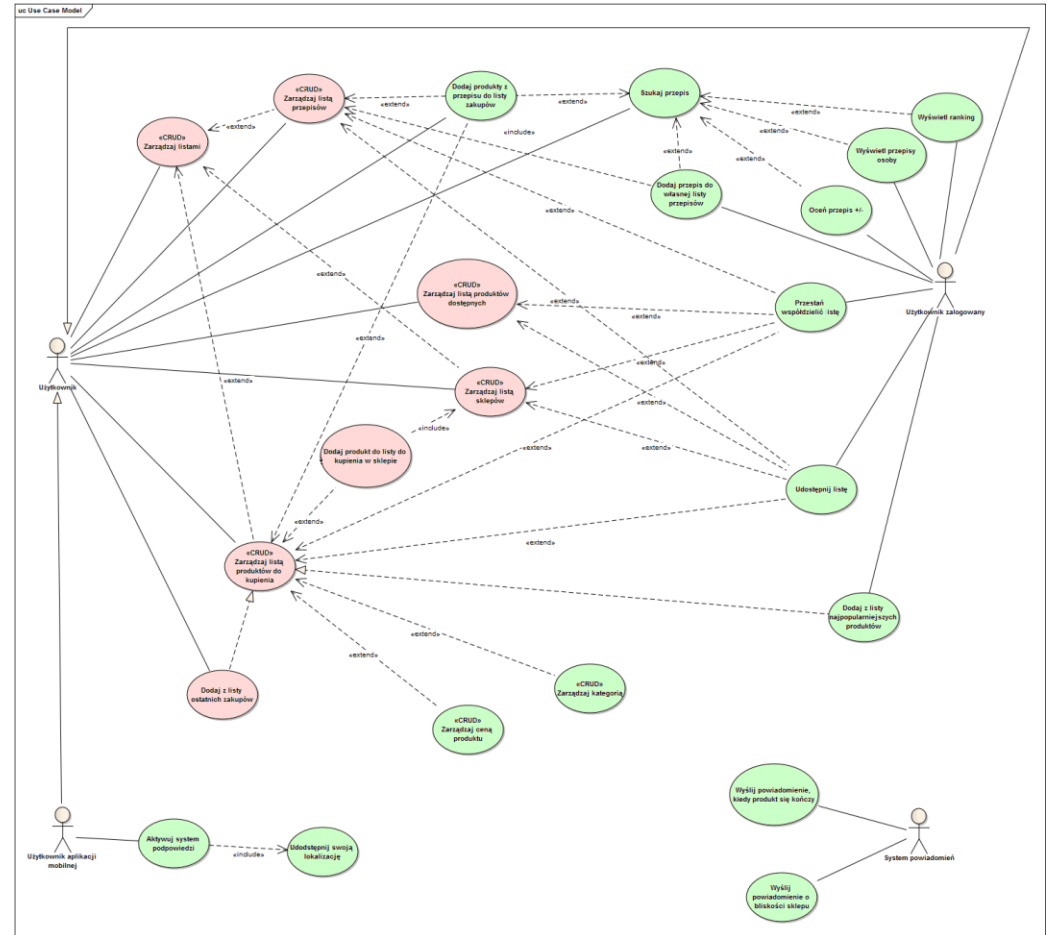
Przypadki użycia – modelowanie przypadków pozostających poza systemem

I. Derengowska: System monitorowania przebiegu realizacji prac dyplomowych

Przypadki użycia – wyodrębnianie

Nadawanie wag lub priorytetów

- w celu ujęcia całościowego kontekstu
- dla wyróżnienia wycinka pracy lub podsystemu
- dla potrzeb realizacyjnych
 - wskazanie kolejności
 - wydaniem



RYSUNEK 3 - DIAGRAM PRZYPADKÓW UŻYCIA (KOLOR CZERWONY – I WYDANIE, KOLOR ZIELONY – II WYDANIE)

A. Jackiewicz, A.Twórz, B. Nowaczyk, P. Jabłoński: Projekt i realizacja interaktywnego asystenta zakupów – Concierge. Praca dypl. ZWI PJATK, 2015

□ Dwa przykłady wzorców analizy

1. Wzorzec Rezerwacja – Użycie
2. Wzorzec *Repair of an Entity*



Przykład wzorca analizy

- W wielu systemach występuje problem rezerwacji i użycia jednostek, np. pokoi w hotelu, książek w bibliotece, miejsc na kursach na uczelni lub w teatrze, samochodów w wypożyczalni, samolocie itp. Klient żąda rezerwacji jednostki określonego typu, aby ją potem użyć w określonym przedziale czasu. Następnie jednostka jest zwracana i może być ponownie użyta. Liczba jednostek jest ograniczona i grupowana w kategorie.
- Model analizy musi reprezentować wymagania i nie może zawierać szczegółów implementacyjnych. Powinien opisywać większość sytuacji związanych z rezerwacją i użyciem jednostki.

różnorodność dziedzin
zastosowania !

Wzorzec: Rezerwacja-Użycie

- Rozwiązanie oparte jest na realizacji powtarzających się przypadków użycia: *Rezerwacja*, *Odwołanie rezerwacji*, *Użycie zarezerwowanej jednostki*, *Zwrot jednostki*. Przyjęte są model klas i diagramy sekwencji, odpowiadające realizacji wymienionych przypadków użycia. Model klas rozróżnia typ (kategorię) jednostki od konkretnej jednostki przydzielanej w momencie jej użycia.

Przykład wzorca analizy *Rezerwacja - użycie zasobów*

(za E. B. Fernandez and X. Yuan: „An analysis pattern for reservation and use of entities”)

Nazwa: Rezerwacja-Użycie zasobów

Problem. W wielu systemach występuje problem rezerwacji i użycia jednostek, na przykład pokoi w hotelu, książek w bibliotece, kursów na uczelni, samochodów w wypożyczalni lub miejsc w teatrze, samolocie itp. Klient żąda rezerwacji jednostki określonego typu, aby ją potem użyć w określonym przedziale czasu. Następnie jednostka jest zwracana i może być ponownie użyta. Liczba jednostek jest ograniczona i grupowana w kategorie.

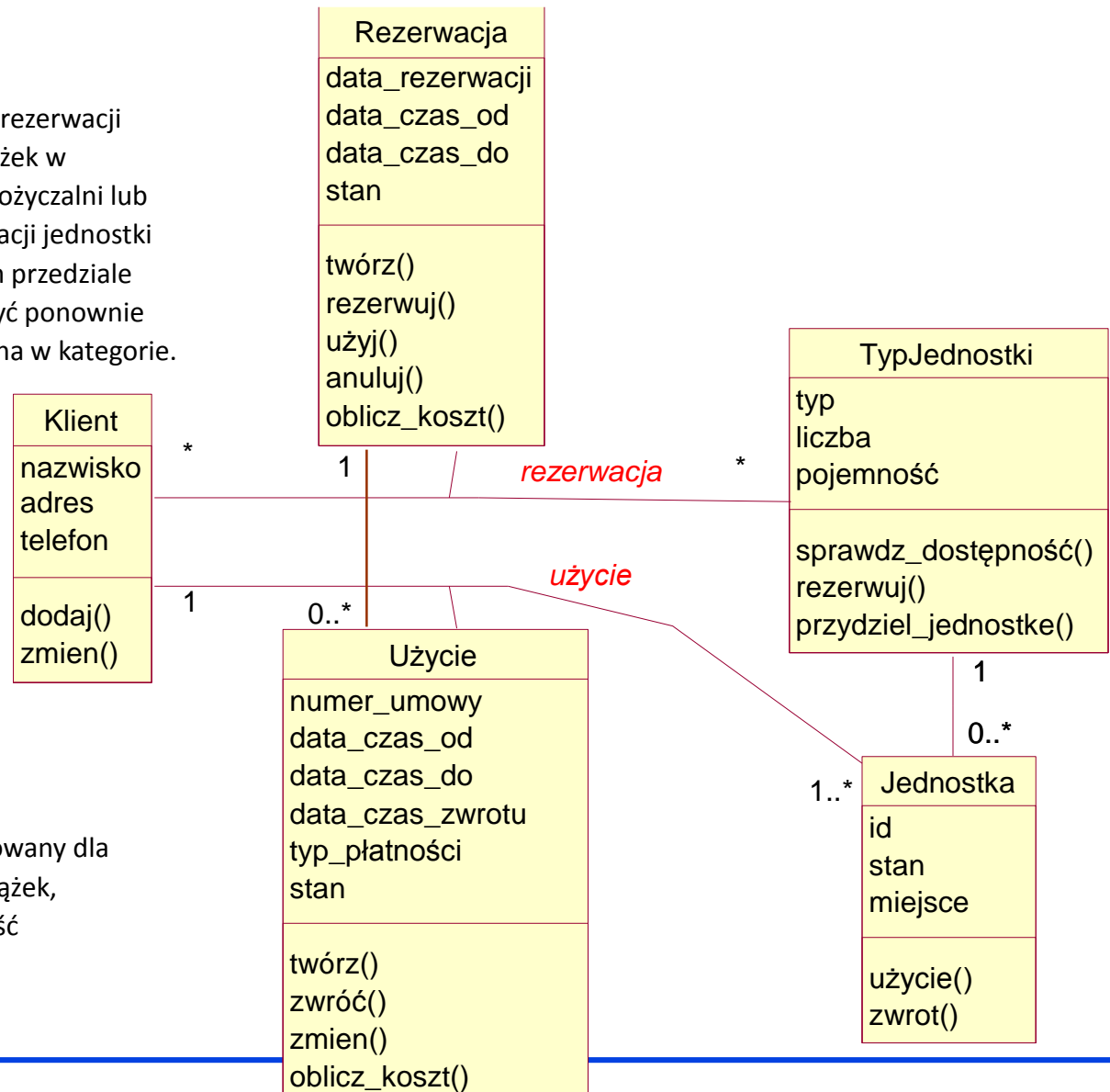
Rozwiązanie:



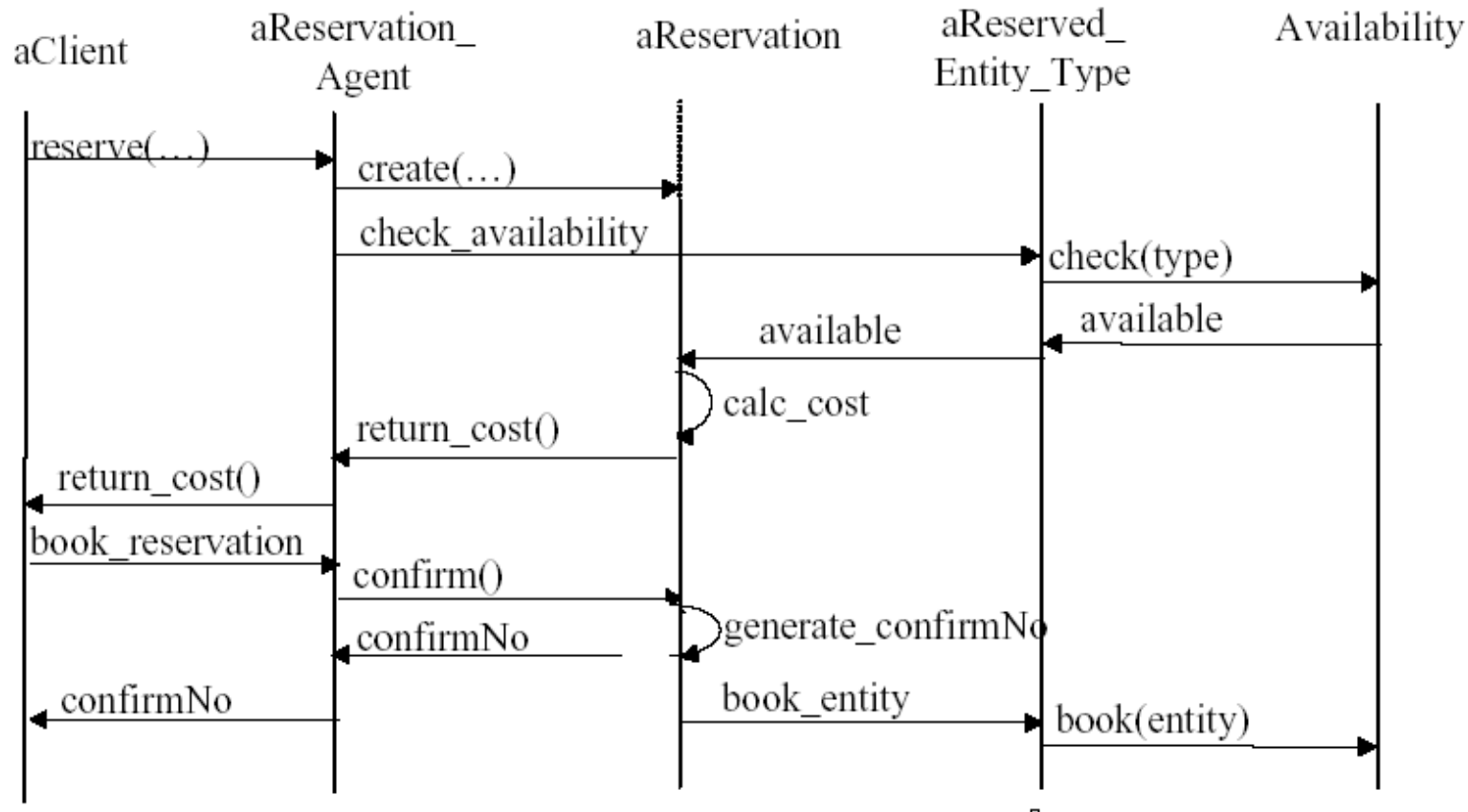
Oparte jest na realizacji powtarzających się przypadków użycia: *Rezerwacja*, *Odwołanie rezerwacji*, *Użycie zarezerwowanej jednostki*, *Zwrot jednostki*. Model klas rozróżnia typ (kategorię) jednostki od konkretnej jednostki przydzielanej w momencie jej użycia.

Prezentacja na wykładzie pomija diagramy i specyfikację PU

Wyniki i konsekwencje: Wzorec może być zastosowany dla modelowania aplikacji rezerwacji i wypożyczeń książek, wideo, łódek, pokoi, biletów itp. Ta wariantowość wskazuje, że wzorec ujmuje podstawowe aspekty problemu. Jest ogólny, prosty i zrozumiały.

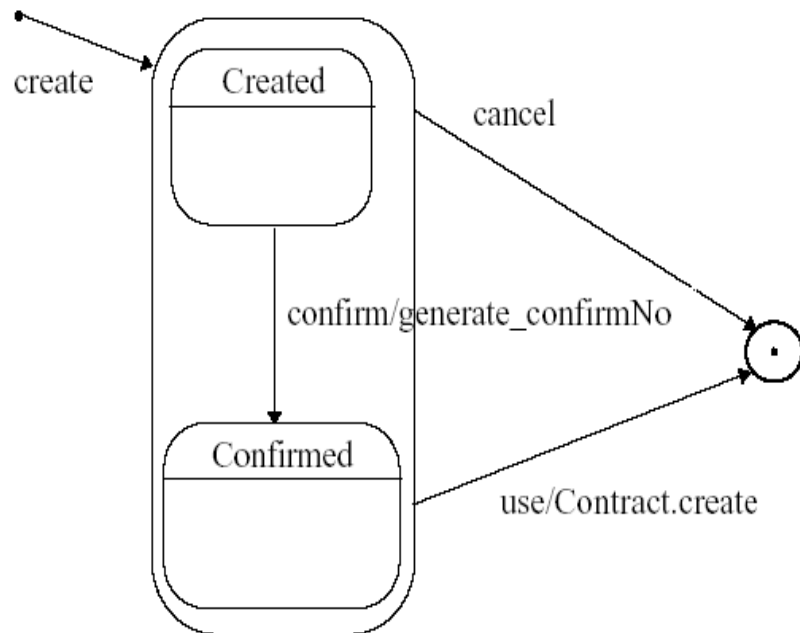


Rezerwacja - użycie zasobów. Diagram sekwencji Rezerwacji

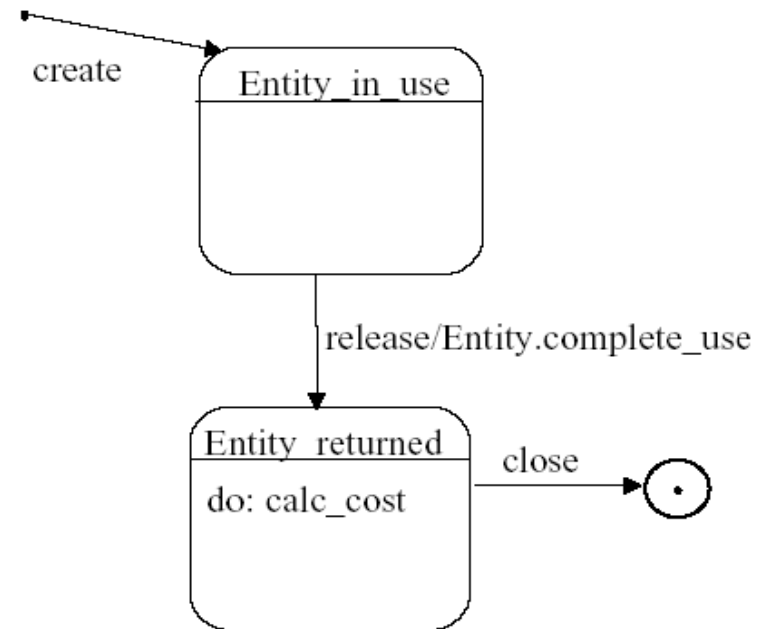


Przebieg rezerwacji danego zasobu: Klient zgłasza, że chce zarezerwować zasób na dany okres czasu. Pracownik sprawdza więc czy w tym czasie będzie wolny zasób określony przez klienta. Jeśli tak – to jest obliczany jest koszt rezerwacji, o którym informuje się klienta. Następnie jest tworzone potwierdzenie rezerwacji i dokonywana jest już rezerwacja konkretnego egzemplarza.

Diagramy stanów

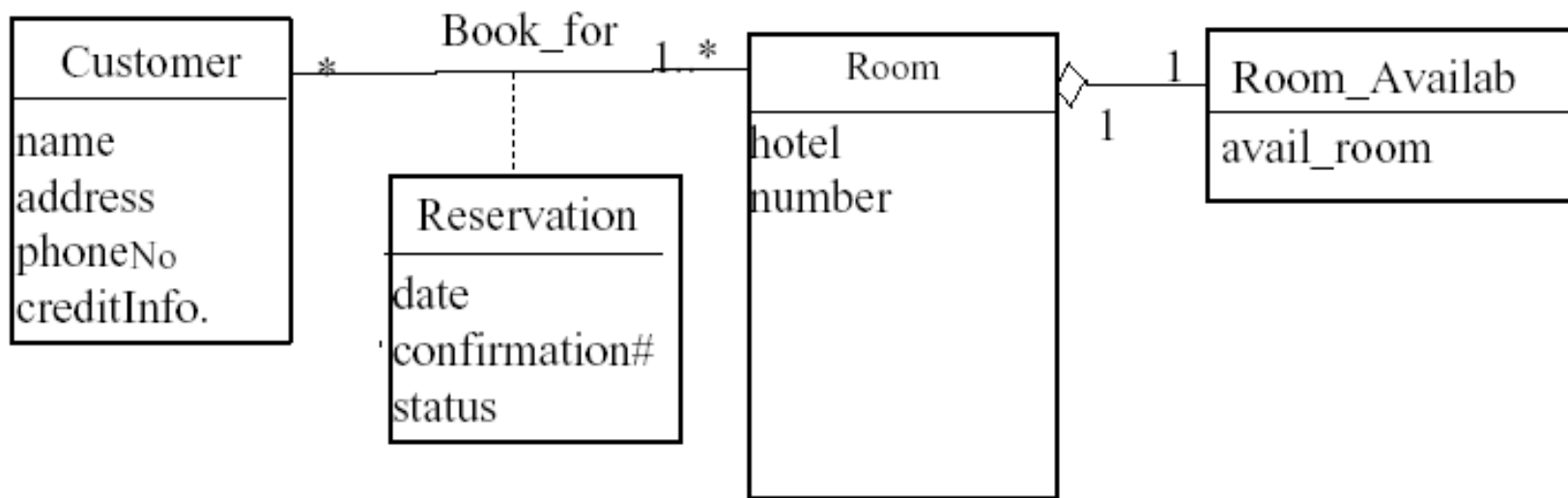


Dla klasy *Rezerwacja*



Dla klasy *Użycie*

Użycie wzorca *Rezerwacja - użycie* *np. rezerwacja pokoju hotelowego*



Ukazany wzorzec można odnieść do rezerwacji pokoju w hotelu, ale także miejsca w kinie czy w samolocie, samochodu w wypożyczalni itp.

Użycie wzorca *kontekst, ograniczenia*

Nie wszystkie sytuacje opisane przez ten wzorzec są takie same:

- otrzymuję bilet, kiedy rezerwuję miejsce w samolocie czy w kinie, ale nie kiedy rezerwuję np. pokój w hotelu
- rezerwacja samochodu jest finalizowana, kiedy klient odbiera samochód, ale rezerwacja w hotelu na przykład, kiedy gość się wprowadza do pokoju
- rezerwacja pokoju w hotelu jest tworzona na określony czas, natomiast rezerwacja miejsca w kinie jest zawierana na dane wydarzenie (nie ma rezerwacji od-do)
- kontrakt może też zostać utworzony bez zawierania rezerwacji

Wzorzec *Repair of an Entity*

E. B. Fernandez and X. Yuan: „*Semantic Analysis Patterns*”

Problem:

Jak możemy zamodelować proces naprawy jakiegoś przedmiotu?

Kto odpowiada za identyfikację uszkodzenia i naprawę?

Może jakaś dokumentacja (rejestr) napraw danego urządzenia...?

Zakres *Repair of an Entity*:

- Zamodelowanie podstawowych aspektów naprawy sprzętu, począwszy od początkowych ocen uszkodzonego sprzętu aż po finalizację reparacji.
- Przechowywanie informacji na temat naprawianego sprzętu
- Umożliwienie klientowi wyboru serwisu na podstawie wstępnych wycen.

Rozwiązanie proponowane przez *Repair of an Entity* :

- Nabywcy przynoszą sprzęt do serwisu lub zgłaszają uszkodzenie sprzętu.
- Nabywca może wstępnie porównać warunki naprawy kilku serwisów i wybrać jeden z nich
- Pracownik recepcji robi wycenę uszkodzenia (cena naprawy), a także odpowiada za prowadzenie dziennika napraw.
- Jeśli użytkownik zdecydował się na nasz serwis, zostaje wyznaczony technik naprawiający
- Każdy sprzęt (obiekt) ma kartę, w której są rejestrowane wszystkie naprawy

Repair of an Entity

Diagram klas

Opis podstawowych klas:

- BrokenEntity – uszkodzony sprzęt
- RepairShop – punkt naprawy
- Customer – klient
- RepairEvent – dziennik napraw
- RepairShopChain – zbiór wszystkich punktów naprawy urządzenia
- ReceptionTechnician – pracownik wykonujący wycenę
- RepairTechnician – pracownik naprawiający sprzęt
- Estimate – wycena naprawy

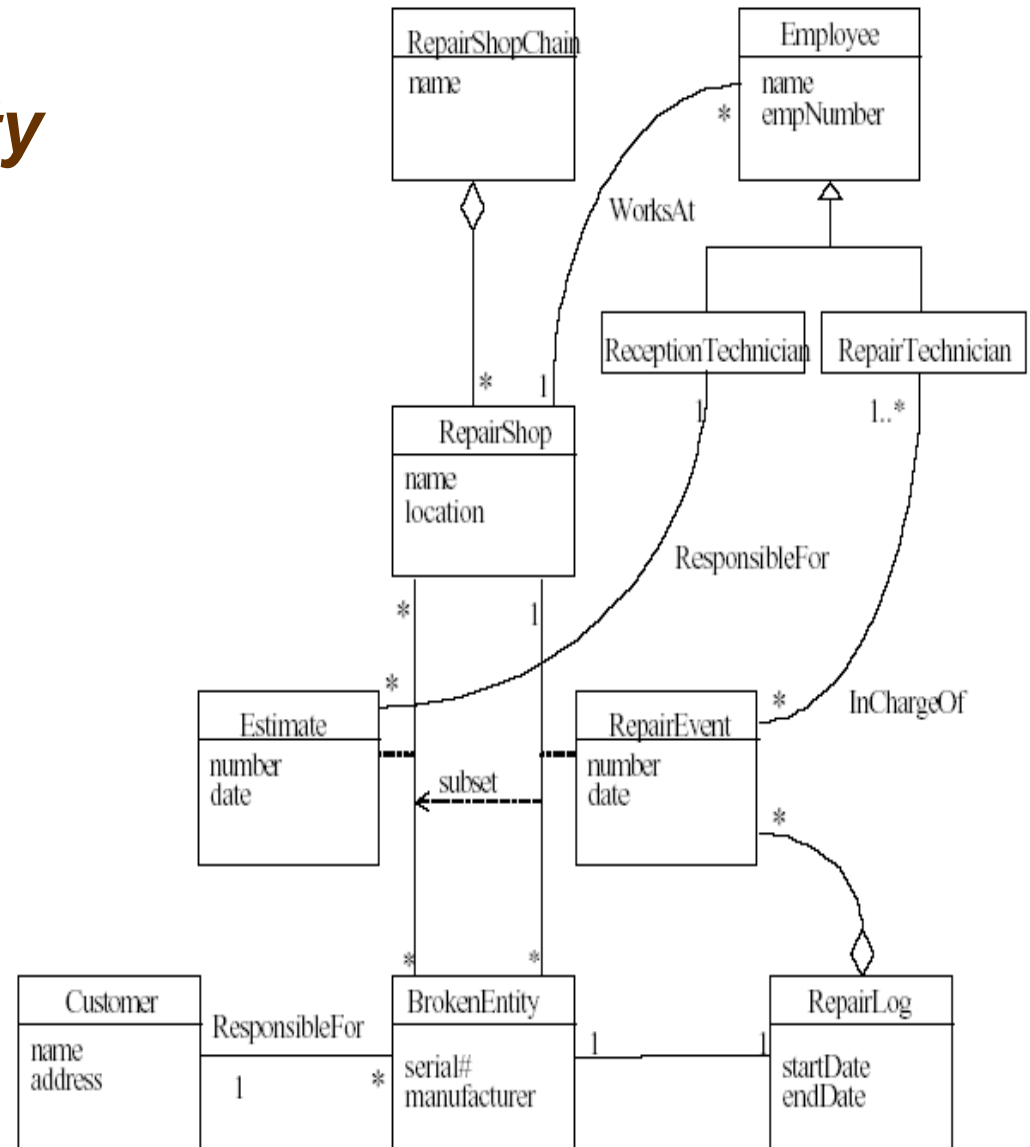


Figure 2. Class diagram for the repair shop pattern

Repair of an Entity

Diagramy sekwencji

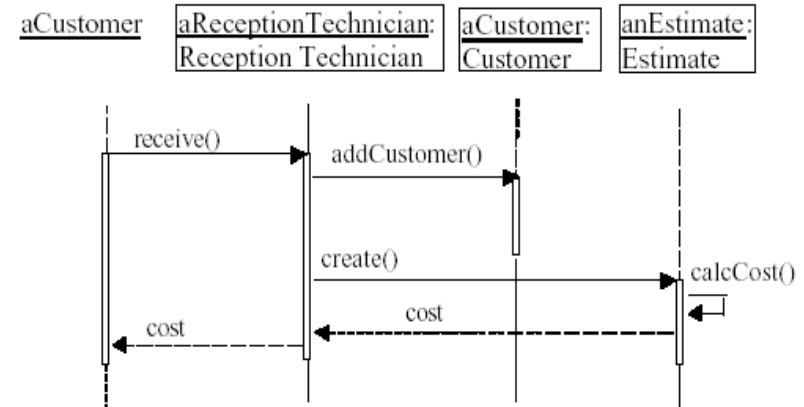


Figure 4. Sequence diagram for getting an estimate for a repair

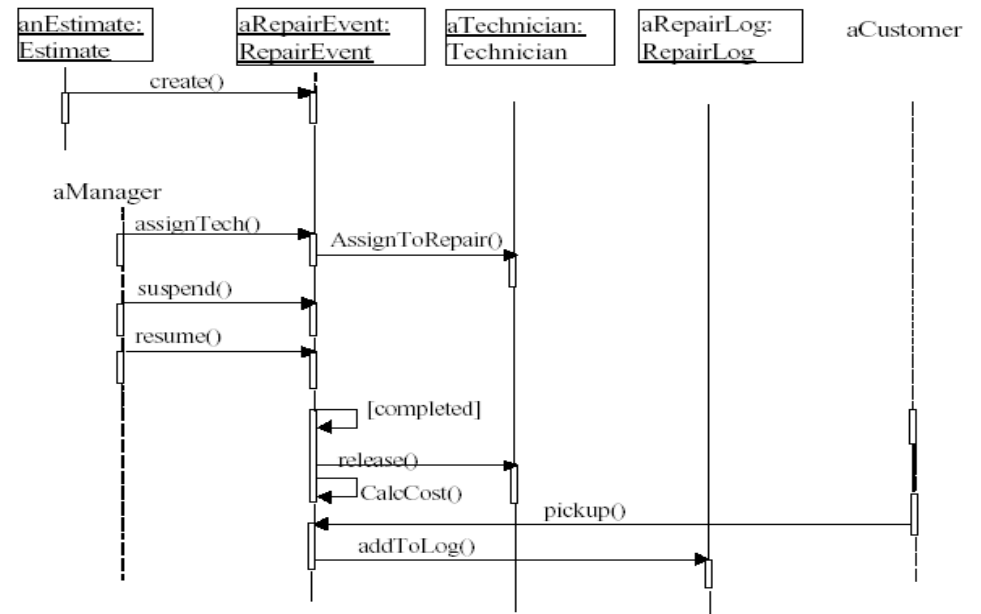


Figure 5. Sequence diagram for assigning repair jobs to technicians

Diagramy

- wycena uszkodzonego sprzętu (Rys.4)
- wyznaczanie prac dla technika naprawiającego (Rys. 5)

Repair of an Entity

Diagram stanów

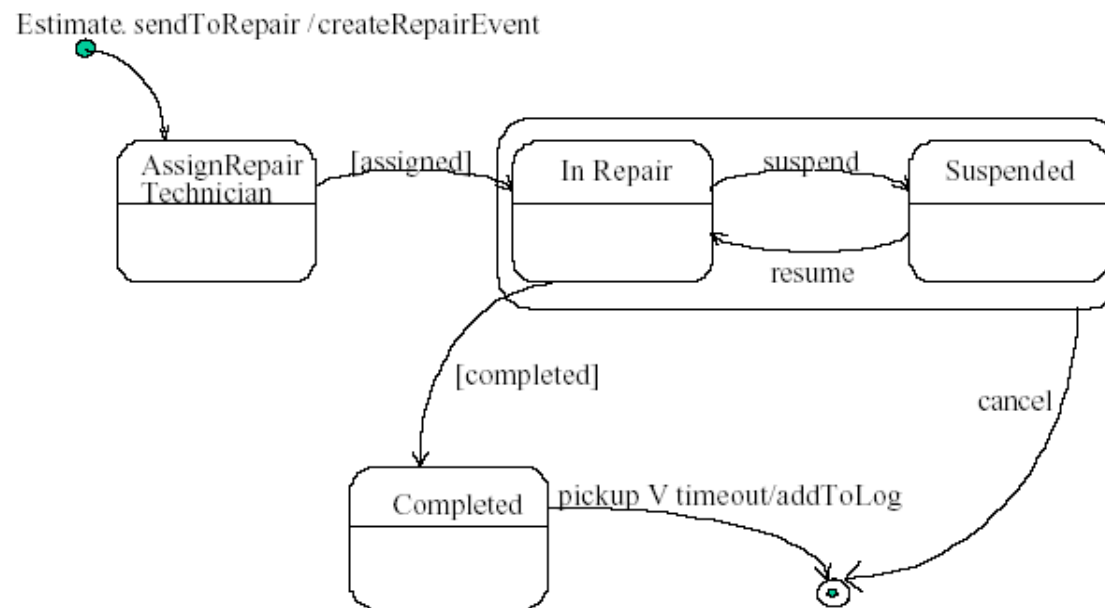


Figure 3. State diagram for class `RepairEvent`

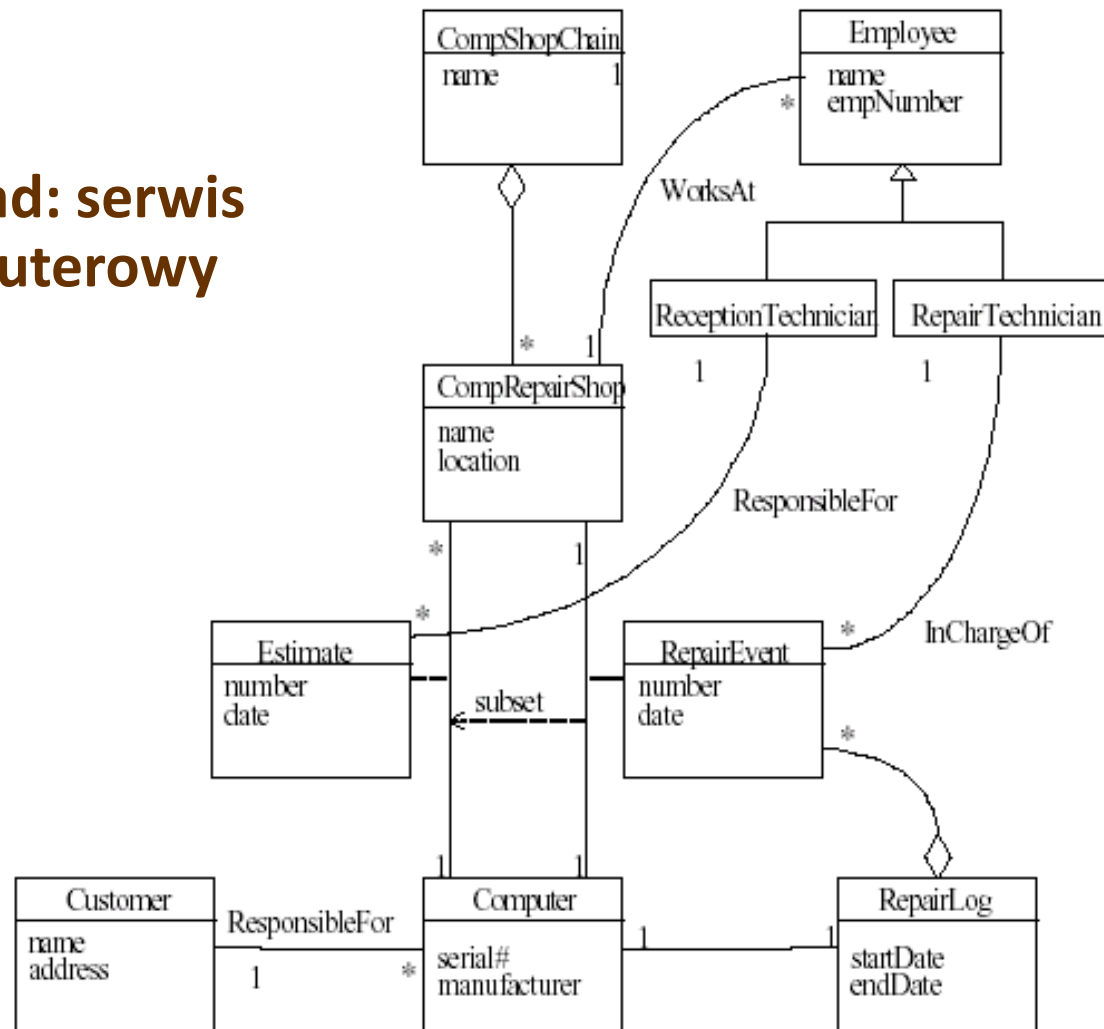
Użycie wzorca

Repair of an Entity

- **Kontekst:** Specjalistyczne zakłady zajmujące się naprawą sprzętu różnych klas, np.: komputery, sprzęt AGD, samochody.
 - Klient zleca naprawę samochodu w warsztacie samochodowym
 - Klient oddaje uszkodzony komputer do serwisu
 - Firma zleca naprawę jakiejś maszyny
 - Klient zamawia naprawę lodówki
- **Ograniczenia wzorca:** Model nie uwzględnia następujących aspektów:
 - Polityki płatności
 - Jak zająć się różnymi klientami np. pojedynczą osobą a zbiorowością
 - Jak zamówić części niezbędne do naprawy
 - Jak ustawić w kolejce klienta, kiedy żadni wolni technicy nie są dostępni
 - W jaki sposób odesłać do klienta naprawiony produkt, kiedy wysyłka jest wymagana
 - Klient umawia się z technikiem, by przynieść uszkodzony przedmiot albo by technik przyszedł do domu jeśli jest to konieczne

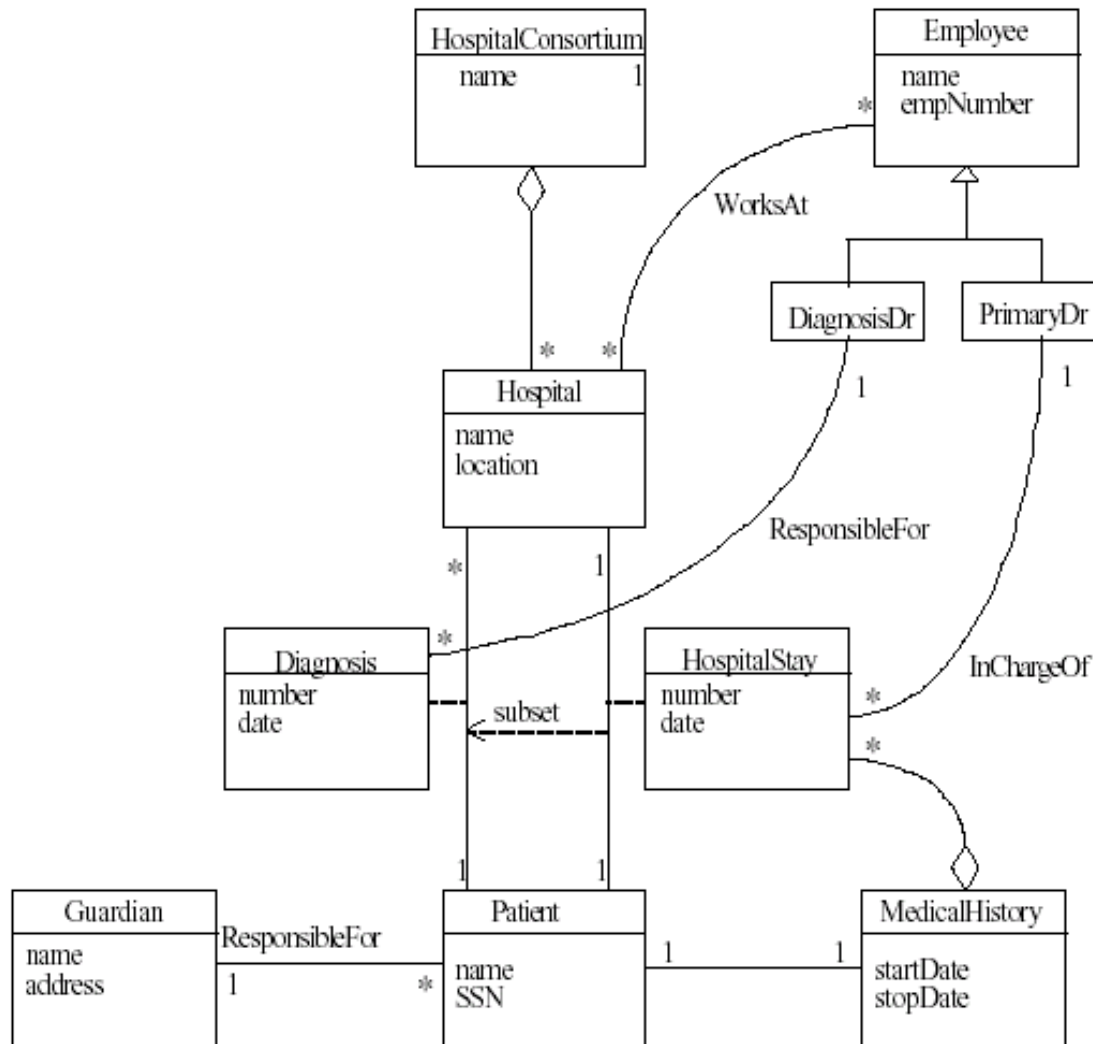
Wykorzystanie wzorca *Repair of an Entity*

Przykład: serwis komputerowy



Wykorzystanie wzorca *Repair of an Entity*

Inna analogia: szpital



Artykuł E. B. Fernandez and X. Yuana:
„*Semantic Analysis Patterns*” ukazuje też
wykorzystanie wzorca *Repair of an Entity*
przy naborze do organizacji (np. uczelni)
jak i ideę uogólnienia wzorca.