



**Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego**

# Rozproszone systemy internetowe 2

## CORBA Services

**„Podniesienie potencjału uczelni wyższych jako czynnik rozwoju gospodarki opartej na wiedzy”**

Nr projektu: UDA-POKL.04.01.01-00-143/09-00

# CORBA Services

- Collection
- Concurrency
- Event
- Externalization
- Licensing
- Lifecycle
- Naming
- Notification
- Persistency
- Property
- Query
- Relationship
- Security
- Time
- Trading Object
- Transaction

# Collection Service 1

*Grupowanie obiektów i operacje na tych grupach, jak również na poszczególnych elementach kolekcji*

Trzy kategorie interfejsów:

- **Kolekcje i fabryki kolekcji** – interfejsy podstawowe
- **Iteratory** – przeglądanie elementów kolekcji
- **Funkcje** – definicje operacji wykonywanych na elementach kolekcji

# Collection Service 2

		Brak porządku		Uporządkowane		
				Sortowane		Sekwen.
		Unikalne	Wielokrotne	Unikalne	Wielokrot.	Wielokrot.
Klucz (spec. równow. klucza)	Równow. ażność	<b>Map</b>	<b>Relation</b>	<b>Sorted Map</b>	<b>Sorted Relation</b>	
	Brak równow.	<b>KeySet</b>	<b>KeyBag</b>	<b>Key Sorted Set</b>	<b>Key Sorted Bag</b>	
Brak klucza	Równow. ażność	<b>Set</b>	<b>Bag</b>	<b>Sorted Set</b>	<b>Sorted Bag</b>	<b>Equality Sequence</b>
	Brak równow.		<b>Heap</b>			<b>Sequence</b>

# Concurrency Service

*Kontroluje współbieżny dostęp do obiektu,  
tak aby jednoczesne obliczenia nie naruszyły  
jego spójności*

Dwa tryby uzyskiwania blokady:

- **Na poziomie transakcji** – usługa transakcji jest odpowiedzialna za zwalnianie blokad
- **Na poziomie wątku wywołującego** – zwalnianie blokad leży w gestii klienta

# Concurrency Service 2

- Wiele blokad na jednym zasobie
- Dwufazowe blokowanie transakcyjne
- Typy blokad: *Read, Write, Upgrade, Intentional Read, Intentional Write*

Aktualna blokada	Żądana blokada				
	IR	R	U	IW	W
IR					x
R				x	x
U			x	x	x
IW		x	x		x
W	x	x	x	x	x

# Event Service

*Asynchroniczne interakcje między obiektami*

- **Dostawcy (*supplier*)** – produkują dane
- **Konsumenci (*consumer*)** – przetwarzają
- **Kanały zdarzeń (*event channel*)** –  
wspomagają komunikację pomiędzy wieloma dostawcami oraz wieloma konsumentami
- **Model PUSH** – dostawca inicjalizuje proces przekazywania danych do konsumentów
- **Model PULL** – konsument żąda danych od dostawcy

# Externalization Service

*Zapis i odczyt stanu obiektu do strumienia*

- Eksportowanie obiektów do różnych mediów: dyski twarde, strumienie sieciowe
- Odtwarzanie obiektów (być może w innym ORB)
- Zdefiniowany standardowy format ale również możliwość wykorzystania innych formatów
- Zależność od Lifecycle Service



# Licensing Service

*Kontrola dostępu do obiektów i usług  
w modelu licencyjnym*

- Konceptcje systemów licencyjnych
  - Rodzaje licencji
  - Zarządzanie licencjami
- Schemat działania usługi licencjonowanej:
  - Klient – Usługa – Menedżer usług licencjonowanych – System zarządzania licencjami

# LifeCycle Service

*Usługa tworzenia, usuwania, kopiowania  
oraz przesuwania obiektów*

- Fabryki różne: kreowanie obiektów
- LifecycleObject
  - Remove
  - Copy
  - Move
- FactoryFinder
  - Reprezentują zdalną lokację przy kopiowaniu i przesuwaniu

# LifeCycle: problem tworzenia

- Czy klient może kontrolować położenie nowego obiektu ?
- Czy taka lokalizacja może zostać określona za pomocą dobrze zdefiniowanej polityki ?
- Z jakim obiektem należy się skontaktować aby utworzyć nowy obiekt ?
- Jak znaleźć powyższy obiekt ?
- Jaką kontrolę ma klient nad implementacją tworzonego obiektu ?
- Czy klient może wpływać na początkowe wartości w nowym obiekcie ?
- Czy klient może stworzyć obiekt posługując się notacją zależną od implementacji ?

# LifeCycle: pozostałe problemy

- Kopiowanie i przesuwanie
  - Podobnie jak tworzenie plus
    - Co się dzieje z kodem implementacji wynikowego obiektu ?
- Operowanie na grafie obiektów
  - Jak określić granice takiego grafu ?
  - Co zrobić z resztą obiektów w grafie ?
  - Czy należy zachować cykle w grafie ?

# Naming Service

## *System nazewniczy dla obiektów*

- *Name binding* – powiązanie obiektu z nazwą zawsze względem kontekstu (*name context*)
- *Name context* – obiekt, który zawiera zbiór unikalnych powiązań
- *Compound name* – ścieżka w grafie nazw, która służy do znalezienia powiązania
- Nazwa w NS składa się z komponentów (elementów ścieżki)
- Przeglądanie zbioru umożliwia interfejs *BindingIterator*

# Notification Service

## *Rozszerzenie komunikacji z Event Service*

- Definicje filtrów komunikatów pozwalające na selekcję interesujących nas zdarzeń
- Wykrywanie typów zdarzeń subskrybowanych/definiowanych na danym kanale – optymalizacja generacji zdarzeń i możliwość subskrybowania nowych
- QoS dla kanału, zdarzenia lub proxy
- Definicja standardowej struktury dla zdarzenia

# Persistent State Service

## *Obiekty trwałe*

- *Datastore* – system zarządzania danymi (baza danych, zbiór plików)
- *Storage object* – obiekt trwały
- *Storage home* – zbiór obiektów trwałych tego samego typu
- Dostęp do obiektów trwałych jest realizowany przez ich *instancje* (*inkarnacje* jeśli są podłączone)
- Instancje *storage home* są dostępne przez *katalogi*
- Dostęp do *datastore* wymaga utworzenia powiązania z procesem zwanego sesją

# Property Service

*System dynamicznych wartości określonych  
typem i nazwą*

- W odróżnieniu od atrybutu, własność (*property*) nie jest określana statycznie w IDL tylko dodawana dynamicznie
- Klient może nadawać wartości własnościom ale może także dodawać i usuwać własności
- Obiekt może implementować jeden z dwóch interfejsów:
  - PropertySet – definiowanie, usuwanie, wyliczanie i sprawdzanie istnienia własności
  - PropertySetDef – dodatkowo tryby dostępu do własności, ograniczenia



# Query Service

*System zapytań dla obiektów rozproszonych*

- Efektem ewaluacji zapytania jest kolekcja obiektów
- Obiekty z kolekcji wynikowej mogą być również modyfikowane i usuwane
- *QueryEvaluator* – interfejs interpretujący zapytanie
- *QueryableCollection* – jednocześnie interpretacja zapytań i implementacja kolekcji (podkolekcje)
- *QueryManager* i *Query* – zapytania z parametrami

# Relationship Service

## *Specyfikacja zależności pomiędzy obiektami*

- Zależności definiowane przez:
  - Typ: typ zależności ale również typy podmiotów
  - Rola: określenie funkcji pełnionej w danej zależności
  - Stopień: ilość spełnianych ról wymaganych w zależności
  - Liczność: maksymalna ilość zależności, które wykorzystują tę rolę (one-to-many...)
  - Semantyka: zależności generują dodatkowe atrybuty i operacje

# Security Service

## *Bezpieczeństwo aplikacji rozproszonych*

- Weryfikacja użytkowników i obiektów (principal)
- Autoryzacja (określanie zasad dostępu do obiektu)
- Audyt wywołań
- Bezpieczeństwo komunikacji
- Zabezpieczenia przed wyparciem
- Administracja systemem bezpieczeństwa

# Time Service

## *Usługi związane z czasem*

- Możliwość podawania aktualnego czasu wraz z estymacją błędu (komunikacja z zewnętrznym źródłem)
- Generowanie zdarzeń określanych czasowo (pilnowanie ustawienia takich zdarzeń)
- Pomiar czasu pomiędzy zdarzeniami
- *TimeService* – aktualny czas, interwały
- *TimerEventService* - zdarzenia

# Trading Object Service

## *Publikacja i wyszukiwanie usług*

- Obiekty opisują usługi, które oferują
- Klienci opisują wymagania jakie określają dla żądanej usługi
- Trading Object Service przeprowadza dopasowanie i zwraca kolekcję obiektów spełniających wymagania
- Lookup: publikacja żądań importerów
- Register: publikacja opisu usługi
- OfferIterator: przeglądanie ofert
- Admin: kontrola procesu dopasowywania

# Transaction Service

## *System transakcji*

- Transakcja – działanie, które posiada następujące własności (ACID):
  - Atomowa (atomic) – jeśli została przerwana przez błąd, wszystkie jej efekty zostają anulowane
  - Spójna (consistent) – wszystkie wyniki operacji wewnątrz transakcji są spójne
  - Izolowana (isolated) – transakcje nie widzą się nawzajem zanim nie zostaną zatwierdzone
  - Trwała (durable) – efekty zatwierdzonej transakcji są trwałe

# Transaction Service c.d.

- Usługa transakcjonowania realizuje:
  - Kontrolę zakresu i czasu trwania transakcji
  - Możliwość wykorzystania wielu obiektów w jednej transakcji
  - Możliwość powiązania zmian stanu obiektów z transakcją
  - Koordynację zatwierdzania transakcji
- Modele transakcji:
  - Płaska (flat)
  - Zagnieżdżona (nested)