PROJEKTOWANIE SYSTEMÓW INFORMATYCZNYCH

6.11.2026

PLAN WYKŁADU

01

Wprowadzenie

Czym zajmuje się projektowanie systemów informatycznych?
Ewolucja metod projektowania.

02

Diagramy behawioralne

Diagram przypadków użycia
Diagram aktywności
Diagram stanów

03

Diagramy strukturalne

Diagram klas
Diagram obiektów
Diagram komponentów i wdrożeniowy

04

Diagramy interakcyjne

Diagram sekwencji
Diagram komunikacji

CEL WYKŁADU

"Design is not just what it looks like and feels like. Design is how it works."

Steve Jobs

1

Przedstawienie podstaw projektowania systemów informatycznych.

2

Omówienie znaczenia i zastosowania diagramów UML w procesie projektowania.

3

Przygotowanie do tworzenia i interpretacji diagramów UML.

Czym jest projektowanie systemów informatycznych?

Projektowanie systemów informatycznych to proces tworzenia struktury, funkcjonalności i interakcji systemu w celu spełnienia określonych wymagań biznesowych.

ANALIZA WYMAGAŃ

Minimalizacja ryzyka błędów

TWORZENIE MODELI SYSTEMU

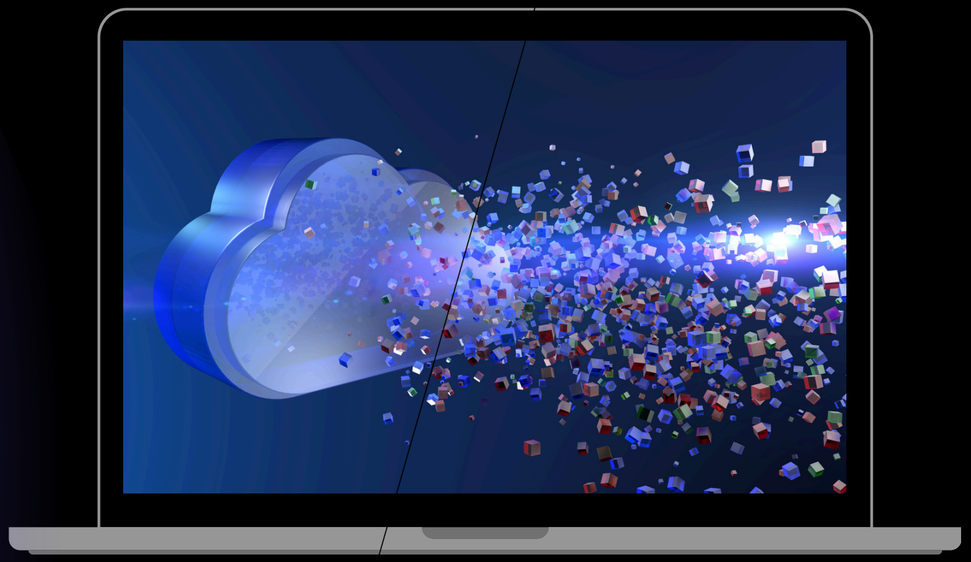
Lepsza komunikacja w zespole projektowym.

PLANOWANIE ARCHITEKTURY

Ułatwienie wdrożenia i utrzymania systemu.



REALLGREATSITE



System informatyczny

Zbiór współpracujących ze sobą komponentów, takich jak sprzęt, oprogramowanie, dane, ludzie i procesy, które wspólnie realizują określone cele.

01 **Systemy ERP**

Finanse i księgowość, KAdry i płace, Human Resources, Produkcja, Magazynowanie, Logistyka, Sprzedaż, Marketing

02 **e-commerce**

sprzedaż internetowa, zarządzanie produktami, płatności, logistyka, drop-shipping, analizy, integracje...

03 **CRM**

Zarządzanie danymi klientów, Automatyzacja procesów marketingowych, Sprzedaż, Komunikacja z klientami, Integracje...

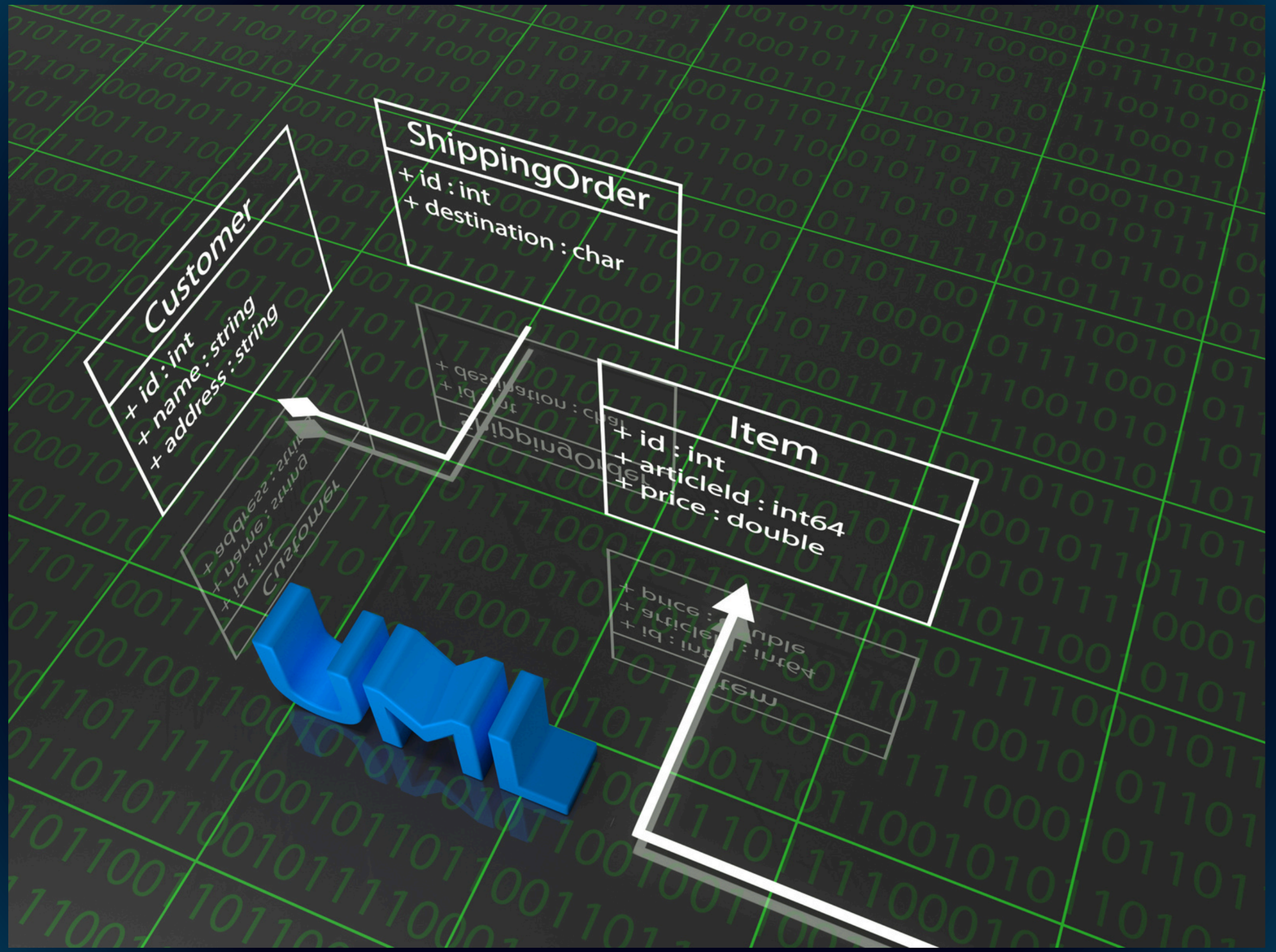
DLACZEGO PROJEKTOWANIE JEST WAŻNE?

Efektywność
Ułatwia spełnienie wymagań biznesowych.

Komunikacja
Tworzy wspólny język między programistami, analitykami i interesariuszami.

Jakość
Pomaga uniknąć problemów technicznych i funkcjonalnych na późniejszych etapach.




Customer
+ id : int
+ name : string
+ address : string
ShippingOrder
+ id : int
+ destination : char
Item
+ id : int
+ articleId : int64
+ price : double
UML

UML (Unified Modeling Language) jest językiem służącym do graficznej wizualizacji, specyfikowania oraz dokumentowania składników systemów informatycznych

UML to notacja umożliwiająca zaprezentowanie systemu w postaci diagramów, który byłby jednakowo interpretowany i zrozumiały dla wszystkich członków zespołu projektowego.

Modele zapisane w języku UML prezentują system od ogółu do szczegółu, umożliwiając oglądanie modelu systemu z wybraną w danym momencie szczegółowością.

Powstał w latach 90. dzięki pracy Grady’ego Boocha, Jamesa Rumbaugha i Ivara Jacobsona.

Historia

2.5.1	December 2017
2.4.1	July 2011
2.3	May 2010
2.2	January 2009
2.1.2	October 2007
2.0	July 2005
1.5	March 2003
1.4	September 2001
1.3	February 2000
1.2	July 1999
1.1	December 1997

01 został opracowany przez Grady Booch, Ivar Jacobson i James Rumbaugh w Rational Software w latach 1994–1995,

02 w styczniu 1997 powstaje wersja 1.0 jako propozycja standardu OMG

03 w sierpniu 2003 została zaprezentowana wersja 2.0

04 najnowsza wersja (2.5.1) pochodzi z 2017 roku

UML



OOSE

OOAD (G. Booch) – jest to metodyka użyteczna w projektowaniu oraz określaniu związków ze środowiskiem implementacji, nie wspiera jednak dostatecznie dobrze fazy rozpoznania i analizy wymagań użytkowników.



OOAD

OOSE (I. Jacobson) – jest to metodyka użyteczna w modelowaniu aspektu użytkowników i cyklu życiowego systemu, nie wspiera jednak w wystarczającym stopniu implementacji.



OMT

OMT (J. Rumbaugh) – jest to metodyka użyteczna w modelowaniu dziedziny przedmiotowej, nie wspiera jednak w wystarczającym stopniu modelowania użytkowników systemu oraz implementacji.

DIAGRAMY UML

STRUKTURALNE

- Diagram pakietów
- Diagram klas
- Diagram obiektów
- Diagram struktur złożonych
- Diagram komponentów
- Diagram wdrożenia

BEHAWIORALNE

- Diagram przypadków użycia
- Diagram aktywności
- Diagram maszyny stanowej
- Diagram komunikacji
- Diagram sekwencji
- Diagram czasu
- Diagram interakcji

UML, CZYLI CO?

NARZĘDZIE

To jak go użyjesz zależy od Ciebie

METODYKA

Opisuje zasady, pamiętaj by sie ich trzymać

SPOSÓB ANALIZY I PROJEKTOWANIA

Należy go ćwiczyć by być ekspertem

PYTANIA NA START :-)



1. Jak systemy informatyczne wpływają na biznes?
2. Jakie wyzwania napotykamy podczas projektowania systemów?
3. Dlaczego diagramy UML są kluczowe w projektowaniu?

BRAIN BREAK

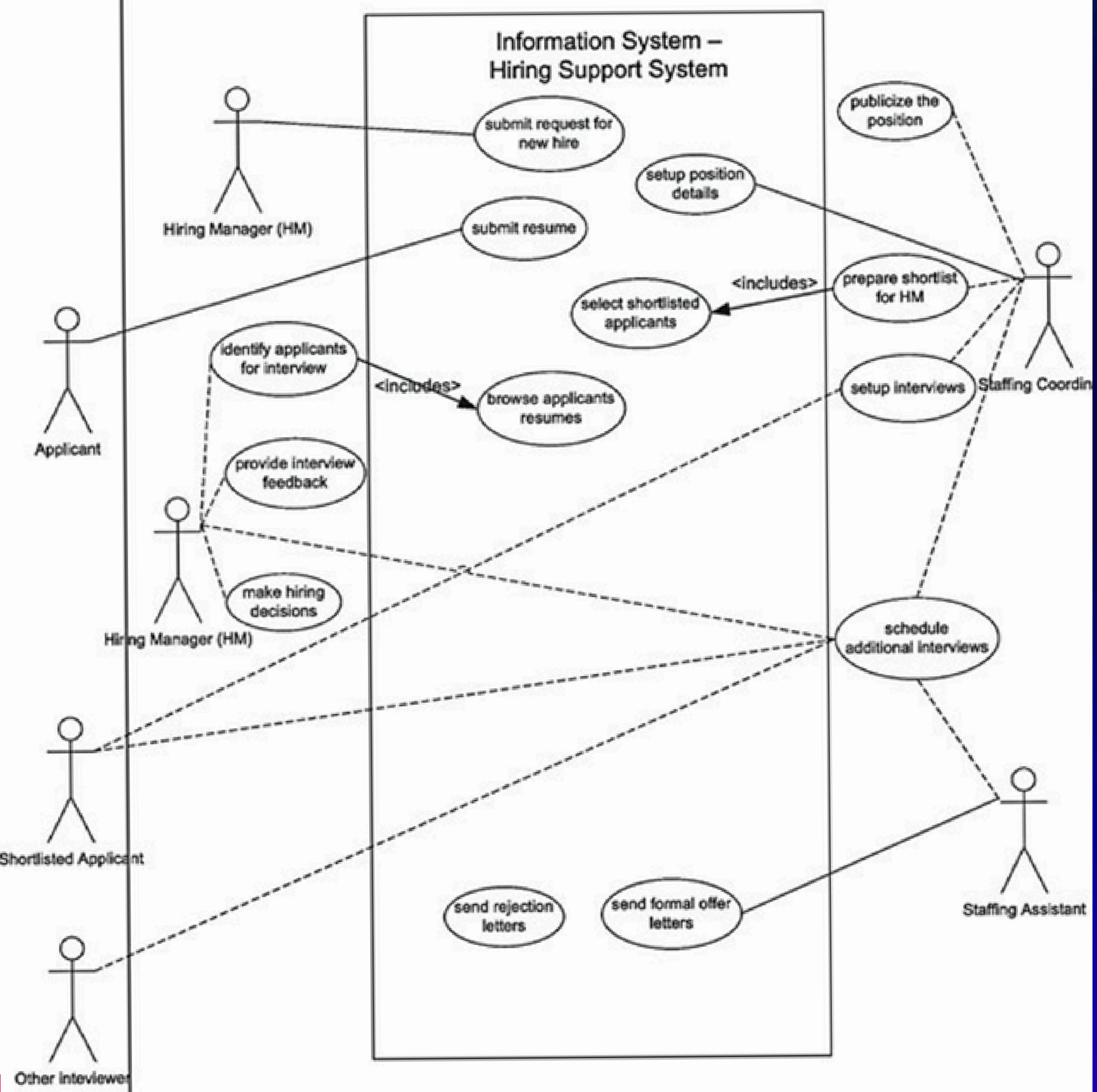


Czy każdy system wymaga takiego samego podejścia do projektowania?

DIAGRAMY BEHAWIORALNE

- Diagramy behawioralne pokazują, jak system działa w odpowiedzi na zdarzenia lub interakcje.

- Są kluczowe do zrozumienia procesów biznesowych, przepływu danych i zależności w systemie.



Przypadki użycia (use case) są opisami sekwencji interakcji, jakie zachodzą między systemem a użytkownikiem (aktorem), gdzie ważne jest, aby użytkownik osiągnął wyznaczony cel.

Przypadki użycia modelują zachowania system i reprezentują kompletną jednostkę funkcjonalności, która ma wartość biznesową dla użytkownika

Przypadki użycia stosujemy w celu formalnego opisu i dokumentowania wymagań funkcjonalnych.

Przypadki użycia są niezależne od technologii implementacyjnej systemu informatycznego

Przypadki użycia

Aktor

Aktor jest to zbiór ról odgrywanych przez użytkowników w trakcie używania system. Aktor jest bytem zewnętrznym w stosunku do systemu i wchodzi z systemem w reakcję, aby osiągnąć określony cel.

Cel

Cel opisuje wartość biznesową, którą realizuje przypadek użycia. Cel determinuje znaczenie poszczególnych działań w przypadku użycia.

System

System definiuje granice dla realizacji przypadku użycia.

Scenariusz

Scenariusz reprezentuje porządek zdarzeń (sekwencję) pomiędzy aktorem i systemem, których zadaniem jest osiągnięcie celu biznesowego.

WYMAGANIA

Wymaganie jest definiowane i formalizowane przez użytkownika

PRZYPADKI UŻYCIA

Przypadek użycia jest definiowany przez analityka biznesowego na podstawie wymagań



Szablon przypadku użycia

- Formularz strukturalizuje definicję przypadku użycia poprzez dostarczenie schematu opisu, zawierającego określone pola.
- Pola formularza prezentują bloki składowe każdego przypadku użycia połączone w predefiniowany i uporządkowany sposób.

Nazwa - definiuje cel, który przypadek użycia ma zrealizować.

ID - Unikalny, numeryczny identyfikator PU.

System – system lub pakiet systemu do którego należy przypadek użycia, do którego należy.

Priorytet - Decyduje o kolejności i istotności projektowania i implementacji PU.

Streszczenie - Opis przypadku użycia, rozszerzający jego nazwę i wprowadzający do scenariusza.

Główny aktor - Jest to aktor, którego cel jest realizowany przez przypadek użycia.

Aktor wspierający - Wspomaga aktora głównego w realizacji celu przypadku użycia.

Interesariusze - byt, człowiek lub inny system, który ma interes w prawidłowej realizacji przypadku użycia i partycypuje w korzyściach wynikających z realizacji danego PU.

Warunek wejściowy - Definiuje stan systemu zanim przypadek użycia rozpocznie swój cykl życia. defines the state of the system before a use case can start;

Warunek wyjściowy - definiuje stan systemu po zakończeniu realizacji przypadku użycia.

Wyzwalacz (trigger) - Zdarzenie inicjujące rozpoczęcie działania PU.

Scenariusz – sekwencja, kroków, które musi wykonać aktor w celu realizacji przypadku użycia.

Wymagania niefunkcjonalne - Tylko w przypadku kiedy wymagania niefunkcjonalne dotyczą realizacji PU powinny być opisane za pomocą omawianego szablonu

Status - Pomaga utrzymać kontrolę nad etapem realizacji danego PU.

Scenariusz przypadku użycia

Każdy krok jawi się jako interakcja, ale w rzeczywistości jest to transakcja pomiędzy aktorem i systemem. To oznacza, że w każdym kroku:

- Aktor wysyła żądanie do systemu,
- System waliduje żądanie,
- System zmienia swój stan jako wynik walidacji i potem,
- System odpowiada.

Kroki mogą być powtarzane

- Pojedynczy krok lub ich sekwencja mogą być powtarzane dopóki jest realizowanych określony warunek. W przykładzie dotyczącym sklepu spożywczego, kasjer skanuje towary tak długo, jak w koszyku klienta jeszcze jakieś pozostają.

Dany krok może komunikować się z innym przypadkiem użycia.

- Każdy krok może komunikować się z innym przypadkiem użycia w celu ukończenia danej funkcji.

Scenariusz podstawowy i alternatywny

Scenariusz podstawowy:	1. Klient wprowadza kartę bankomatową. 2. Klient wprowadza hasło. 3. System weryfikuje hasło. 4. System wyświetla listę transakcji, które klient może zrealizować. 5. Klient wybiera typ transakcji.

Scenariusz alternatywny/Wyjątki:	a.Pacjent jest nowy, rejestratorka kieruje pacjenta do procesu rejestracji. b. Pacjent nie jest nowy, ale zmieniły się dane dotyczące ubezpieczenia. Rejestratorka kieruje pacjenta do procesu rejestracji. c. Pacjent zgubił dokument RMUA (identyfikator ubezpieczenia). Rejestratorka kieruje pacjenta do procesu rejestracji.

Opisuje najlepszy dla PU scenariusz

Scenariusz alternatywny identyfikuje wyjątki, oraz scenariusz negatywny

Podproces

Scenariusz podstawowy:	**1.** Rejestratorka aktualizuje dane osobowe pacjenta.
Podproces:	1. Rejestratorka otwiera formularz nowego pacjenta. 2. Rejestratorka wprowadza lub aktualizuje adres pacjenta. 3. Rejestratorka wprowadza lub aktualizuje nr telefonu pacjenta.. 4. Rejestratorka wprowadza lub aktualizuje dane osobowe, dane najbliższych itp. pacjenta.

Podproces identyfikuje dodatkowe,szczegółowe kroki dla określonego etapu w procesie podstawowym

Modelowanie diagramu przypadków użycia

Diagramy przypadków użycia (ang. use case) reprezentują wymagania funkcjonalne projektowanego systemu.

Diagramy przypadków użycia ilustrują interakcję pomiędzy systemem i podmiotami zewnętrznymi do system czyli aktorami.

Diagram przypadków użycia tworzony jest zazwyczaj w początkowych fazach modelowania.



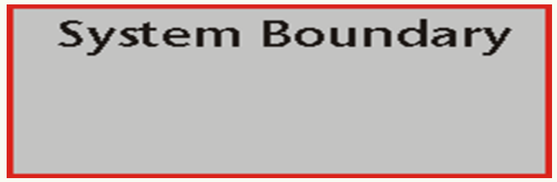
Diagram przypadków użycia jest meta modelem, który:

- definiuje granice modelowanego systemu,
- określa jego kontekst,
- definiuje użytkowników systemu i jednostki zewnętrzne,
- przedstawia funkcje dostępne dla użytkowników, określa powiązania i zależności pomiędzy nimi.

Notacja w diagramie przypadków użycia



Prosta figura identyfikująca aktora. Więcej niż jeden aktor może być powiązany z jednym przypadkiem użycia.

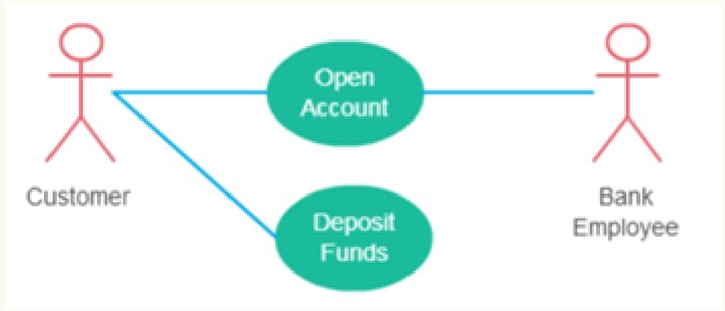


Kwadrat oznacza granice systemu lub jego zakres. Z punktu widzenia PU encja występująca poza granicami systemu może być jedynie aktorem.



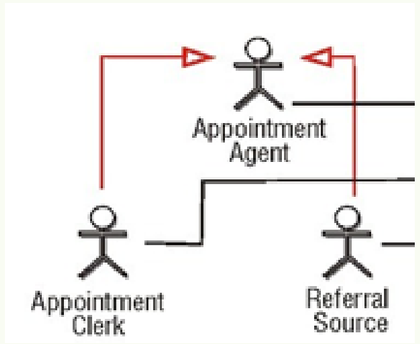
Owal lub elipsa reprezentuje przypadek użycia. Występują tylko w ramach granic systemu.

Notacja w diagramie przypadków użycia

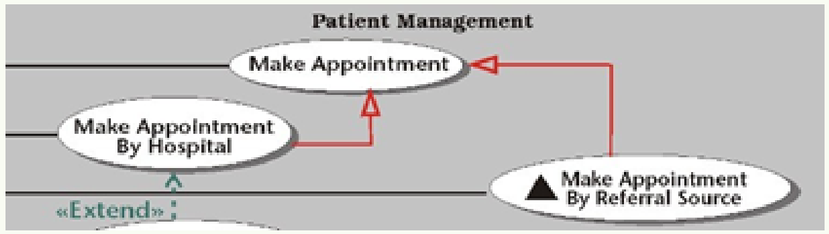


Asocjacja – związek wskazujący na to, że aktor używa danego przypadku użycia.

- Aktor musi być powiązany z co najmniej jednym przypadkiem użycia.
- Aktor może być powiązany z wieloma przypadkami użycia.

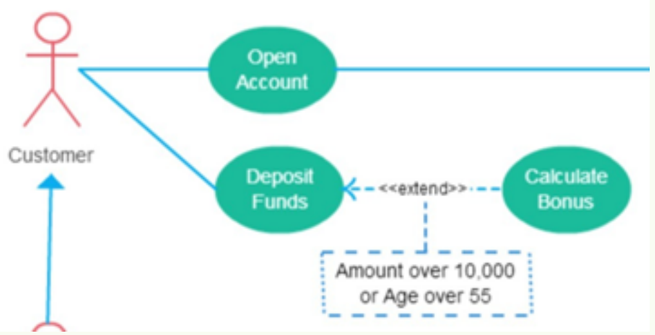


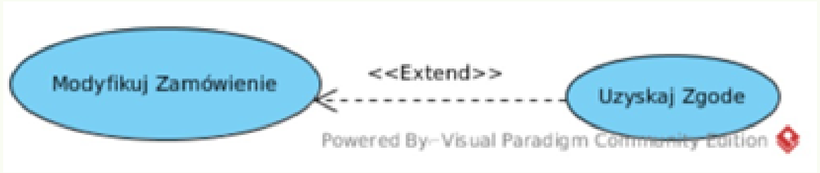
Generalizacja aktorów oznacza, że jeden aktor może dziedziczyć rolę innego aktora. Potomek dziedziczy wszystkie przypadki użycia przodka. Potomek ma jeden lub więcej przypadków użycia, które są specyficzne dla tej roli.



Generalizacja przypadków użycia – zachowanie(funkcjonalność) przodka jest dziedziczone przez potomka. Jest to używane, gdy istnieje wspólne zachowanie między dwoma przypadkami użycia. Uogólnienie ma postać strzałki z linią ciągłą i zamkniętym grotem.

Notacja w diagramie przypadków użycia



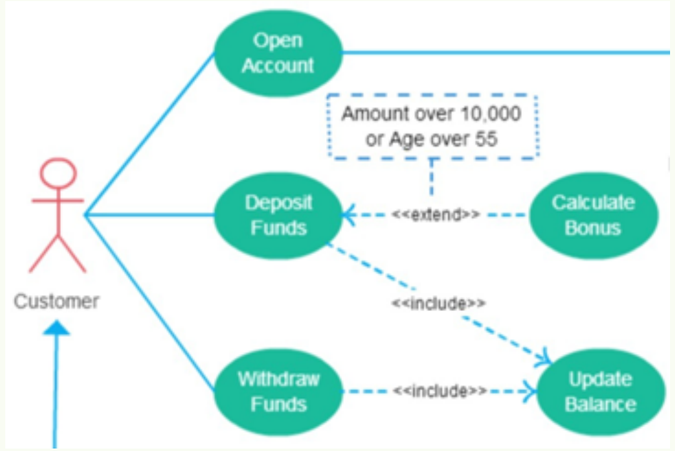


Relacja rozszerzenia **<<extend>>** pokazuje przypadek użycia utworzony w celu rozszerzenia funkcjonalności podstawowego PU. Relacja ta pozwala na wydzielenie przypadku użycia, który w pewnych sytuacjach może zostać wzbogacony o dodatkowe opcje.

- Rozszerzający przypadek użycia jest zależny od rozszerzonego (podstawowego) przypadku użycia.
- Rozszerzający przypadek użycia jest zazwyczaj opcjonalny i może być uruchamiany warunkowo.
- Rozszerzony (podstawowy) przypadek użycia musi mieć znaczenie sam w sobie. Oznacza to, że powinien być niezależny i nie może polegać na zachowaniu rozszerzającego przypadku użycia.

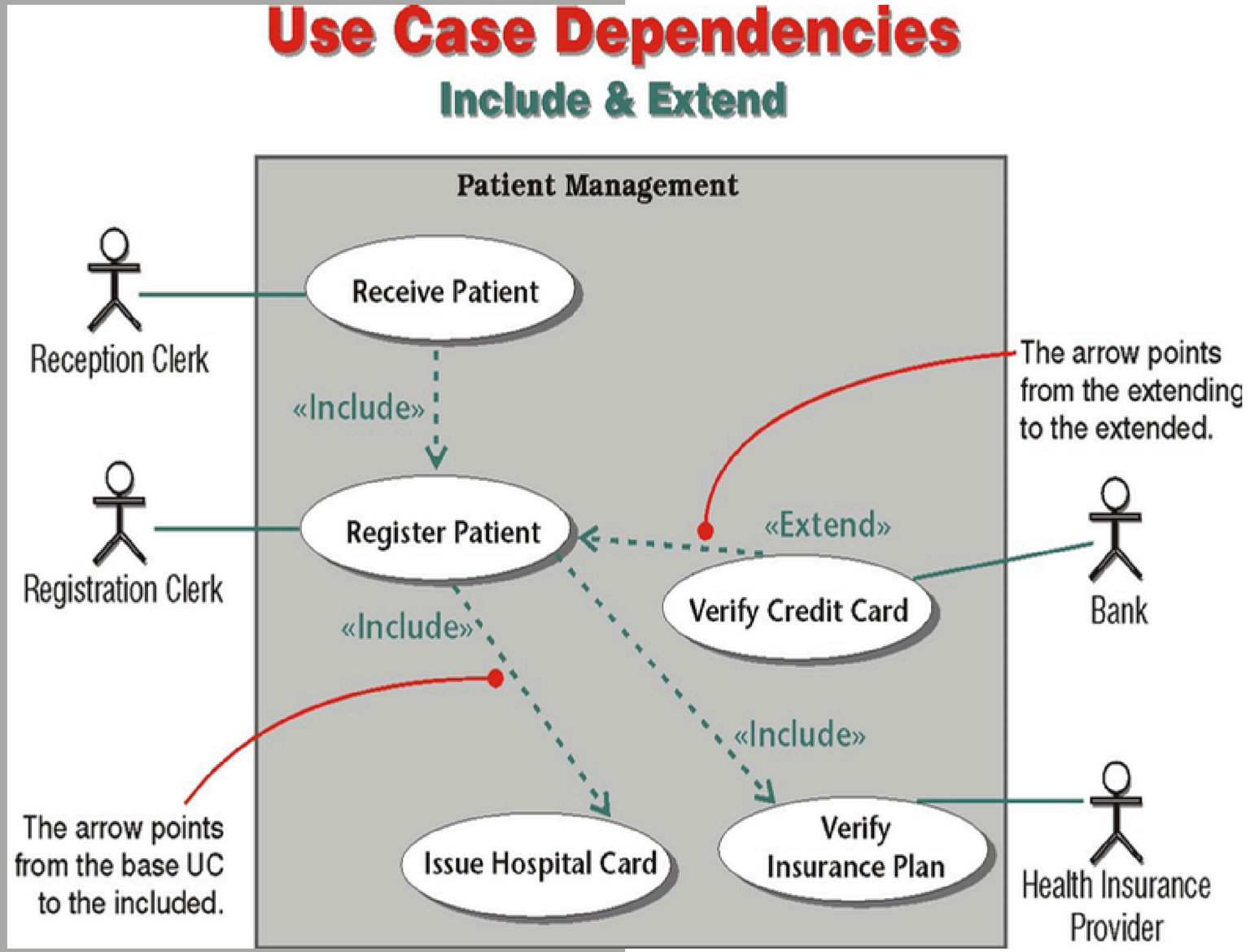
Relacja rozszerzenia ma postać przerywanej strzałki ze stereotypem «extend», biegnącej od przypadku użycia rozszerzającego do rozszerzanego.

Notacja w diagramie przypadków użycia

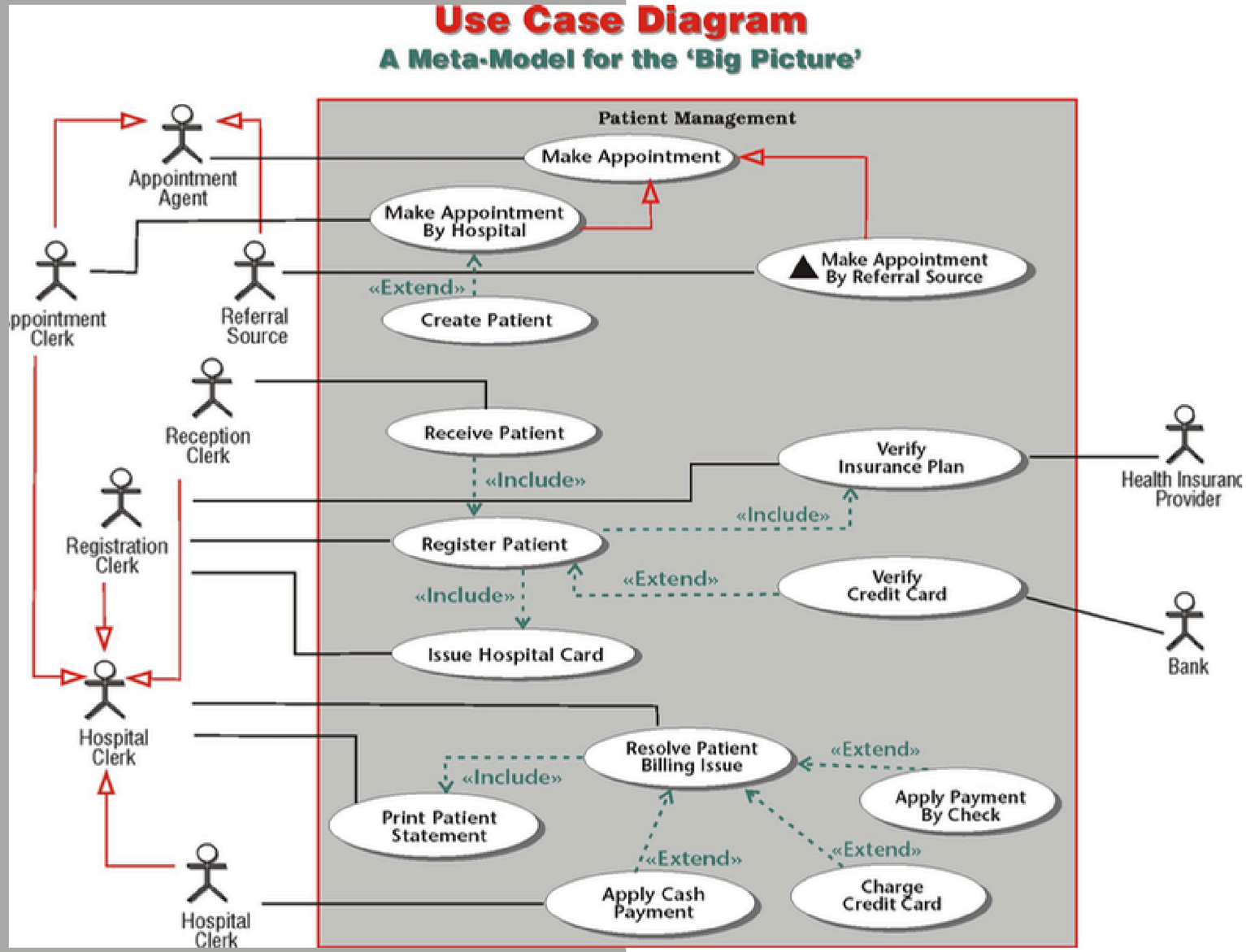


Relacja zawierania **<<include>>** pokazuje relację w której podstawowy PU wykorzystuje funkcjonalność innego przypadku użycia.

- Przypadek użycia może zawierać funkcjonalność innych przypadków użycia jako część ich normalnej obsługi.
- Zawierany przypadek użycia nie jest wykonywany samodzielnie; wykonywany jest zawsze, gdy wykonywany jest zawierający przypadkek użycia.
- Relację zawierania stosuje się wówczas, gdy z kilku innych przypadków użycia można wydzielić pewną część wspólną.
- Relacja zawierania ma postać przerywanej strzałki ze stereotypem **«include»**, biegnącej od przypadku użycia zawierającego do zawieranego.



Postarajmy sie odczytać diagram :)



Postarajmy sie odczytać diagram :)

ZASTOSOWANIE PRZYPADKÓW UŻYCIA

01

Gromadzenie wymagań

Przykłady użycia dostarczają podstawowych narzędzi do zbierania wymagań w znaczącym kontekście.

02

Śledzenie implementacji wymagań

Przypadki użycia i ich dokumenty wspierające są głównymi źródłami do śledzenia wymagań.

03

Zasady biznesowe

Przypadki użycia są podstawą do zbierania reguł biznesowych.

04

Zachowania systemu

Zewnętrzne zachowanie dowolnego systemu otwartego można skutecznie uchwycić za pomocą przypadków użycia.

BRAIN BREAK

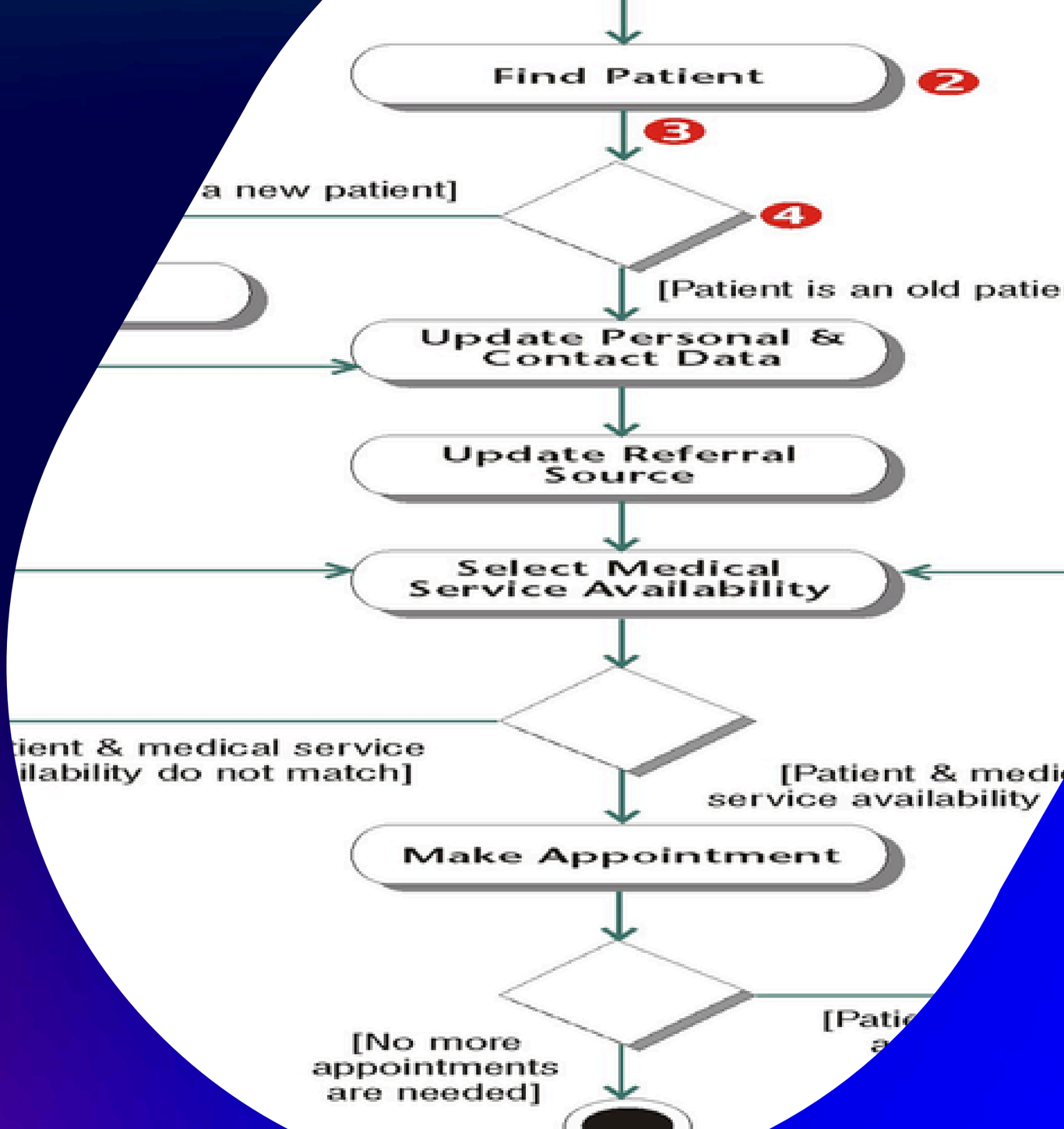


A gdybyś to Ty był aktorem...

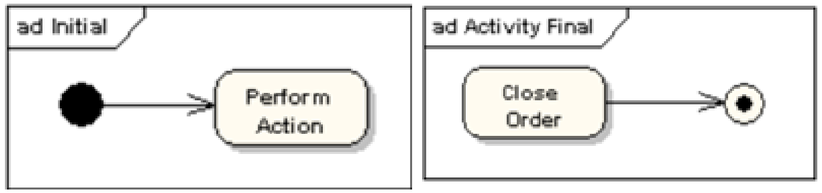
Diagram aktywności obrazuje i modeluje przypływy pracy od działania do działania (activity).

Diagram prezentuje wizualną, dynamiczną postać systemu i jego elementów.

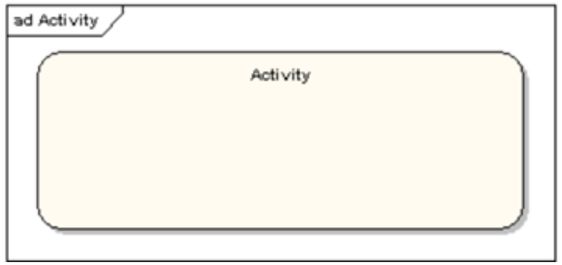
Diagram może być także użyty do modelowania przepływów dla przypadków użycia.



Notacja diagram aktywności

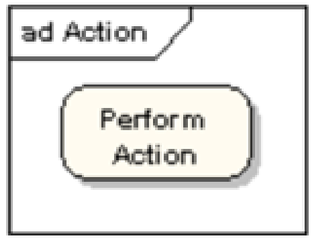


Początkowy i końcowy stan akcji, przedstawiane odpowiednio jako wypełnione koło oraz wypełnione koło w okręgu.

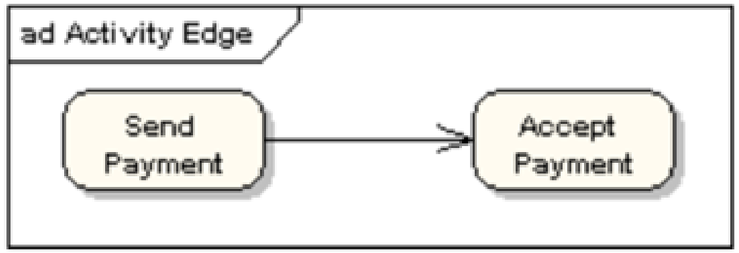


Atywność to specyfikacja sparametryzowanej sekwencji zachowań.
Aktywność jest przedstawiona jako zaokrąglony prostokąt obejmujący wszystkie akcje, przepływy sterowania i inne elementy, które składają się na aktywność.

Notacja diagram aktywności

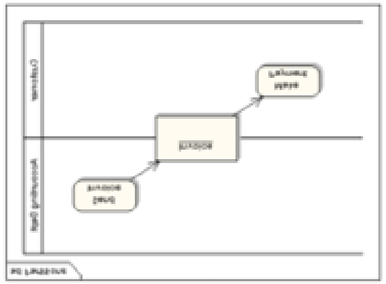


Akcja reprezentuje pojedynczy krok w ramach aktywności. Akcje są oznaczone prostokątami o zaokrąglonych rogach.

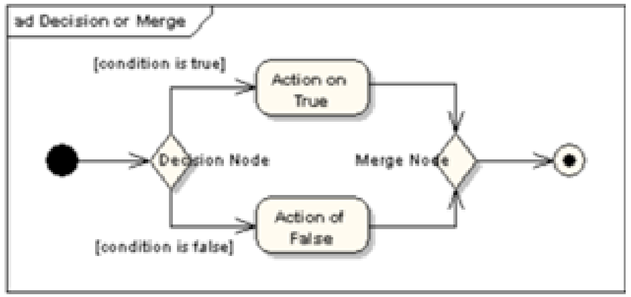


Przejście przepływu sterowania (ciągła strzałka), występujące pomiędzy czynnościami. Zakończenie jednej czynności powoduje rozpoczęcie drugiej.

Notacja diagram aktywności

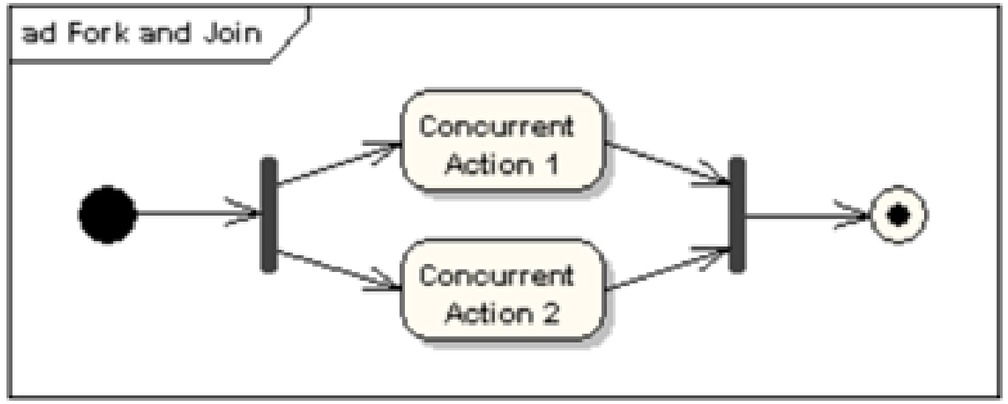


Tory (swimline), rysowane za pomocą linii ciągłych. Służą do określania, który element systemu wykonuje dane akcje. Mogą występować zarówno pionowo jak I poziomo.



Decyzje (obrazowane za pomocą rombów) służące do wyboru jednego przejścia przepływu sterowania.
Odpowiednie wyjścia opisywane są warunkami, które muszą zostać spełnione by dane przejście mogło zajść

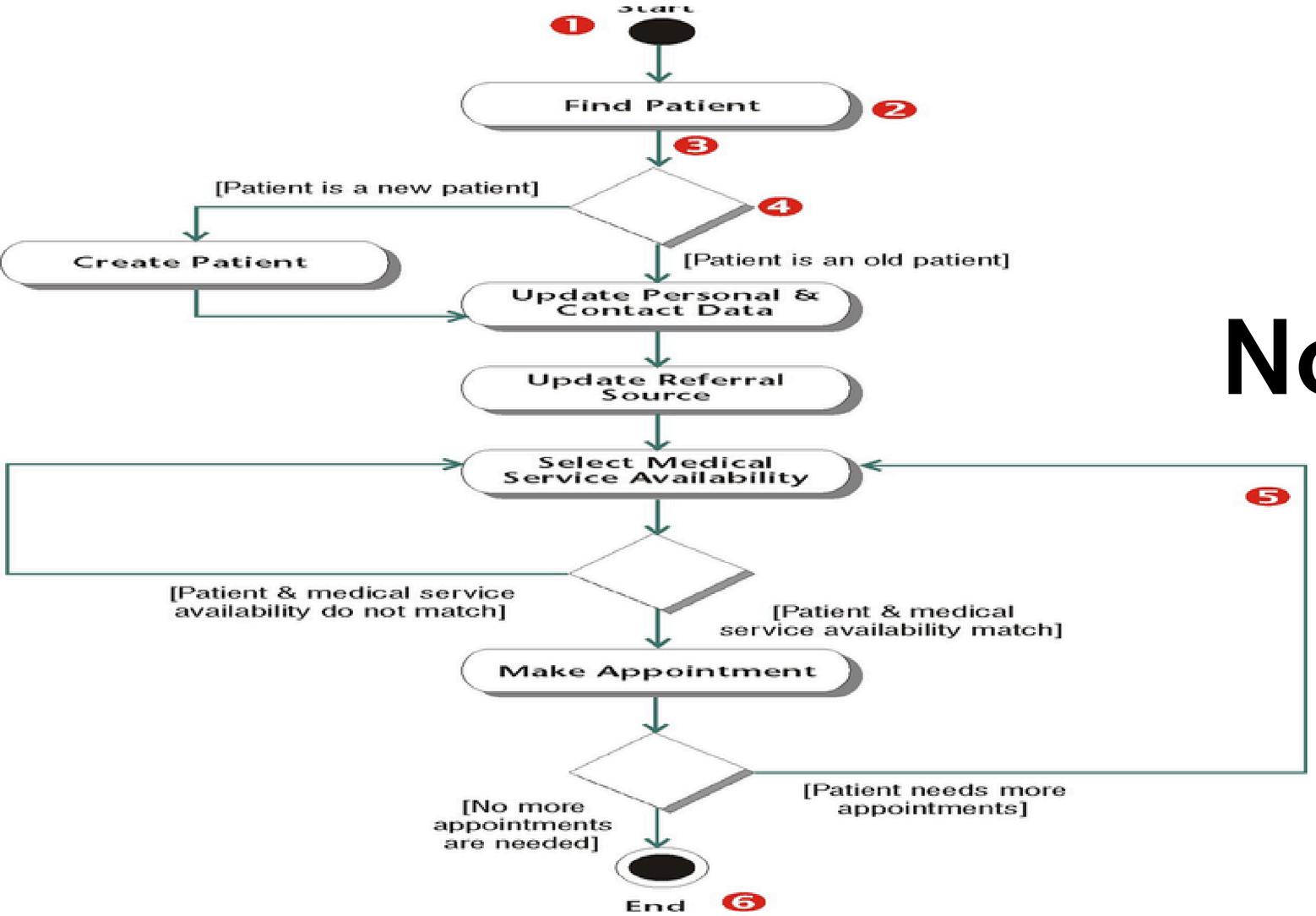
Notacja diagram aktywności



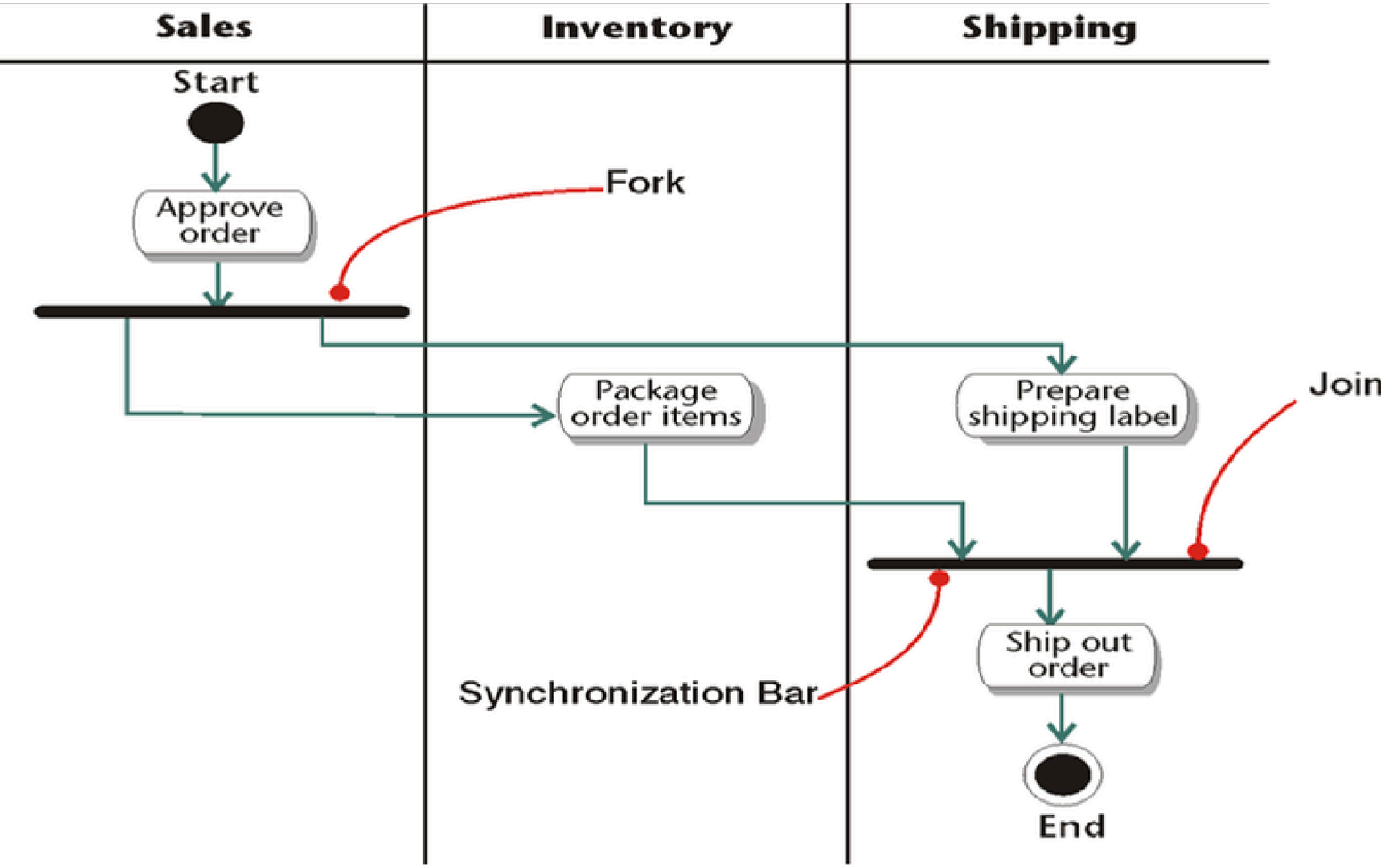
Węzły Fork i Join mają tę samą notację: poziomy lub pionowy pasek (orientacja zależy od tego, czy przepływ sterowania przebiega od lewej do prawej, czy od góry do dołu). Wskazują one początek i koniec współbieżnych wątków sterowania.

Fork różni się od Join tym, że łączy synchronizuje napływy i wytwarza pojedynczy wypływ.
Wypływ z połączenia nie może zostać wykonany, dopóki nie zostaną odebrane wszystkie wpływy.

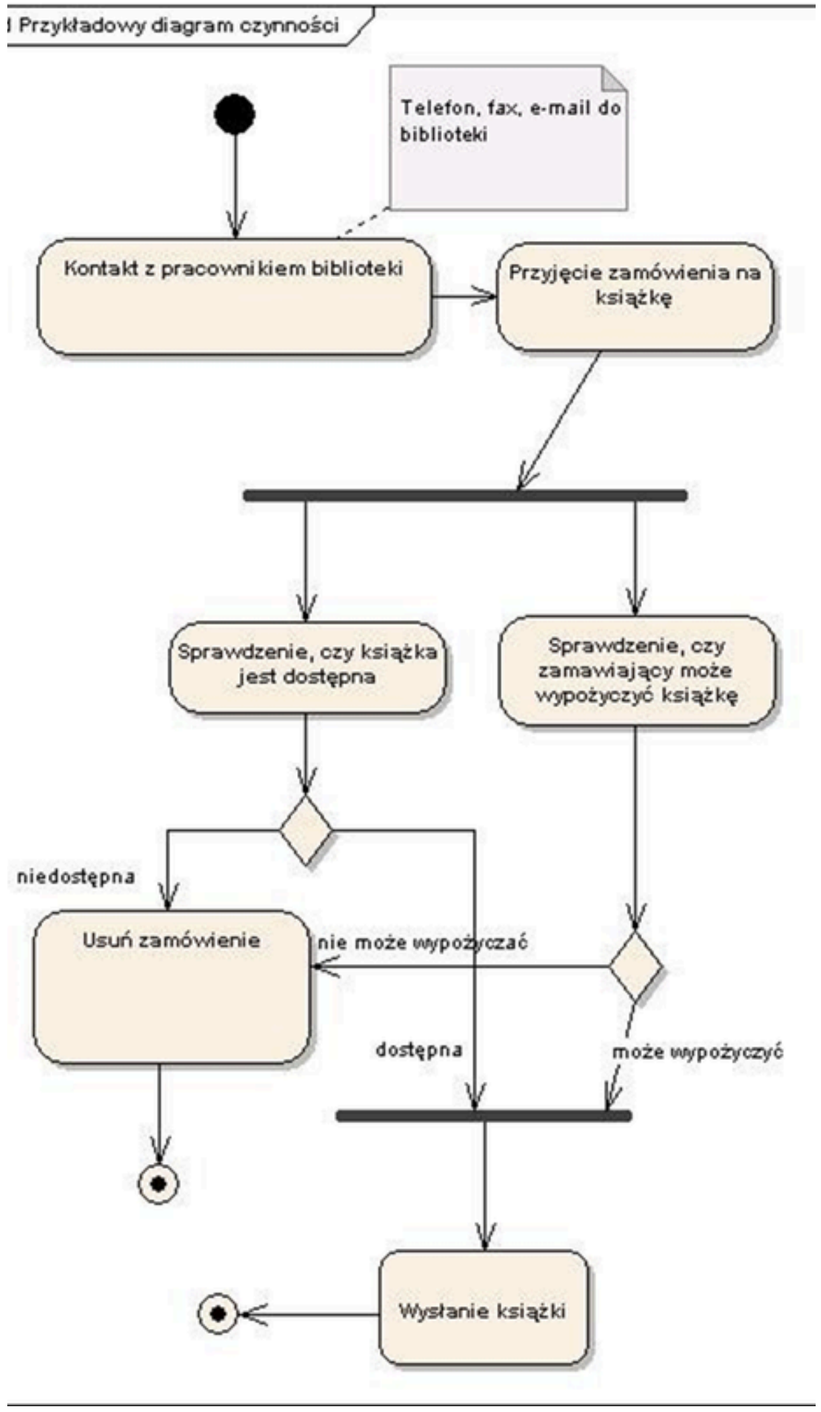
Join przekazuje wszystkie przepływy sterowania bezpośrednio przez nie. Jeśli dwa lub więcej wpływów zostanie odebranych przez symbol łączenia, akcja wskazywana przez jego wypływ jest wykonywana dwa lub więcej razy.



Notacja diagram aktywności



Notacja diagram aktywności



Gdzie jest błąd?

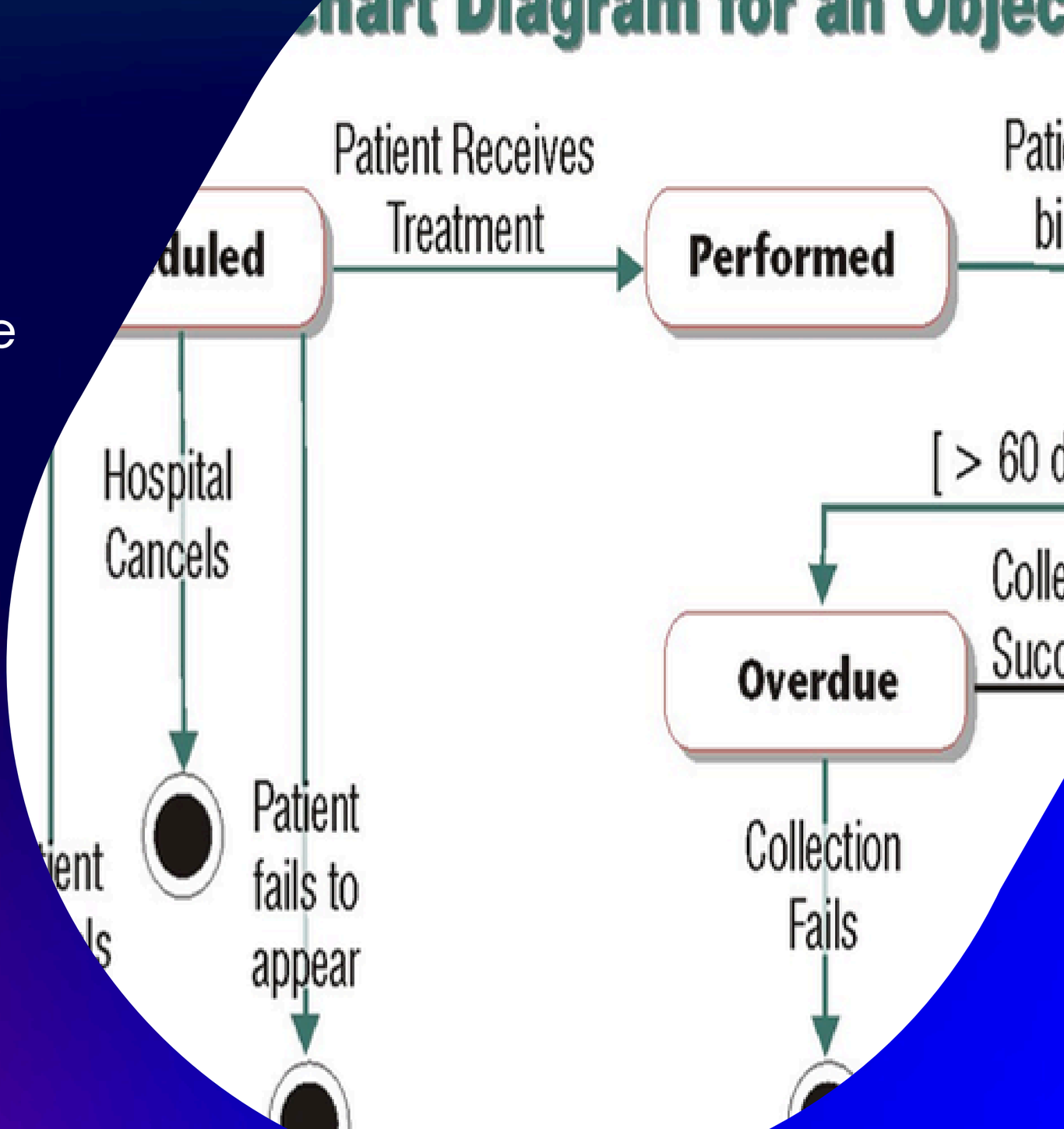
Diagram stanów prezentuje kamienie milowe w cyklu życia systemu w momencie zmiany stanu obiektu

Stan jest to zachowanie obiektu na określonym etapie życia prezentowane z okreslonego punktu widzenia

Jest to “snapshot” obiektu w okreslonej sytuacji i czasie

Diagram składa się ze stanów obiektu oraz z przepływu zdarzeń, które zmianią ten stan

Obiekt, reagując na nadchodzące zdarzenia, jeżeli spełnione są określone warunki, zmienia swój stan i położenie na diagramie stanu



Elementy diagramu stanów

Dodaj więcej tekstu

01 Stan początkowy

02 Stan (pośredni)

03 Zdarzenia / transakcje

04 Stan końcowy

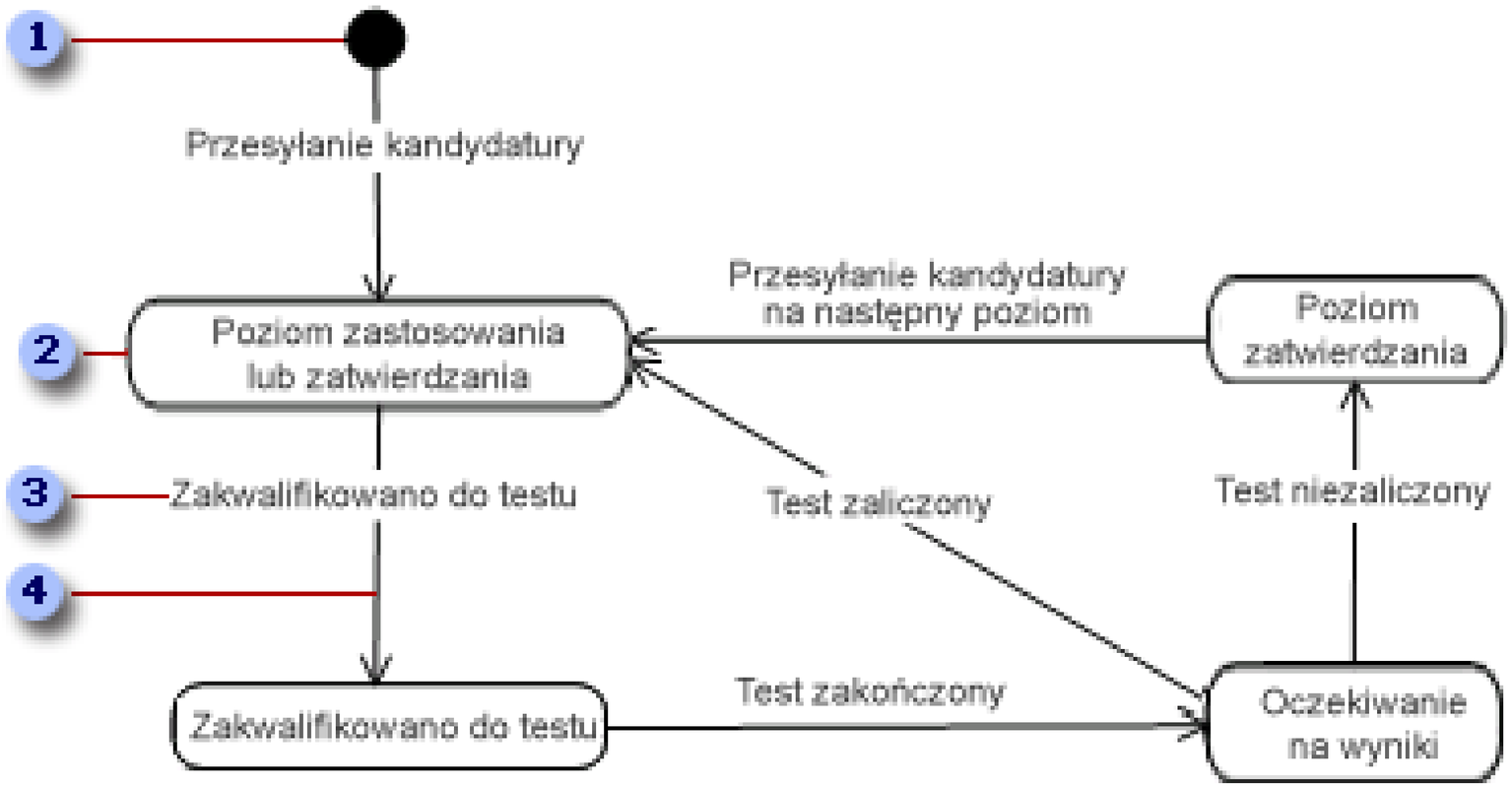


Diagram stanów

Modelowanie zmian stanów obiektów

Pytania

Co robi system?

Jak działa system?

Jak zmienia się stan systemu?

BRAIN BREAK



Czy jesteśmy w stanie kontynuować?