Budowa i oprogramowanie komputerowych systemów sterowania

# Laboratorium 5

## Metody wymiany danych w systemach automatyki

OPC

# 1 Wprowadzenie do OPC

COM (Component Object Model) jest opracowaną przez Microsoft technologią umożliwiającą efektywną komunikację między aplikacjami.
COM definiuje komponenty programowe niezależne od języka programowania, co umożliwia włączanie do tworzonych aplikacji elementów należących do innych programów i wymianę danych między poszczególnymi obiektami za pomocą tzw. interfejsów.
Technologia COM jest wykorzystywana w wielu aplikacjach, jak np. Microsoft Office, gdzie umożliwia dynamiczne łączenie elementów np. z arkusza MS Excel do dokumentów edytora MS Word.

Component Object Model definiuje:

- binarny standard wywoływania funkcji między komponentami,
- struktury interfejsów udostępnianych przez poszczególne obiekty,
- mechanizmy jednoznacznej identyfikacji komponentów i ich interfejsów.

Obiekty COM udostępniają swoje funkcje obiektom zewnętrznym za pośrednictwem interfejsów o następujących cechach:

- Interfejs stanowi zbiór wskaźników do funkcji składowych komponentu COM (metod), zgromadzonych w tabelach vtable (Virtual Function Pointer Table).
- Interfejs nie jest klasą, nie posiada własnej implementacji, to komponent COM implementuje interfejs.
- Każdy komponent może implementować wiele interfejsów - oferować wiele zestawów usług.
- Komponenty odwołują się do interfejsów za pośrednictwem wskaźników.
- Każdy interfejs posiada własny, unikalny identyfikator (GUID). Nowa wersja interfejsu (np. z rozszerzonym zestawem funkcji) nie powoduje konfliktu ze starą wersją - otrzymuje ona inny GUID.

Technologia DCOM jest rozszerzeniem technologii COM, umożliwiającym wymianę danych za pośrednictwem sieci, podczas gdy COM dotyczył jedynie komunikacji na lokalnym komputerze.
DCOM zastępuje protokołem sieciowym lokalną komunikację między procesami, korzystając z technologii DCE RPC (Distributed Computing Environment / Remote Procedure Call).
W rzeczywistości, z punktu widzenia klienta bądź serwera, nie ma różnicy, czy druga strona znajduje się na tej samej maszynie czy zdalnie.

Komunikacja realizowana jest to za pośrednictwem tzw. RPC stubs oraz obiektów Proxy, istniejących po obu stronach.
Przykładowa komunikacja wygląda następująco:

1. Klient wywołuje lokalnie pewną funkcję.
2. Proxy przejmuje wywołanie i tworzy wiadomość do przesłania przez sieć.
3. Wiadomość jest przesyłana.
4. System operacyjny serwera otrzymuje wiadomość i przekazuje ją do stub'a serwera. Stub rozpakowuje wiadomość i lokalnie wywołuje funkcję na serwerze.

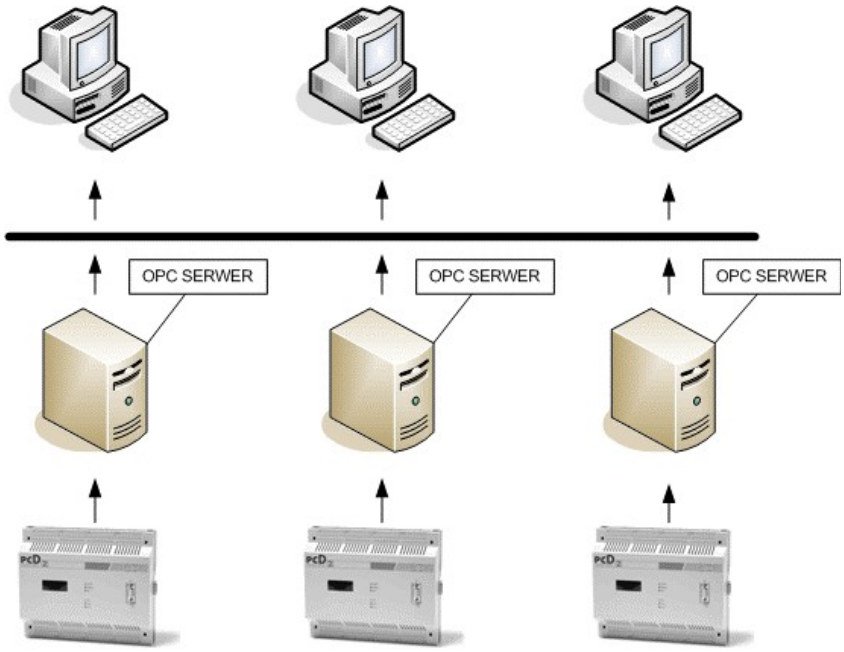
Aby umożliwić interakcję zdalnych aplikacji za pośrednictwem DCOM należy skonfigurować DCOM na każdym komputerze, np. przy pomocy programu DCOMCNFG.

OPC jest przemysłowym standardem komunikacji stworzonym przez producentów sprzętu i oprogramowania.
Utworzyli oni organizację OPC Foundation, której zadaniem jest rozwijanie tego standardu.

W chwili jej członkami jest ponad trzysta firm, wśród nich: CAS, Microsoft, GE,Siemens, Rockwell, ABB.
Standard OPC definiuje sposoby komunikacji między urządzeniami przemysłowymi, przez co pozwala uniezależnić oprogramowanie monitorujące i sterujące od producentów sprzętu.

Do zalet technologii OPC można zaliczyć m.in.:
standaryzację komunikacji i wymiany danych przemysłowych,
dużą uniwersalność i skalowalność rozwiązań,
znaczne obniżenie kosztów integracji dużych systemów przemysłowych.

OPC zostało zbudowane w oparciu o architekturę klient-serwer.
Aplikacja wymagająca dostępu do określonego urządzenia, komunikuje się przez wbudowanego w nią klienta OPC z serwerem OPC, który jest odpowiedzialny za bezpośrednią komunikację z urządzeniem.
Bez OPC każdy klient musiałby posiadać sterownik odpowiedni do komunikacji z danym urządzeniem.
Przykładowo, jeśli mamy 3 różne urządzenia i chcemy, aby trzech różnych klientów pobierało z nich dane, musimy dla każdego klienta zaimplementować sterownik właściwy dla protokołu komunikacji z każdym urządzeniem.
Ponadto, jeśli trzech klientów będzie potrzebowało tych samych danych, każde urządzenie wyśle dane wielokrotnie.



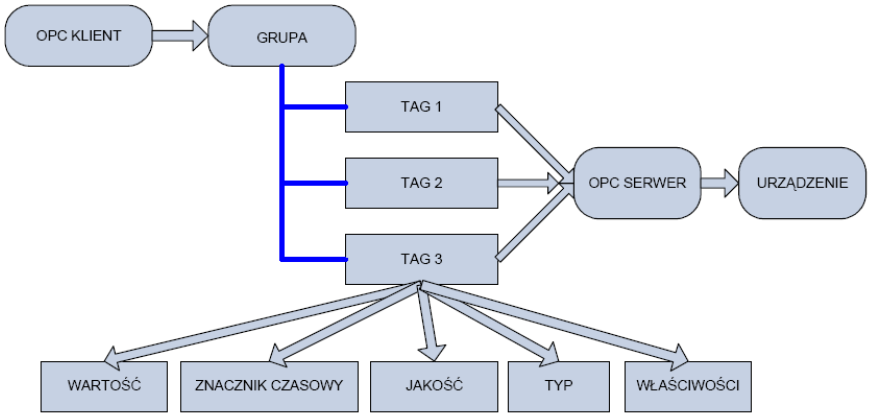

W ramach projektu zajmującego się standaryzacją OPC powstały różne specyfikacje, z których każda definiuje odrębną funkcjonalność.
Wśród istniejących specyfikacji możemy wyróżnić:

- OPC Data Access (OPC DA)
- OPC Historical Data Access (OPC HDA)
- OPC Alarms &Events (OPC A&E)
- OPC Security
- OPC BatchOPC
- OPC XML DA
- Unified Architecture (OPC UA)

OPC Data Access umożliwia dostęp do aktualnych danych generowanych w czasie rzeczywistym.
Przy pomocy OPC DA do serwera OPC kierowane są zapytania o aktualne wartości zmiennych procesowych - temperatur, ciśnień itp.
Komunikacja z każdym serwerem odbywa się w taki sam sposób, z wykorzystaniem tego samego formatu.
Klient nie musi wiedzieć, w jaki sposób serwer komunikuje się z urządzeniem.
Wielu klientów może korzystać jednocześnie z tych samych danych udostępnianych przez serwer.
OPC DA daje dostęp (możliwość odczytu lub zapisu) do pojedynczych elementów (tagów), z których każdy posiada wartość, znacznik czasowy, typ i jakość.
Znacznik czasowy może być generowany przez urządzenie lub przez serwer OPC (jeżeli dane urządzenie nie generuje znacznika).
Przy pomocy OPC DA nie jest możliwe przeglądanie wartości wcześniejszych, a jedyne aktualnych.

Klient OPC może logicznie podzielić odczytywane dane na grupy, charakteryzujące się np. różnymi czasami skanowania (czasem pomiędzy dwoma kolejnymi odczytami), trybem odczytu, itp.
Charakterystyczną cechą grupy jest jej odczyt w jednej transakcji.
W zależności od wersji OPC DA możliwe są dwa tryby odczytu danych:

- synchroniczny – odczyt występuje zawsze w jednakowych odstępach czasowych,
- asynchroniczny – odczyt występuje wtedy, gdy pewne dane ulegną zmianie – możliwa jest definicja progów, po przekroczeniu których powinien nastąpić odczyt.



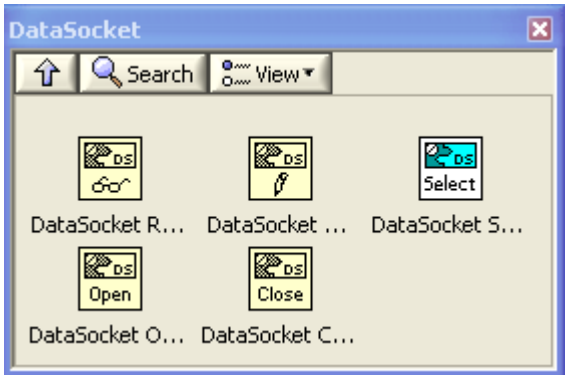

Dostęp do danych przy pomocy OPC DA może odbywać się na trzy sposoby:
z wykorzystaniem COM/DCOM,
z wykorzystaniem XML (eXtensible Markup Language) i protokołu SOAP (Simple Object Access Protocol),
za pośrednictwem technologii .NET Remoting, posiadającej szersze możliwości niż DCOM (obsługa różnych formatów i protokołów komunikacji, łatwa komunikacja za pośrednictwem Internetu).
OPC DA występuje w wielu wersjach, z których najnowszą jest wersja 3.0. (każda wersja zapewnia inny zestaw interfejsów, jednak powinna być zachowana kompatybilność wsteczna).

# 2 Obsługa OPC w Labview

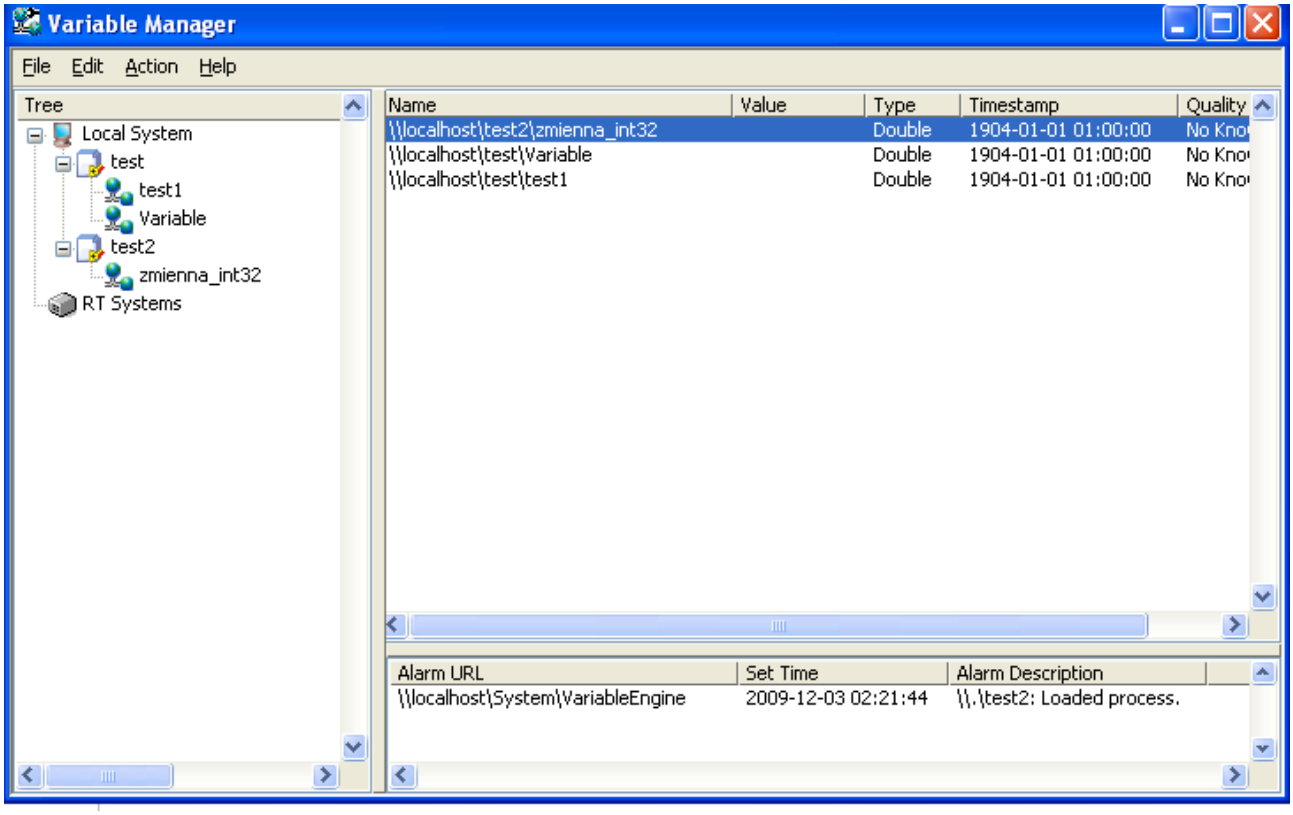
OPC Foundation wydało trzy główne wersje specyfikacji OPC. LabVIEW obecnie wspiera następujące wersje specyfikacji OPC DA:

- Wersja 1.0—DataSocket Client.
- Wersja 2.x—DataSocket Client, Variable Engine OPC Server, (DSC Module) DSC OPC Client.
- Wersja 3.0—Variable Engine OPC Server, (DSC Module) DSC OPC Client.

Funkcje Data Socket Client znajdują się w palecie *Data Communication/Data Socket.*




W celu wykorzystanie mechanizmu Variable Engine OPC Server, należy uruchomić program NI Distributed System Manager i utworzyć w nim zmiennych współdzielone.



# 3 Obsługa OPC w pakiecie Scilab

W środowisku Scilab należy doinstalować bibliotekę OPC Client poprzez narzędzie menadżera modułów zewnętrznych Atoms.

W bibliotece dostępne są funkcje

opc_add_group — Add a OPC group to OPC Server
opc_add_item — Add a OPC Items to OPC Group
opc_connect — Connect an OPC Server
opc_disconnect — Disconnect an OPC Server
opc_item_browse — Browse An OPC Server's items
opc_item_read — Read all OPC Items Value from OPC Server
opc_item_write — Write OPC Item Value to OPC Server
opc_server_browse — Browse Locale Computer's OPC Server

# 4 Zadania

1. W programie Measurement & Automation Explorer utworzyć Global Virtual Channels dla kilku sygnałów z symulowanej karty.
2. Uruchomić NI Distributed System Manager i utworzyć kilka zmiennych współdzielonych różnych typów.
3. Uruchomić program ServerExplorer, połączyć się z serwerem Variable Engine i utworzyć Grupy zawierające utworzone wcześniej kanały oraz zmienne, dla poszczególnych grup ustawić inne czasy odświeżania danych. Wykonać podobne operacje dla serwera OPCDemo i przeanalizować dostępne zmienne.
4. Uruchomić w systemie Labview i przeanalizować budowę przykładów *Industry Application/Process Control* dotyczących OPC.
5. Utworzyć własną aplikację w Labview, która przekazuje generowane w niej dane oraz sygnały z przełączników i zadajników do zmiennych współdzielonych. Sprawdzić działanie obserwując zmiany sygnałów w programie ServerExplorer oraz utworzyć drugą aplikację odczytującą zmienne współdzielone i wyświetlającą ich stan.
6. Przeanalizować funkcje klienta OPC dostępne w pakiecie Scilab i przejrzeć przykłady.
7. Utworzyć programy w systemach Labview i Scilab komunikujące się pomiędzy sobą w obydwu kierunkach i przesyłające dane o różnych typach.