

Programowanie składowe w modelu COM wprowadzenie

Jarosław Francik

COM - Component Object Model

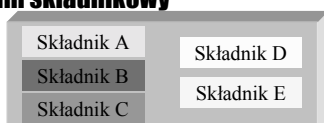
Programowanie składowe

Programowanie składowe

⌚ Program - monolit

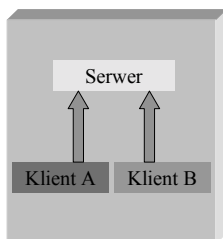


⌚ Program składowy

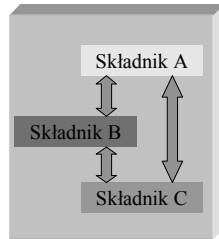


Architektura składowa a architektura klient serwer

Architektura klient-serwer



Program składowy



Programowanie obiektowe a programowanie składowe

Interfejs:

Grupa właściwości i metod implementowanych przez klasę.

Kontrakt pomiędzy klientem a składnikiem

C++: Dziedziczenie

COM: Interfejsy i ich implementacja

Java: Jedno i drugie

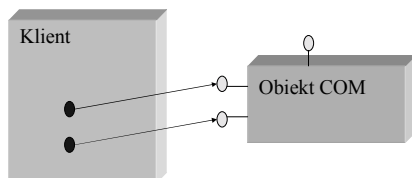
Programowanie składowe podejście firmy Microsoft

COM - Common Object Model

- OLE - Object Linking and Embedding
- Automatyzacja - Automation (dawniej: Automatyzacja OLE)
- ActiveX
- DirectX
- DAO - Data Access Objects
- DCOM - Distributed COM

COM+ - technologia dla XXI wieku

Programowanie składnikowe podejście firmy Microsoft



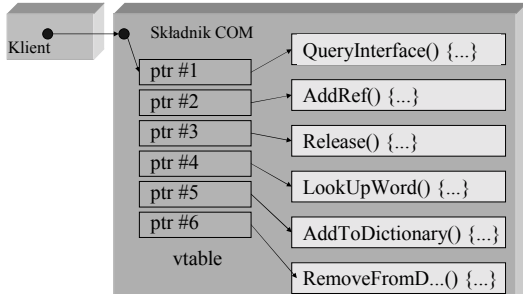
Programowanie składnikowe podejście firmy Microsoft

- ⌚ **Obiekt: kombinacja danych i metod**
- ⌚ **Klasy i interfejsy**
COM pozwala stosować wiele interfejsów dla jednej klasy
- ⌚ **Hermetyzacja**
- ⌚ **Polimorfizm**
- ⌚ **Dziedziczenie**
COM zapewnia tylko dziedziczenie interfejsów

Programowanie składnikowe podejście firmy Microsoft

- ⌚ **Obiektość**
- ⌚ **Spójność**
- ⌚ **Niezależność od języka**
- ⌚ **Kontrola wersji**

Interfejsy



Interfejsy są niezmiennie

**Dodanie nowej funkcjonalności
wymaga zdefiniowania nowego
interfejsu**

Typy składników COM

- ⌚ Składniki wewnątrzprocesowe - DLL
- ⌚ Składniki zewnątrzprocesowe, lokalne - EXE
- ⌚ Składniki zewnątrzprocesowe, zdalne
- ⌚ Formanty OLE (wewnątrzprocesowe) - OCX

Składniki wewnątrzprocesowe

- ℳ Działają w obrębie procesu klienta
- ℳ Korzystają z przestrzeni adresowej klienta
- ℳ Mniejsza stabilność
- ℳ Wysoka wydajność

Składniki zewnątrzprocesowe

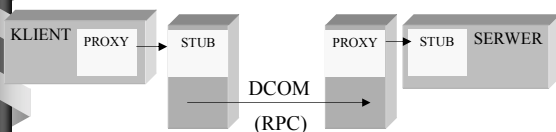
- ℳ Przetwarzanie żądań klienta w osobnym procesie i wątku
- ℳ Osobna przestrzeń adresowa
- ℳ Wysoka stabilność
- ℳ Niższa wydajność (przekazywanie danych poprzez granice procesów)
- ℳ Możliwość uruchomienia jako niezależnej aplikacji
- ℳ Przydatne przy automatyzacji i w DCOM

Przetwarzanie rozproszone w relacji klient - serwer

- ℳ Korzystanie z lokalnego składnika



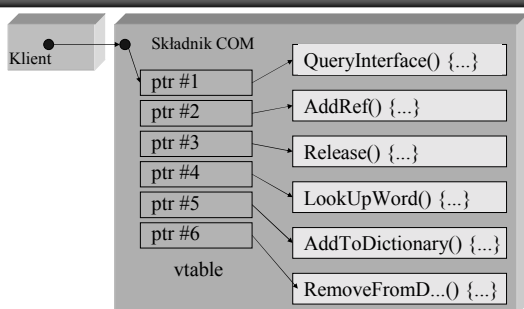
- ℳ Korzystanie ze zdalnego składnika



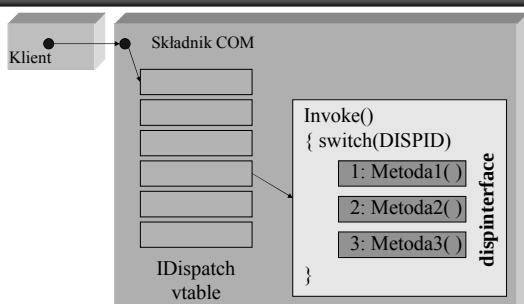
Automatyzacja

- Programowalność
- Udostępnianie interfejsów
- Użytkowanie za pomocą Visual Basic
- „Uproszczone interfejsy”:
 - IDispatch
 - dispatch interfaces (dispinterfaces)

Interfejsy (przypomnienie)



dispinterface IDispatch::Invoke

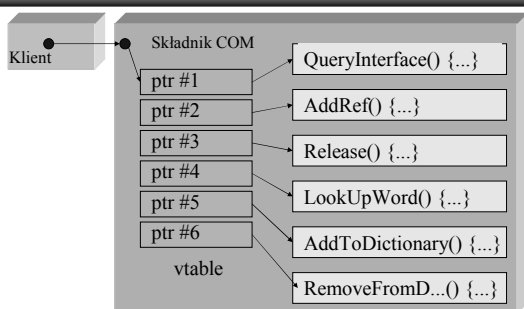


vtable vs. IDispatch

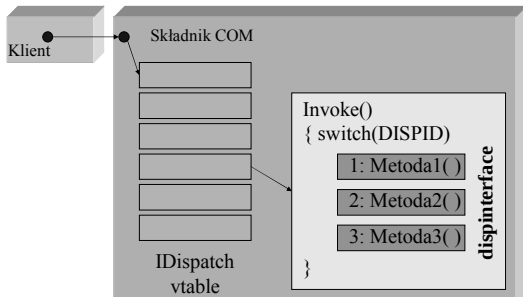
- Visual Basic – IDispatch
- C++ – vtable
- Automatyzacja – na ogół IDispatch
- dual interfaces

ActiveX

Interfejsy (przypomnienie)



dispinterface IDispatch::Invoke (przypomnienie)



Kontrolki ActiveX

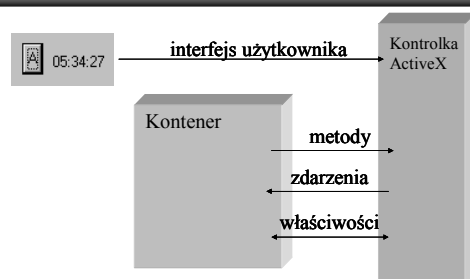
Evolution:

- VBX – Visual Basic Extension
- OLE Controls
- ActiveX Controls

Features:

- Składniki oparte na COM (IUnknown)
- Możliwość samorejestracji (np. przy pobieraniu z Internetu)

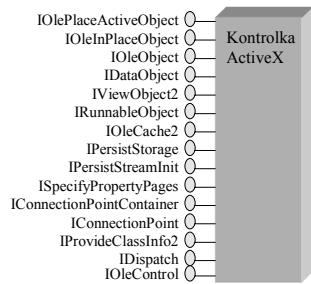
Kontrolki ActiveX



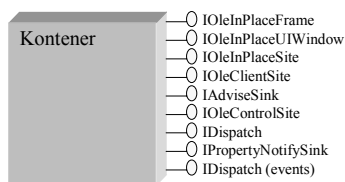
Kontrolki ActiveX

- Implementacja obejmuje wiele interfejsów COM (typu vtable)
- Z punktu widzenia użytkownika widoczne są przede wszystkim dispinterfaces

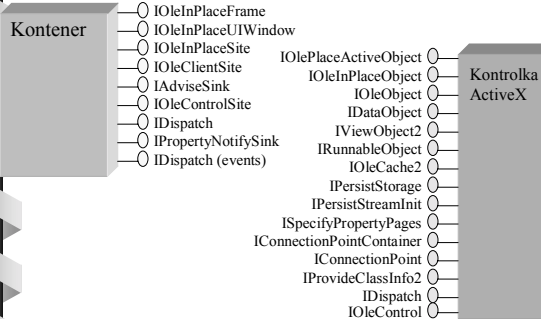
Kontrolki ActiveX



Kontrolki ActiveX



Kontrolki ActiveX



VB – implementacja COM. Podsumowanie

- ⌚ **Automatyzacja – użycie interfejsu IDispatch**
- ⌚ **Metody – zdefiniowane w dispinterface i dostępne poprzez IDispatch::Invoke**
- ⌚ **Właściwości – metody specjalnego typu dostępne w dispinterface**
- ⌚ **Zdarzenia – wywoływanie metod dispinterface kontenera (odwrócenie sytuacji: kontener jest serwerem)**

VB – implementacja COM. Podsumowanie

Przy implementacji wszystkich typów składników COM Visual Basic korzysta z elementów technologii ActiveX