



dr inż. Michał Malinowski
michal.malinowski@warszawa.merito.pl

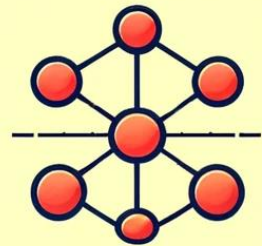


[https://www.linkedin.com/in/
michał-malinowski-malkon/](https://www.linkedin.com/in/michał-malinowski-malkon/)

Modele i diagramy w projekcie systemu informatycznego

Wykład 03

PROJEKTOWANIE SYSTEMÓW
INFORMATYCZNYCH



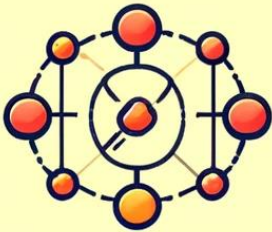
DEFINE NODES
REFATTIOWIPS



MODEL C
AREFITEICSTOHP S



MOLEL NODOS
AND EIOCTIHIRE



APPLYING INTIDEES
Q KUNRONUNTS



APPLIVIG INDEXDES
& ORSSTUCINS



IMPOR IG
CORARCAT



SUIDATING THE GRAP I



oTSGAD



VALIDATINOE GRAPH

Zagadnienia (2h)

- [Język modelowania UML](#)
- [Diagram dekompozycji funkcji](#)
- [Diagramy przypadków użycia](#)
- [Diagram aktywności/czynności](#)
- [Diagram związków encji ERD](#)
- [Modelowanie procesów biznesowych BPMN](#)
- [Narzędzia do modelowania](#)

Biały Dom pilnie domaga się, aby programiści używali języków programowania bezpiecznych dla pamięci.

Luty 2024

Biuro Krajowego Dyrektora ds. Cyberbezpieczeństwa (ONCD)
w Białym Domu



Bezpieczne języki programowania

Automatyczne zarządzanie pamięcią

Eliminuje błędy związane z ręcznym zarządzaniem pamięcią, takie jak wycieki pamięci i podwójne zwalnianie pamięci, co zapobiega wielu potencjalnym problemom bezpieczeństwa.

Bezpieczne typowanie

Pomaga wykrywać błędy na wczesnym etapie (podczas kompilacji), zapobiegając nieprawidłowym operacjom na danych, które mogłyby prowadzić do błędów

Sprawdzanie granic

Dla tablic i innych struktur danych zapobiega przepełnieniom bufora, które mogą prowadzić do nadpisania danych i potencjalnych luk w bezpieczeństwie.

Zapobieganie wstrzyknięciu kodu

Uniemożliwia wstrzyknięcie i wykonanie nieautoryzowanego kodu, na przykład przez odpowiednie sanitowanie danych wejściowych i stosowanie bezpiecznych API.

Model własności i zarządzanie dostępem

Stosowanie modeli własności i pożyczania, które na poziomie kompilacji zapewniają, że dostęp do danych jest zarządzany w sposób bezpieczny, eliminując wyścigi danych i inne błędy związane z konkurencyjnym dostępem.

Immutability (niemutowalność)*

Wymuszanie stosowania niemutowalnych struktur danych pomaga w zapobieganiu błędom wynikającym ze zmian stanu, co jest szczególnie ważne w aplikacjach współbieżnych i rozproszonych.

*Immutability, czyli niemutowalność, to właściwość obiektów (zmiennych) w programowaniu, która nie pozwala na zmianę ich stanu po utworzeniu

Bezpieczne języki programowania

Bezpieczne języki	Pośrednie języki	Niebezpieczne języki
Rust	JavaScript	C
Swift	TypeScript	C++
Go	Kotlin	Assembly
Java	Scala	
C#	Python	
Haskell	Ruby	
Elixir	PHP	

Język modelowania UML

UML (Unified Modeling Language)

to standardowy język modelowania używany w inżynierii oprogramowania do wizualizacji, projektowania i dokumentowania systemów



Cechy

Wizualizacja

Umożliwia graficzną reprezentację struktury i działania systemów, co ułatwia ich zrozumienie i analizę.

Wieloaspektowość

Oferuje różne typy diagramów, które pozwalają na modelowanie różnych aspektów systemu, takich jak struktura, zachowanie i interakcje.

Skalowalność

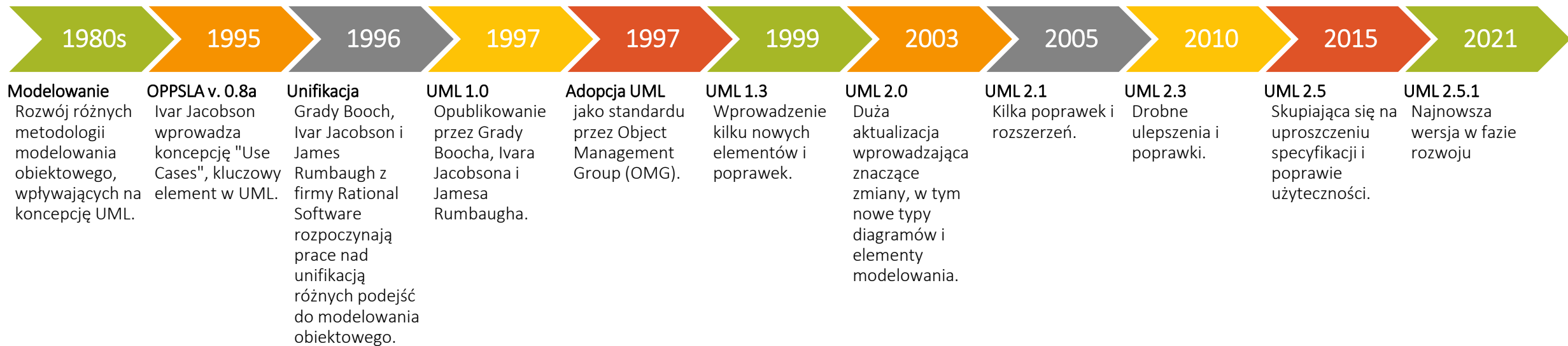
Jest na tyle elastyczny, że może być stosowany w różnorodnych projektach, od małych aplikacji do złożonych systemów korporacyjnych.

Abstrakcja

Pozwala na modelowanie na różnych poziomach abstrakcji, od wysokopoziomowych przypadków użycia do szczegółowych diagramów klas i sekwencji.

Język modelowania UML

Historia



Język modelowania UML

UML obejmuje 13 rodzajów diagramów podzielone na dwie grupy:

Modele strukturalne	Nazwa polska	Nazwa angielska	Opis
	Diagram przypadków użycia	Use case diagram	Przedstawia interakcje pomiędzy użytkownikami systemu oprogramowania a systemem.
	Diagram klas	Class diagram	Przedstawia strukturę systemu oprogramowania w kategoriach klas, atrybutów, asocjacji i operacji.
	Diagram pakietów	Package diagram	Przedstawia hierarchię pakietów w systemie oprogramowania. Pakiety są używane do grupowania powiązanych elementów modelu.
	Diagram obiektów	Object diagram	Przedstawia instancje klas w systemie oprogramowania.
	Diagram struktury złożonej	Composite structure diagram	Przedstawia wewnętrzną strukturę obiektu.
	Diagram komponentów	Component diagram	Przedstawia skład systemu oprogramowania w kategoriach komponentów, połączeń i zależności.
	Diagram wdrożenia	Deployment diagram	Przedstawia, jak system oprogramowania jest instalowany i konfigurowany.

Język modelowania UML

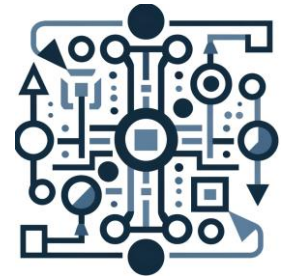
Oraz...

Modele dynamiki	Nazwa polska	Nazwa angielska	Opis
	Diagram interakcji	Interaction diagram	Przedstawia interakcje pomiędzy obiektami w systemie oprogramowania. Diagramy interakcji mogą być używane do modelowania różnych aspektów zachowania systemu, takich jak przepływ kontroli i danych, oraz interakcje pomiędzy obiektami w czasie.
	Diagram sekwencji	Sequence diagram	Przedstawia interakcje pomiędzy obiektami w systemie oprogramowania w czasie. Jest to najpopularniejszy diagram interakcji UML.
	Diagram współpracy	Collaboration diagram	Przedstawia interakcje pomiędzy obiektami w systemie oprogramowania. Diagram współpracy jest podobny do diagramu sekwencji, ale skupia się na obiektach i ich relacjach, a nie na kolejności zdarzeń.
	Diagram aktywności/czynności	Activity diagram	Przedstawia przepływ kontroli i danych w systemie oprogramowania. Jest to pomocny diagram do modelowania złożonych procesów biznesowych lub algorytmów.
	Diagram maszyny stanowej	State machine diagram	Przedstawia stany i przejścia pomiędzy stanami systemu oprogramowania. Jest to pomocny diagram do modelowania zachowania systemu, które jest zależne od stanu systemu.
	Diagram uwarunkowań czasowych	Timing diagram	Przedstawia czas trwania i kolejność zdarzeń w systemie.

Diagram dekompozycji funkcji

FDD (Functional Decomposition Diagram)

to narzędzie używane w analizie systemowej i inżynierii oprogramowania do przedstawienia hierarchicznej struktury systemu lub procesu. Jest to graficzna reprezentacja, która dzieli główną funkcję systemu na zestaw podfunkcji, które są następnie dalej dekomponowane na kolejne poziomy hierarchii.



Cechy

Hierarchiczna Struktura

Jedną z głównych cech FDD jest jego hierarchiczna natura. Diagram zaczyna się od jednej głównej funkcji lub celu, który jest następnie dekomponowany na mniejsze, bardziej szczegółowe podfunkcje.

Modularność

Każda podfunkcja w FDD jest modulem, który można rozważać niezależnie od innych. To ułatwia zarządzanie projektem, ponieważ różne zespoły mogą pracować nad różnymi modułami równocześnie.

Jasność i Zrozumiałość

Jest narzędziem wizualnym, co sprawia, że jest łatwy do zrozumienia nawet dla osób niebędących ekspertami w dziedzinie. Umożliwia to efektywną komunikację między różnymi zainteresowanymi stronami, takimi jak programiści, analitycy biznesowi i menedżerowie.

Fokus na Funkcjonalności

Głównym celem jest analiza i prezentacja funkcjonalności systemu. Nie zajmuje się on aspektami implementacyjnymi, co pozwala na skupienie się na tym, co system ma robić, a nie jak ma to robić.

Elastyczność

Diagramy te są elastyczne i skalowalne. Mogą być stosowane w różnorodnych projektach, od prostych aplikacji do złożonych systemów wielomodułowych. Ponadto, mogą być łatwo aktualizowane lub modyfikowane w miarę ewolucji projektu.

Diagram dekompozycji funkcji

Historia

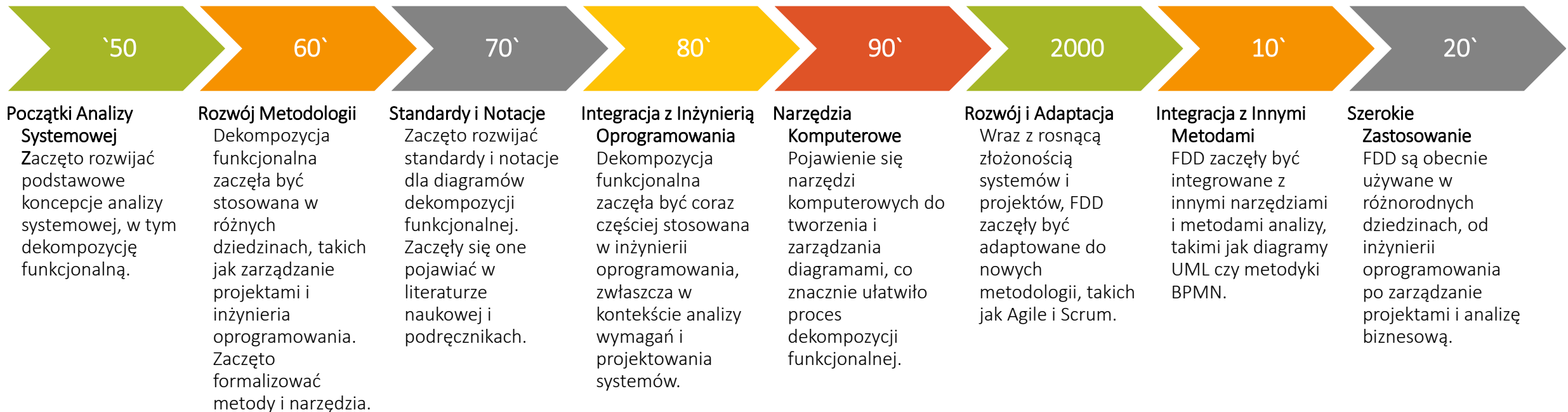
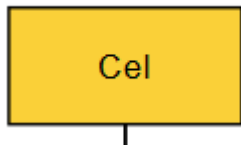


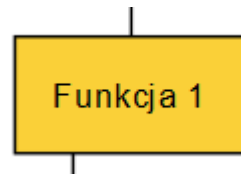
Diagram dekompozycji funkcji

Element diagramu



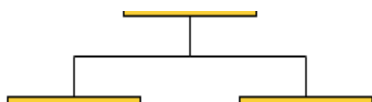
Główna Funkcja lub Cel (Main Function or Objective)

Jest to najwyższy poziom w hierarchii diagramu, reprezentujący główną funkcję lub cel systemu



Podfunkcje (Sub-functions)

Są to funkcje, które są niezbędne do realizacji głównej funkcji.



Linie Zależności (Dependency Lines)

Są to linie łączące różne funkcje i podfunkcje, symbolizujące zależność między nimi.

Diagram dekompozycji funkcji

Przykład – sklep online

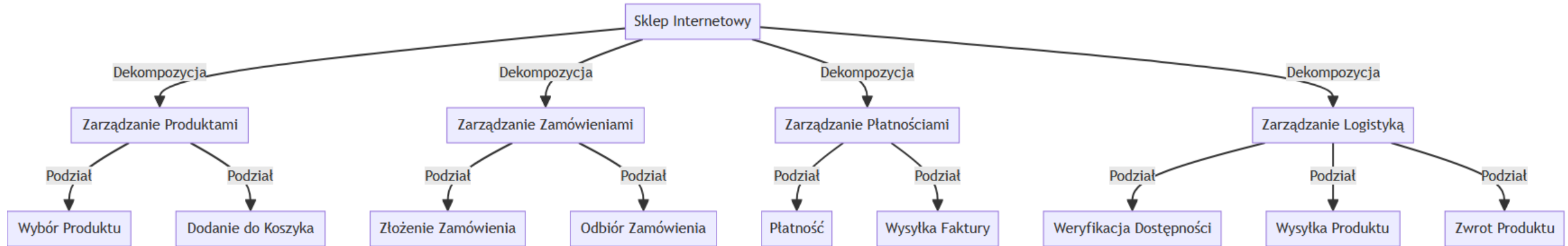
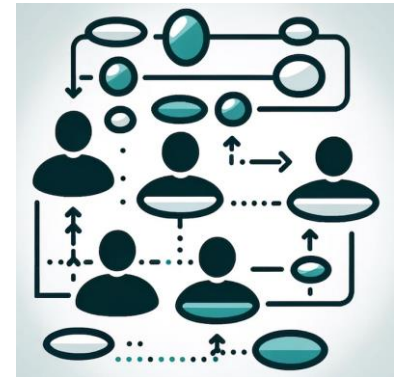


Diagram przypadków użycia

Diagram przypadków użycia (Use case diagram)

Służy do przedstawienia interakcji między systemem a jego otoczeniem (aktorami). Skupia się na tym, "co" system robi, a nie "jak". Przypadki użycia są generalnie wymaganiami funkcjonalnymi dla oprogramowania.



Zastosowanie

Analiza Wymagań

Używany do identyfikacji i organizacji wymagań funkcjonalnych systemu. Pomagają zrozumieć, co system ma robić i jakie są oczekiwania różnych aktorów.

Projektowanie Systemu

Służy do przedstawienia głównych funkcji i interakcji w systemie, co ułatwia późniejsze etapy projektowania i implementacji.

Dokumentacja

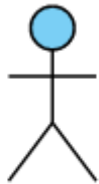
Stanowi graficzną dokumentację funkcjonalności systemu, co ułatwia komunikację i zrozumienie systemu przez różne grupy interesariuszy.

Planowanie Testów

Wykorzystywany jako podstawa do tworzenia scenariuszy testowych, co pomaga w zapewnieniu, że system spełnia określone wymagania funkcjonalne.

Diagram przypadków użycia

Element diagramu

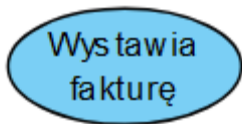


Sprzedawca

Aktor (Actor)

Osoba, system lub obiekt zewnętrzny interagujący się z systemem.

Typy Aktora: Może być osobą, innym systemem, podsystemem.



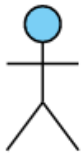
Przypadek Użycia (Use Case)

Reprezentuje funkcję lub zadanie, które system może wykonać.

Abstrakcja: Skupia się na "co" system robi, nie "jak".

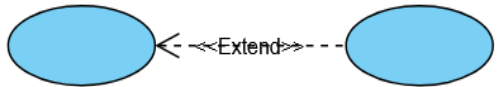
Diagram przypadków użycia

Relacje (Relationships)



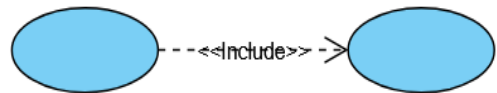
Asocjacja (Association)

między aktorem a przypadkiem użycia oznacza, że aktor może uczestniczyć w przypadku użycia



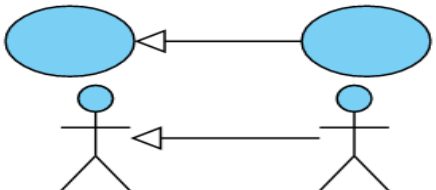
Rozszerzenie (Extend)

Relacja rozszerzenia między przypadkami użycia oznacza, że przypadek użycia rozszerza funkcjonalność innego przypadku użycia.



Uwzględnienie (Include)

między przypadkami użycia oznacza, że przypadek użycia agreguje funkcjonalność innego przypadku użycia.



Generalizacja (Generalization):

wskazuje na dziedziczenie cech między aktorami lub przypadkami użycia.

Diagram przypadków użycia

Przykład – sklep online

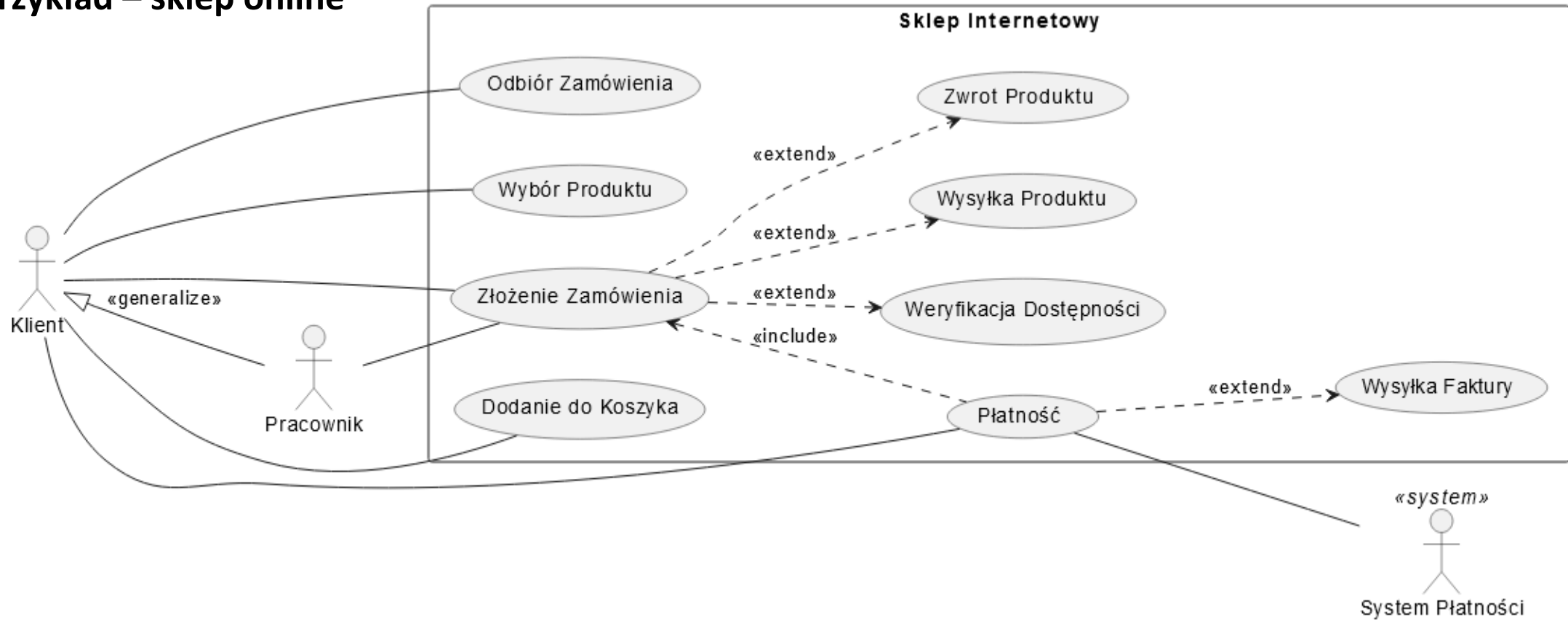


Diagram aktywności/czynności

Diagram aktywności/czynności (Activity diagram)

Służy do modelowania dynamicznych aspektów systemu. Jego zasadniczą funkcją jest przedstawienie sekwencji kroków, które są wykonywane przez modelowany fragment systemu.



Zastosowanie

Procesy Biznesowe

Służą do przedstawienia złożonych procesów biznesowych, w tym przepływu informacji i decyzji, co pozwala na ich optymalizację i automatyzację.

Scenariusze Użytkowania

Używane do modelowania interakcji i przepływu sterowania w różnych scenariuszach użytkowania, co pomaga w zrozumieniu, jak użytkownicy lub systemy współdziałają z daną aplikacją.

Algorytmy i Logika

Pomagają w wizualizacji algorytmów i logiki sterującej w systemach informatycznych, co jest szczególnie użyteczne w fazie projektowania i analizy.

Workflow i Orkiestracja

Używane do modelowania przepływu zadań i aktywności w ramach różnych systemów i aplikacji, co pozwala na lepsze zrozumienie i zarządzanie złożonymi operacjami.

Diagram aktywności/czynności

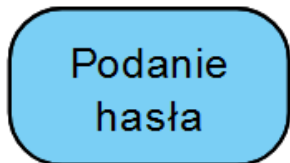
Element diagramu



Węzeł początkowy (Start Node)
Oznacza początek przepływu w diagramie.



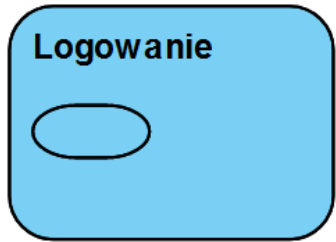
Węzeł końcowy (End Node)
Oznacza zakończenie przepływu w diagramie



Akcja (Actions)
Oznaczają konkretne zadania lub operacje wykonywane w systemie.

Diagram aktywności/czynności

Element diagramu



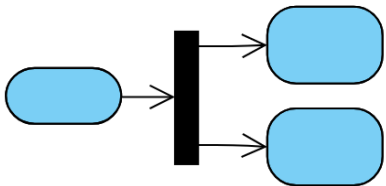
Aktywność (Activity)

złożony procesem, który można dalej rozłożyć na mniejsze akcje



Decyzje (Decision Nodes)

Używane do reprezentowania punktów, w których podejmowane są decyzje.

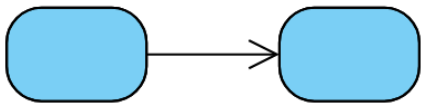


Równoczesność (Forks and Joins)

Używane do rozdzielenia (Fork) i łączenia (Join) równoczesnych ścieżek przepływu..

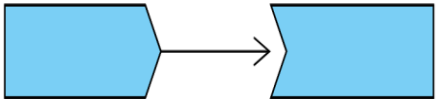
Diagram aktywności/czynności

Element diagramu



Łączniki (Control Flows)

Wskazują kierunek przepływu sterowania między różnymi elementami diagramu



Sygnały (Signals)

Przesyłanie i odbieranie sygnałów asynchronicznych



Partycje aktywności (Activity Partitions)

Grupują zestawy powiązanych akcji i aktywności

Diagram aktywności/czynności

Przykład - logowanie

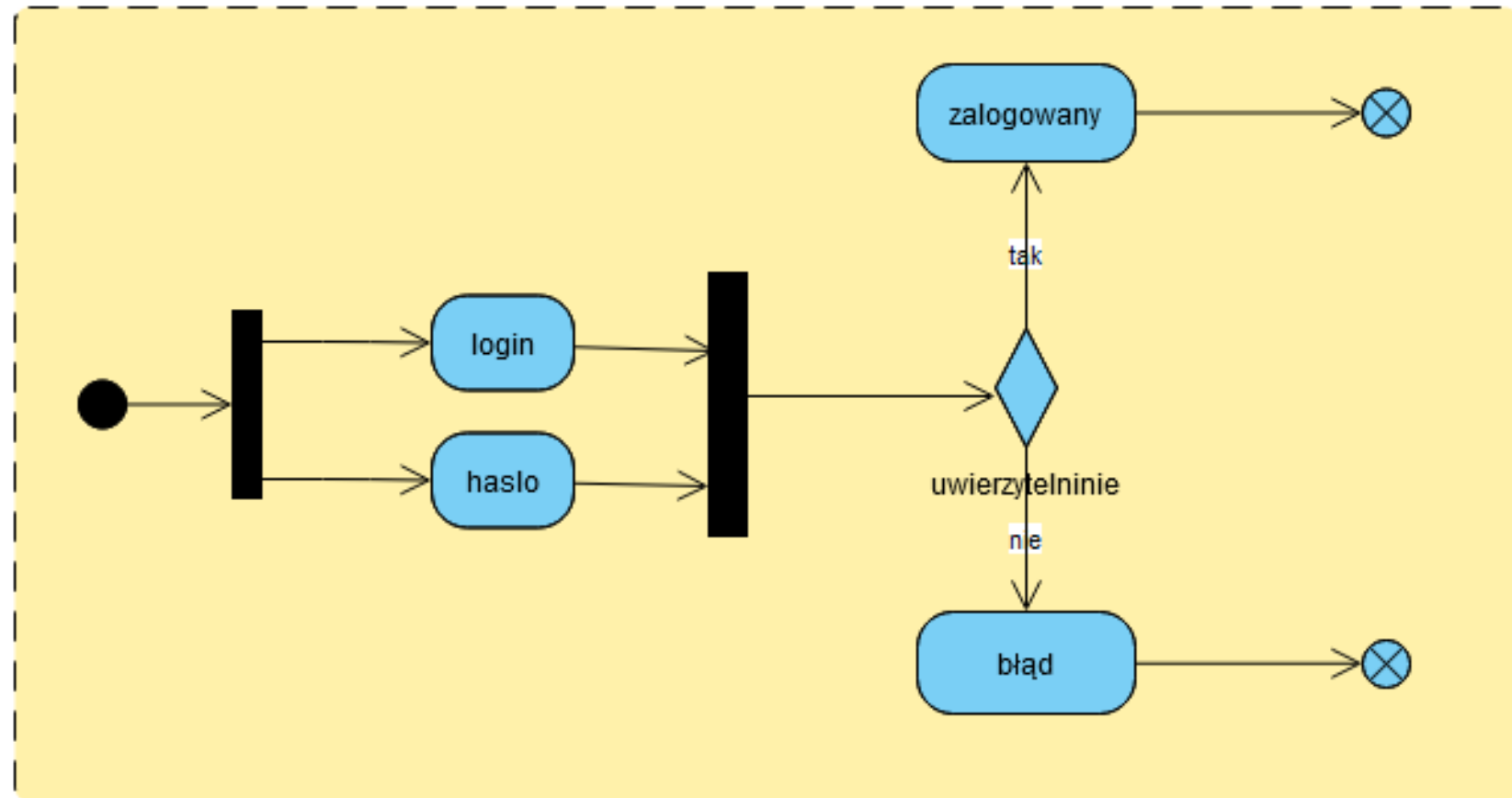


Diagram aktywności/czynności

Przykład – sklep online

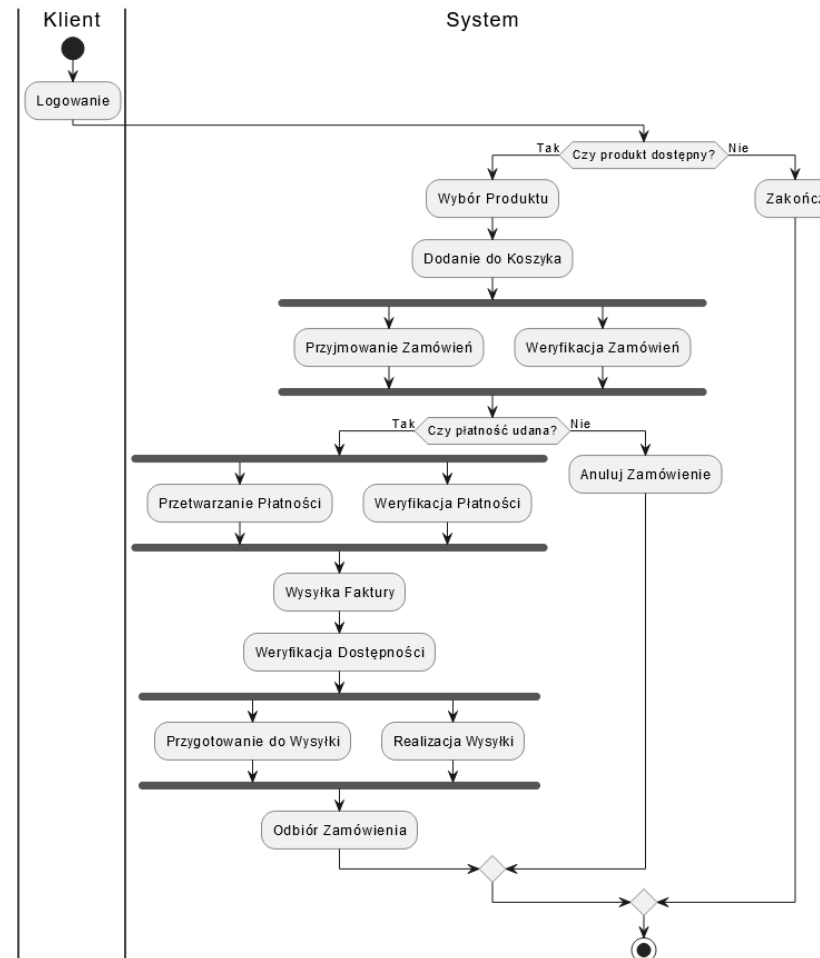
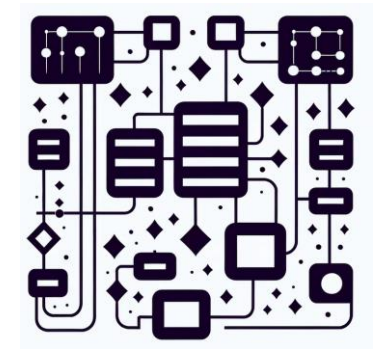


Diagram związków encji ERD

Diagram związków encji (Entity-Relationship Diagram, ERD)

to graficzna reprezentacja struktury bazy danych. Jest to narzędzie używane do modelowania i analizy systemów baz danych. ERD pomaga w zrozumieniu relacji pomiędzy różnymi encjami (takimi jak tabele w bazach danych) oraz atrybutami tych encji (takimi jak pola w tabelach).



Cechy

Reprezentacja modelu danych

Jest graficzną reprezentacją modelu danych, czyli sposobu, w jaki dane są przechowywane i powiązane ze sobą.

Modelowanie struktury bazy danych

Jest używany do modelowania struktury bazy danych, czyli sposobu, w jaki dane są przechowywane i powiązane ze sobą.

Komunikacja z interesariuszami

Jest wizualnym sposobem przedstawienia złożonych informacji o danych. Może być używany do komunikacji z interesariuszami, którzy nie są technicznymi ekspertami.

Elastyczność

Jest elastycznym narzędziem, które można dostosować do różnych potrzeb. Może być używany do modelowania różnych typów danych, w tym danych relacyjnych, obiektowych i rozproszonych.

Diagram związków encji ERD

Historia

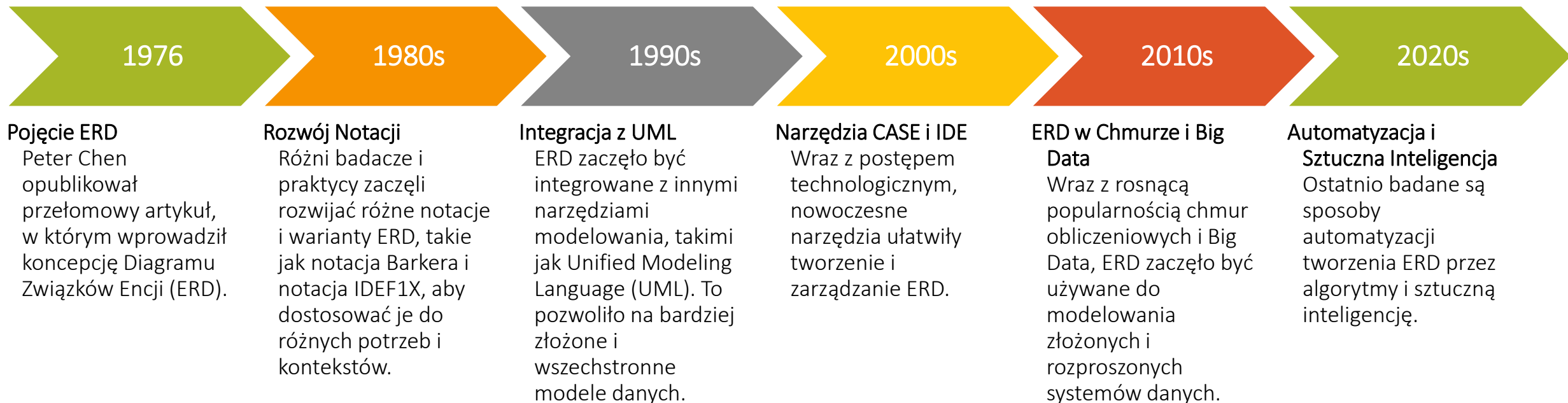


Diagram związków encji ERD

ERD obejmuje kilka rodzajów diagramów

Nazwa polska	Nazwa angielska	Opis
Koncepcyjny ERD	Conceptual ERD	Najbardziej ogólny rodzaj ERD, który służy do przedstawienia wysokopoziomowego obrazu struktury danych. Nie zawiera on szczegółowych informacji o atrybutach czy relacjach, ale skupia się na głównych encjach i ich związkach.
Logiczny ERD	Logical ERD	Bardziej szczegółowy niż model koncepcyjny i zawiera informacje o atrybutach encji oraz o relacjach między nimi. Jest to model niezależny od technologii, co oznacza, że nie uwzględnia on szczegółów implementacyjnych.
Fizyczny ERD	Physical ERD	Najbardziej szczegółowy rodzaj ERD, który uwzględnia konkretne elementy implementacyjne, takie jak tabele, klucze i indeksy w bazie danych. Jest on zwykle używany przez programistów podczas faktycznej implementacji bazy danych.
Notacja Chen	Chen Notation	Klasyczna notacja ERD, stworzona przez Petera Chena. Jest ona szeroko używana i skupia się na reprezentacji encji, atrybutów i relacji w sposób jasny i zrozumiały.
Notacja Crow's Foot	Crow's Foot Notation	Popularna notacja, która jest często używana w narzędziach do modelowania baz danych. Jest ona bardziej zorientowana na praktyczne aspekty i łatwość czytania.
IDEF1X	IDEF1X	Formalna notacja stworzona przez U.S. Air Force, która jest bardziej formalna i zawiera dodatkowe elementy, takie jak ograniczenia integralności.

Diagram związków encji ERD

Notacji Crow's Foot

to popularny sposób graficznego przedstawiania Diagramów Związków Encji (ERD), w którym encje są reprezentowane jako prostokąty, a ich związki za pomocą linii z symbolami na końcach, przypominającymi "łapki kruka". Te symbole określają kardynalność związku, czyli ilość rekordów jednej encji powiązanych z rekordami drugiej encji.

Zastosowanie

Projektowanie Baz Danych

Jest często używana przez architektów i projektantów baz danych do modelowania struktury danych przed jej implementacją.

Analiza Biznesowa

Analitycy biznesowi korzystają z tej notacji do zrozumienia i dokumentowania zasobów informacyjnych w organizacji.

Inżynieria Oprogramowania

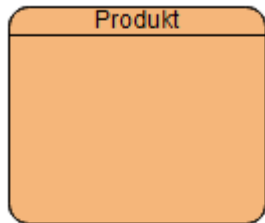
W fazie projektowania systemów informacyjnych, programiści i inżynierowie korzystają do przedstawienia, jak różne komponenty systemu będą ze sobą interagować na poziomie danych.

Edukacja i Szkolenia

Jest powszechnie używana w materiałach dydaktycznych i kursach z zakresu baz danych, dzięki swojej czytelności i łatwości zrozumienia.

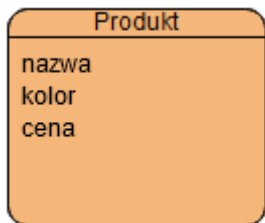
Diagram związków encji ERD

Element notacji



Encja (Entity)

Podstawowy obiekt w bazie danych. Na przykład, "Klient" czy "Produkt".

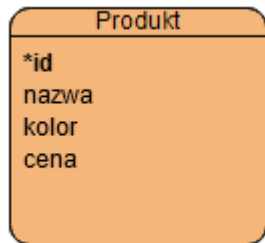


Atrybut (Attribute)

Właściwości encji. Na przykład, „Nazwa” czy „kolor”.

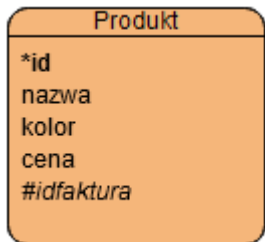
Diagram związków encji ERD

Element notacji



Klucz Główny (Primary Key)

Atrybut lub zestaw atrybutów, które jednoznacznie identyfikują rekord w encji (*).



Klucz Obcy (Foreign Key)

Atrybut służący do łączenia dwóch różnych encji (#)

Diagram związków encji ERD

Element notacji

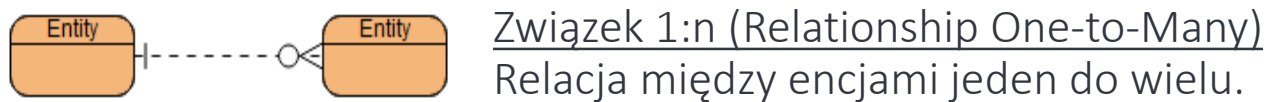
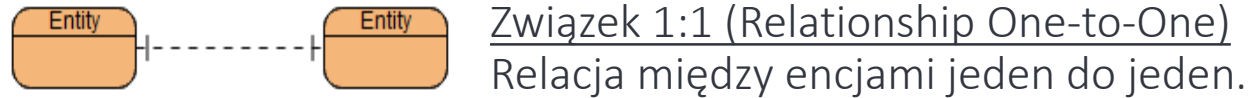
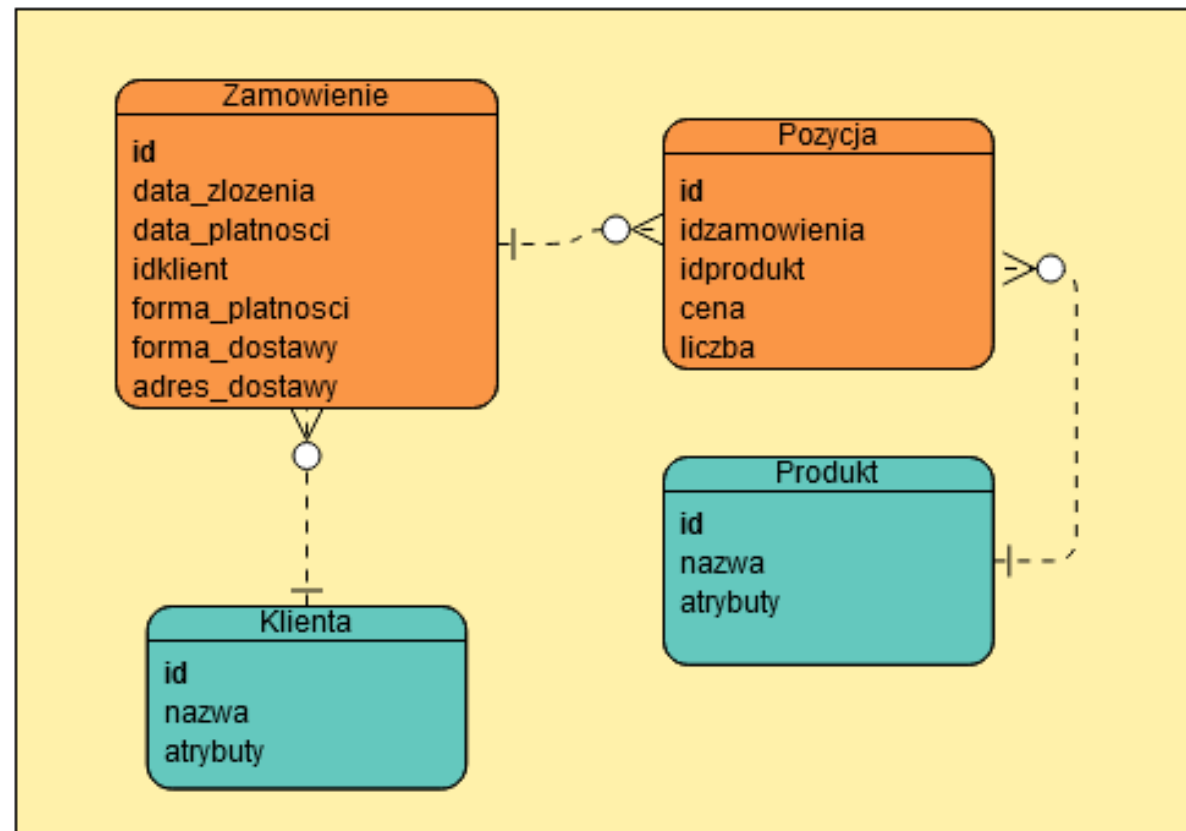


Diagram związków encji ERD

Przykład – sklep online



Modelowanie procesów biznesowych BPMN

Modelowanie Procesów Biznesowych (Business Process Model and Notation, BPMN)

to ustandaryzowany język graficznej notacji służący do modelowania i wizualizacji procesów biznesowych. Jest on zaprojektowany w celu zapewnienia spójnego i zrozumiałego schematu, który umożliwia zarówno analizę, jak i implementację procesów w organizacjach.



Cechy

Uniwersalność

Jest standardem akceptowanym na szeroką skalę, co umożliwia jego zastosowanie w różnych branżach i kontekstach organizacyjnych.

Wysoka Czytelność

Został zaprojektowany tak, aby była zrozumiała nie tylko dla specjalistów IT, ale również dla osób z działów biznesowych. Umożliwia to efektywną komunikację i współpracę.

Kompleksowość

Pozwala na modelowanie zarówno prostych, jak i złożonych procesów, oferując różnorodne elementy do opisu zdarzeń, aktywności, decyzji i interakcji.

Możliwość Symulacji

Umożliwia nie tylko statyczne modelowanie procesów, ale również ich dynamiczną symulację. To pozwala na analizę i optymalizację procesów przed ich faktyczną implementacją.

Modelowanie procesów biznesowych BPMN

Historia



Modelowanie procesów biznesowych BPMN

BPMN obejmuje kilka rodzajów diagramów

Nazwa polska	Nazwa angielska	Opis
Diagram Procesów	Process Diagram	Ilustruje sekwencję aktywności i zdarzeń w jednym procesie biznesowym.
Diagram Kolaboracji	Collaboration Diagram	Pokazuje interakcje pomiędzy różnymi uczestnikami (np. działami, systemami) w procesach.
Diagram Konwersacji	Conversation Diagram	Skupia się na komunikacji pomiędzy uczestnikami, bez uwzględnienia szczegółów procesu.
Diagram Choreografii	Choreography Diagram	Opisuje, jak różne jednostki współpracują, aby zrealizować określony cel biznesowy.
Diagram Struktury Organizacyjnej	Organizational structure diagram	Przedstawia strukturę organizacyjną firmy (hierarchię i podległości).

Modelowanie procesów biznesowych BPMN

Diagram Procesów (Process Diagram)

to graficzna reprezentacja sekwencji aktywności, zdarzeń i decyzji, które razem tworzą jeden spójny proces biznesowy. Jest to najbardziej podstawowy i powszechnie używany rodzaj diagramu w BPMN, służący do ilustrowania i analizy przepływu pracy w organizacji. Diagram ten umożliwia zrozumienie, jakie kroki są niezbędne do osiągnięcia określonego celu biznesowego, i w jaki sposób te kroki są ze sobą powiązane.

Zastosowanie

Analiza i Optymalizacja Procesów

Są często używane do identyfikacji słabych punktów, nadmiernych kosztów czy nieefektywności w istniejących procesach, co pozwala na ich optymalizację.

Automatyzacja Procesów

Służą jako podstawa do automatyzacji określonych aktywności w ramach systemów zarządzania procesami biznesowymi.

Dokumentacja i Audyt

Są używane do tworzenia oficjalnej dokumentacji procesów w organizacji, co jest szczególnie ważne w kontekście zgodności z regulacjami i standardami jakości.

Komunikacja i Szkolenie

Są efektywnym narzędziem w komunikacji pomiędzy różnymi działami w organizacji oraz w procesie szkolenia nowych pracowników.

Modelowanie procesów biznesowych BPMN

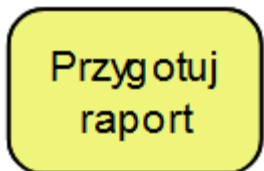
Element notacji



Zdarzenie Startowe (Start Event)
Oznacza początek procesu lub ścieżki w procesie.



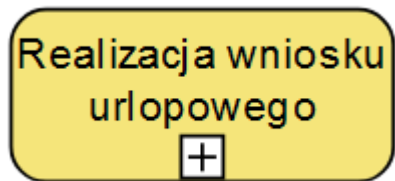
Zdarzenie Końcowe (End Event)
Sygnalizuje zakończenie procesu lub ścieżki.



Zadanie (Task)
Reprezentuje pojedynczą jednostkę pracy w procesie, taką jak "Zatwierdź fakturę" czy "Przygotuj raport".

Modelowanie procesów biznesowych BPMN

Element notacji



Podproces (Sub-Process)

Złożona aktywność, która może zawierać inne aktywności, zdarzenia i bramki. Umożliwia modelowanie procesów na różnych poziomach szczegółowości.



ALBO

Bramki Wyłączająca (Exclusive Gateway)

Umożliwia wybór jednej z kilku dostępnych ścieżek w procesie na podstawie określonych warunków.



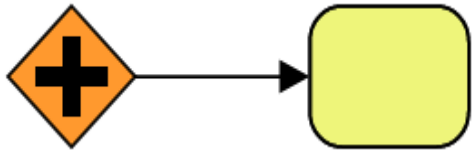
RÓWNOL
EGLE

Bramka Równoległa (Parallel Gateway)

Umożliwia równoczesne wykonywanie kilku ścieżek w procesie, bez żadnych warunków decyzyjnych.

Modelowanie procesów biznesowych BPMN

Element notacji



Sekwencyjny Przepływ Sterujący (Sequence Flow)

Strzałka pokazująca kierunek przepływu sterowania między elementami w procesie.



Pula (Pool)

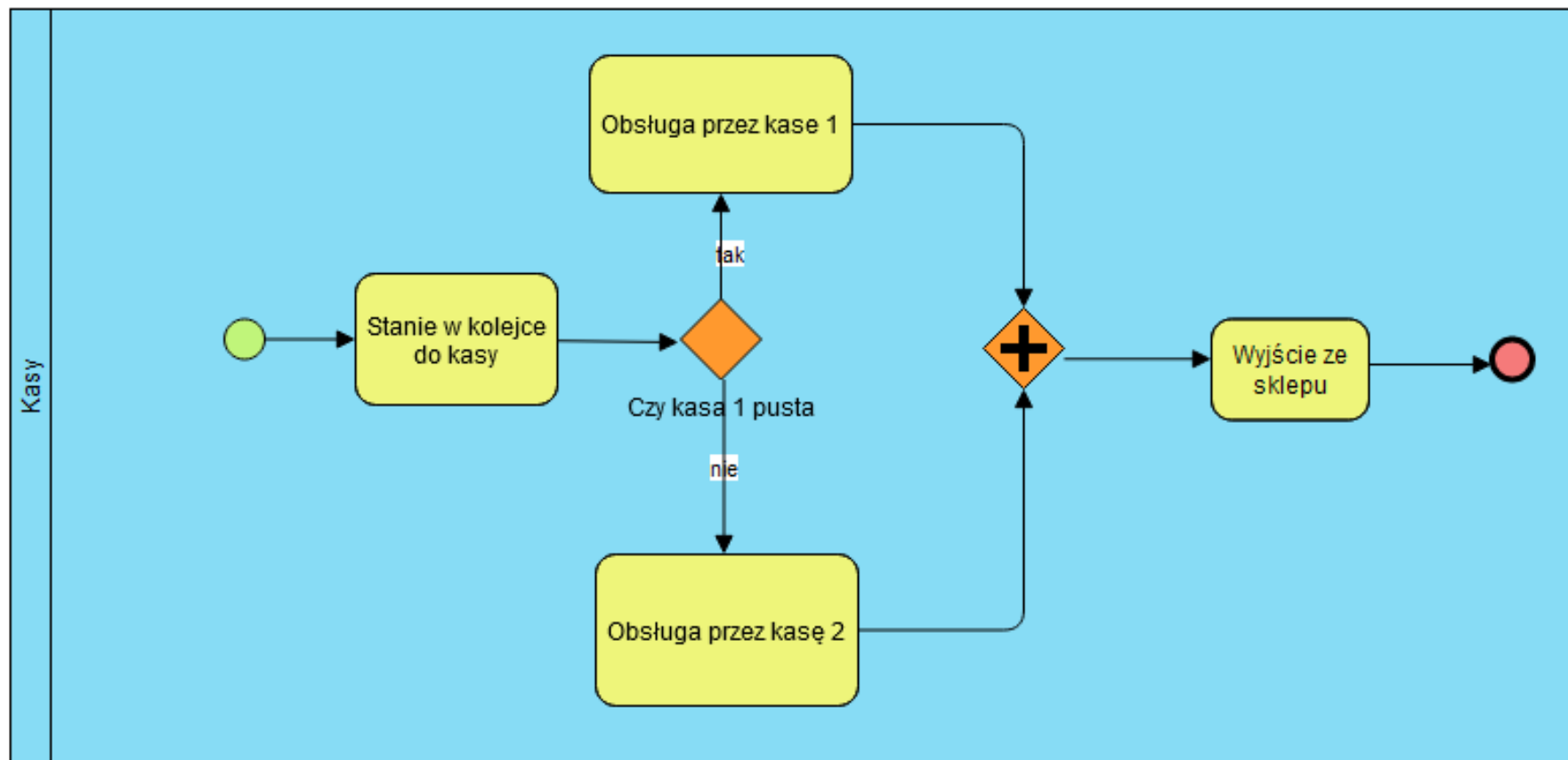
Reprezentuje uczestnika w interakcjach biznesowych i jest kontenerem dla procesów i torów.

Poster http://www.bpmb.de/images/BPMN2_0_Poster_PL.pdf



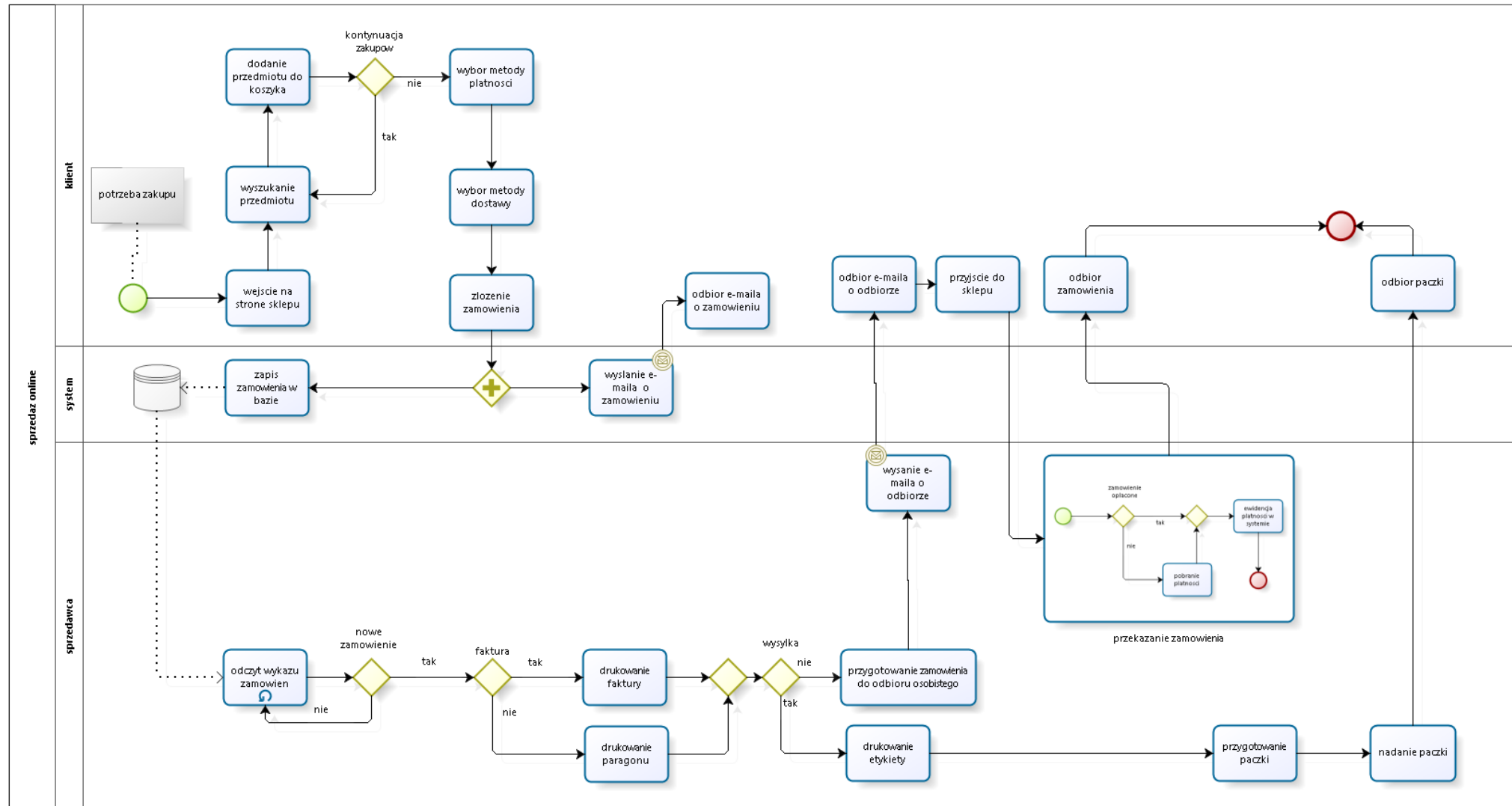
Modelowanie procesów biznesowych BPMN

Przykład - kasa



Modelowanie procesów biznesowych BPMN

Przykład - sklep online



Narzędzia do modelowania



Visual Paradigm

to uniwersalne narzędzie do modelowania, które obsługuje szeroki zakres diagramów UML, BPMN, oraz możliwość tworzenia ERD i innych schematów.

<https://online.visual-paradigm.com/pl/>



Lucidchart

to aplikacja bazująca na chmurze, która umożliwia tworzenie diagramów UML, ERD, BPMN i wielu innych, z łatwością współpracy w zespole.

<https://www.lucidchart.com/>



draw.io (teraz diagrams.net)

to darmowe narzędzie online, które umożliwia tworzenie prostych diagramów UML, ERD oraz BPMN.

<https://diagrams.net/>

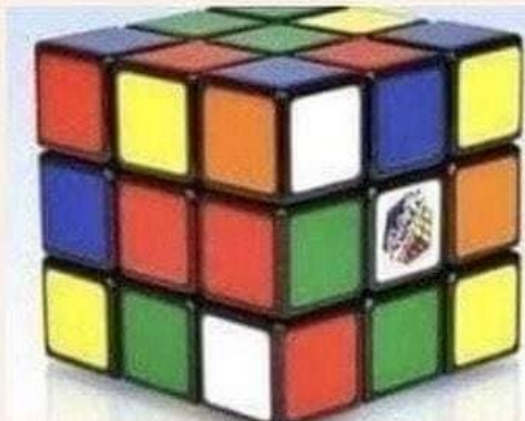


Enterprise Architect

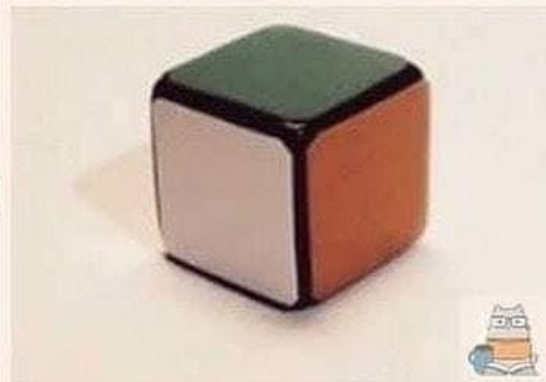
to kompleksowe narzędzie do modelowania, które wspiera UML i wiele innych standardów, w tym BPMN oraz ERD.

<https://sparxsystems.com/products/ea/>

Co omówił
wykładowca



Co
zapamiętałem



Co było
na zaliczeniu

