



Standard Komunikacyjny OPC

Wprowadzenie

Darek Kominek, Alberta, Kanada - 2015

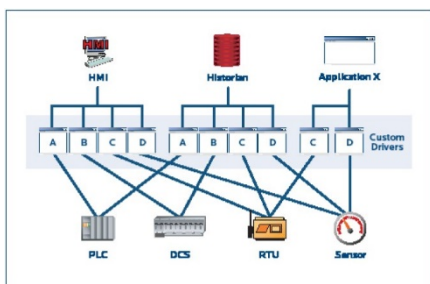
Streszczenie

Niniejszy artykuł przybliży główną ideę OPC, pokazuje czym OPC różni się od innych, często własnościowych protokołów komunikacyjnych, a także objaśnia w jaki sposób OPC pomaga wyeliminować ograniczenia tych rozwiązań dedykowanych.

Artykuł wyjaśnia jak współpracują klient i serwer OPC, aby umożliwić dostęp do danych. Korzystając z czytelnych ilustracji i minimum żargonu technicznego, artykuł ten pomaga czytelnikom na wszystkich poziomach zaawansowania technicznego szybko zrozumieć, czym jest OPC i jak mogą ten standard wykorzystać.

JAK OPC ROZWIĄDUJE PROBLEM WYMIANY DANYCH W AUTOMATYCE

W dzisiejszych czasach automatyka zajmuje znaczące miejsce w każdej działości przemysłu. Podczas gdy różne branże często korzystają z dedykowanych, specjalistycznych urządzeń czy systemów sterowania i aplikacji, wszyscy stają przed tym samym wyzwaniem - jak udostępnić dane, zarówno pomiędzy elementami systemów automatyki, jak i resztą przedsiębiorstwa.



Rysunek 1: Problem z dedykowanym sterownikiem - każda aplikacja wymaga własnego sterownika dla urządzenia lub protokołu, aby umożliwić komunikację z odpowiednim źródłem. Sterowniki nie mogą być ponownie wykorzystywane pomiędzy aplikacjami, ponieważ każda aplikacja używa własnego formatu (-ów) danych.

Zanim przyjrzymy się czym jest OPC i jak uśmierza on jeden z największych komunikacyjnych bólów głowy, warto w pierwszej kolejności przyrzeć się temu, jakie były problemy. Poniżej znajduje się lista czynników, które tradycyjnie powodowały największe problemy z udostępnianiem danych, po czym przejdziemy do krótkiego wyjaśnienia, jaki wpływ miał standard OPC na każdą z tych kwestii:

Protokoły własnościowe: Dostawcy sprzętu często korzystali z protokołów własnościowych, które pozwalały produktom z danej rodziny komunikować się między sobą, ale wymagały dedykowanych sterowników komunikacyjnych w celu wymiany danych z produktami innych dostawców. Co gorsza, nawet różne rodziny produktów tego samego dostawcy czasami nie mogły komunikować się między sobą bez zastosowania dodatkowych elementów komunikacyjnych.

OPC rozwiązuje ten problem, sprawiając, że aplikacja dla której dane są przeznaczone nie musi nic wiedzieć o tym, w jaki sposób źródło udostępnia lub jak organizuje swoje dane.

Dedykowane sterowniki: Każde połączenie pomiędzy aplikacją, a źródłem danych wymagało dedykowanego sterownika komunikacyjnego. Przykładowo dla systemu HMI konieczny był własny, dedykowany sterownik komunikacyjny obsługujący określony protokół wykorzystywany przez połączony z nim PLC. Jeżeli dane z PLC miały być dodatkowo archiwizowane, wówczas dla aplikacji archiwizującej również wymagany był dedykowany sterownik komunikacyjny. Sterownik dla HMI mógł być stosowany wyłącznie do komunikacji z HMI i nie mógł być wykorzystywany przez inne aplikacje. Jeżeli nie był dostępny dedykowany sterownik komunikacyjny dla określonych źródeł danych (np. PLC), wymiana danych była pracochłonna i kosztowna w implementacji.

OPC eliminuje konieczność stosowania dedykowanych sterowników pomiędzy aplikacją, a źródłem danych. Patrząc na rys. 1, pojedynczy standardowy sterownik komunikacyjny dla PLC mógłby być współdzielony zarówno przez HMI jak i aplikację archiwizującą. Dodatkową korzyścią wynikającą z zastosowania współdzielonego sterownika komunikacyjnego jest zmniejszone obciążenie interfejsu PLC.

„OPC eliminuje konieczność stosowania dedykowanych sterowników pomiędzy aplikacją, a źródłem danych.”

Kompleksowa integracja: Wykorzystywanie dedykowanych sterowników komunikacyjnych pomiędzy każdym z punktów końcowych oznacza, że nawet niewielka liczba urządzeń i aplikacji szybko wymagała zastosowania wielu sterowników. Ten sam system HMI działający na wielu komputerach, z których wszystkie komunikowały się z tym samym urządzeniem, wymagał licznych instalacji i konfiguracji tego samego sterownika na każdym z komputerów. Jeżeli systemy HMI komunikowały się z dodatkowymi urządzeniami, każdy z nich wymagał własnego zestawu sterowników niestandardowych dla każdego z urządzeń. To z kolei spowodowało koszmar w zakresie zarządzania wersjami.

„Ruch, a tym samym obciążenie urządzenia, jest znacznie zmniejszone przy użyciu sterowników OPC [ponieważ wszyscy klienci korzystają z jednego połączenia ze źródłem danych.]”

„OPC umożliwia komunikację najnowszych aplikacji nawet z najstarszymi systemami, przedłużając ich żywotność.”

Korzystanie z OPC znacznie upraszcza integrację, ponieważ gdy sterownik OPC dla określonego źródła danych zostanie skonfigurowany, wszystkie aplikacje, które obsługują OPC mogą rozpocząć wymianę danych z tym źródłem danych bez potrzeby stosowania dodatkowych, dedykowanych sterowników.

Obciążenie urządzenia i sterownika: Każdy ze sterowników komunikacyjnych ustanawia własne połączenie ze źródłem danych (np. PLC), poprzez które odbywa się komunikacja. Biorąc pod uwagę dużą liczbę dedykowanych sterowników komunikacyjnych wykorzystywanych w typowej instalacji, urządzenie często było „bombardowane” wieloma zapytaniami o te same informacje, przez każdą z aplikacji, z którą musiał się komunikować. Często, wiele urządzeń mogło obsłużyć jedynie ograniczoną liczbę aktywnych w tym samym czasie połączeń. Jeżeli liczba sterowników próbujących połączyć się z urządzeniem przekroczyła liczbę dostępnych połączeń, potrzebne były dalsze obejścia.

Ruch, a więc obciążenie urządzenia, można znacznie zmniejszyć przy użyciu sterowników OPC, ponieważ pojedynczy sterownik OPC dla danego źródła danych wymaga tylko jednego połączenia, w tym samym czasie udostępniając dane wielu aplikacjom końcowym.

Przestarzałość dotychczasowej infrastruktury: Kiedy producenci wypuszczają na rynek nowe produkty, w końcu przestają obsługiwać te starsze. Kiedy wychodzi nowa wersja systemu HMI, może ona wymagać własnego zestawu sterowników urządzeń, które czasami mogą już nie obsługiwać poprzedniej wersji systemu HMI.

OPC przedłuża żywotność dotychczasowych systemów, ponieważ gdy już zostanie skonfigurowane źródło OPC dla dotychczasowego systemu, pozwala ono każdej aplikacji obsługującej OPC (a większość z nich obsługuje) komunikować się z dotychczasowymi systemami, niezależnie od tego, czy aplikacja natywnie obsługuje komunikację z istniejącym systemem, czy nie. W ten sposób OPC pozwala na dalszą komunikację najnowszych aplikacji z najstarszymi systemami.

Wymiana danych w całym przedsiębiorstwie: Ponieważ w przedsiębiorstwie potrzeby związane z dostępem do danych związanych z systemami automatyki ciągle rosną, problemy z komunikacją narastają. Wynika to z tego, że aplikacje ze strony korporacyjnej nie zostały zaprojektowane do komunikacji z urządzeniami automatyki. Wymiana danych z systemami korporacyjnymi może stanowić dodatkowe obciążenie dla infrastruktury automatyki i wpływać na bezpieczeństwo systemów automatyki.

OPC umożliwia udostępnianie danych z systemów automatyki w całym przedsiębiorstwie, umożliwiając tym samym, wykorzystywanym aplikacjom biznesowym dostęp do danych z urządzeń i systemów automatyki, bez potrzeby instalowania dedykowanych sterowników. Wszystko co jest potrzebne, to interfejs OPC.

Czy istnieją jakieś rzeczywiste przykłady OPC w akcji?

Tak. Rzeczywiste przykłady, w jaki sposób wykorzystano OPC, aby rozwiązać problemy z łącznością danych zostały przedstawione w poniższych studiach przypadku (<http://www.matrikonopc.com/resources/case-studies.aspx>).

WPROWADZENIE DO OPC

Czym jest OPC?

OPC jest najbardziej popularną na świecie metodą wymiany danych opartą na standardach. Jest ona stosowana, aby zmierzyć się z jednym z największych wyzwań dla automatyki: jak komunikować się między urządzeniami, kontrolerami i/lub aplikacjami bez wpadania w pułapkę typowych problemów z łącznością opartą na dedykowanych sterownikach.

Dlaczego OPC odnosi sukces tam, gdzie przegrywają sterowniki niestandardowe?

Kluczem do sukcesu OPC w tworzeniu prawdziwie niezależnej od producenta komunikacji jest fakt, że OPC abstrahuje szczegóły realizacji źródła danych (np. PLC) oraz odbiorcy danych (np. HMI) tak, aby możliwa była wymiana danych pomiędzy nimi bez konieczności znajomości ich protokołu komunikacji natywnej oraz wewnętrznej organizacji danych. Znajduje się to w ostrej opozycji wobec podejścia bazującego na dedykowanych sterownikach, które związane jest z tworzeniem aplikacji, które z definicji wymagają natywnej komunikacji zarówno ze źródłem danych, jak i z odbiorcą danych.

Jak działa komunikacja OPC? (koncepcja)

OPC można przedstawić jako poziom „abstrakcji” warstwy, która znajduje się pomiędzy źródłem danych, a ich odbiorcą, pozwalając na wymianę danych pomiędzy nimi bez wiedzy o sobie nawzajem.

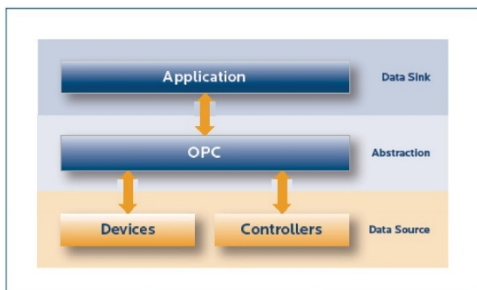
Jak działa OPC? (przeгляд funkcjonalny)

„Abstrakcja urządzenia” OPC jest realizowana za pomocą dwóch wyspecjalizowanych elementów OPC kryjących się pod nazwą Klient OPC i Serwer OPC. Każdy z nich został opisany w poniższej części. Co ważne, tylko dlatego, że źródło danych i odbiorca danych mogą komunikować się za pośrednictwem OPC, nie oznacza to, że ich odpowiednie protokoły natywne nie są już konieczne lub zostały zastąpione przez OPC. Zamiast tego, te protokoły natywne i/lub interfejsy są nadal obecne, aby komunikować się z jednym z dwóch elementów OPC. Z kolei elementy OPC wymieniają informacje pomiędzy sobą i pętla zostaje zamknięta. Dane mogą być przekazywane z aplikacji do źródła danych bez bezpośredniej ich komunikacji między sobą.

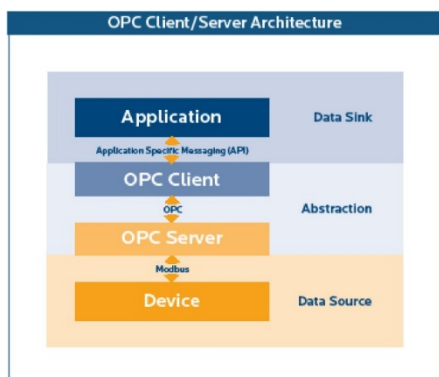
Korzyści płynące ze stosowania interfejsów OPC

Na pierwszy rzut oka, wymiana dwóch Sterowników Niestandardowych na dwa elementy OPC (Klient OPC i Serwer OPC) może nie wyglądać na znaczne ulepszenie, ale tak właśnie jest. Oto kilka kluczowych korzyści płynących z wykorzystania OPC:

1. Aplikacja z obsługą OPC może swobodnie komunikować się z każdym źródłem danych obsługującym OPC, które jest dla niej widoczne w sieci bez potrzeby stosowania jakiegokolwiek oprogramowania czy dedykowanych sterowników dla konkretnego źródła danych.
2. Aplikacje obsługujące OPC mogą komunikować się z tak dużą liczbą źródeł danych, jakiej potrzebują. Nie ma istotnych wewnętrznych ograniczeń liczby nawiązywanych połączeń ze strony OPC.
3. Obecnie standard OPC jest tak powszechny, że dla prawie każdego nowoczesnego i starszego urządzenia jest dostępny na rynku serwer OPC. Bardzo łatwo rozpocząć korzystanie z OPC.



Rysunek 2: OPC stanowi warstwę abstrakcyjną pomiędzy źródłami danych, a odbiorcami danych – umożliwia komunikację bez wiedzy każdej ze stron o protokoły natywne drugiej strony.



Rysunek 3: Architektura Klient/Serwer OPC - bliższe spojrzenie na warstwę abstrakcji OPC ujawnia dwa elementy: Klienta OPC i Serwera OPC. Standard OPC określa komunikację pomiędzy tymi dwoma komponentami.

„OPC upraszcza pracę z różnymi kategoriami danych poprzez niezależne określenie, jak każda z nich ma być przekazywana.”

4. Źródła danych obsługujące OPC mogą być zamieniane, wymieniane lub aktualizowane bez potrzeby aktualizacji sterowników wykorzystywanych przez poszczególne aplikacje (odbiorców danych) komunikujących się ze źródłem za pośrednictwem OPC. Aktualizowany musi być tylko serwer OPC dla tego źródła danych.
5. Użytkownicy mogą swobodnie wybierać najlepiej dostosowane do nich urządzenia, kontrolery i aplikacje do swoich projektów, nie martwiąc się, od którego każde z nich pochodzi i czy będzie możliwość ich komunikacji... wzajemna komunikacja stanowi teraz podstawowe założenie!

Jakie rodzaje danych obsługuje OPC?

Najczęstsze typy informacji spotykane w systemach automatyki, przekazywanych pomiędzy urządzeniami, kontrolerami i aplikacjami dzielą się na trzy obszerne kategorie:

- dane bieżące przekazywane w czasie rzeczywistym
- dane historyczne
- informacje o wystąpieniu zdarzenia lub sytuacji alarmowej

Każdy z powyższych typów obsługuje również szeroki zakres rodzajów wartości. Do typowych przykładów formatów danych można zaliczyć: liczby całkowite, zmiennoprzecinkowe, ciągi znaków, daty i różne rodzaje macierzy ... to tylko niewielki ułamek tychże formatów obsługiwanych przez OPC. OPC jest odpowiedzią na wyzwania pracy z tymi różnymi kategoriami danych poprzez niezależne określenie, jak każda z nich ma być przekazywana za pośrednictwem architektury Klienta i Serwera OPC.

Trzy specyfikacje OPC odpowiadające trzem kategoriom danych to:

1. Specyfikacja dostępu do danych bieżących (OPC DA) – opisuje sposób udostępniania bieżących wartości danych w czasie rzeczywistym
2. Specyfikacja dostępu do danych historycznych (OPC HDA) – opisuje sposób udostępniania danych historycznych poprzez aplikację archiwizującą te dane
3. Specyfikacja opisująca sposób udostępniania komunikatów i alarmów (OPC A&E) – opisuje sposób przekazywania informacji o wystąpieniu zdarzenia

Czy każdy serwer OPC obsługuje wszystkie specyfikacje OPC?

Nie. Serwery OPC nie muszą obsługiwać wszystkich specyfikacji OPC. W przeszłości najbardziej popularne były serwery OPC, które obsługiwały dostęp do danych bieżących, na kolejnym miejscu były serwery obsługujące specyfikację OPC HDA. Warto, przed wyborem serwera OPC sprawdzić, jakie specyfikacje OPC obsługuje.

Czy ma znaczenie to, jaką specyfikację OPC obsługuje klient OPC lub serwer OPC?

Tak, jest to kluczowa kwestia. Podczas gdy wszystkie trzy specyfikacje OPC (OPC DA, OPC HDA, OPC A&E) korzystają z tej samej podstawowej architektury klienta / serwera OPC do przekazywania różnych rodzajów danych, zarówno klient, jak i serwer OPC muszą obsługiwać tę samą specyfikację OPC, aby odpowiednio koordynować dane przepływające pomiędzy nimi, a także pracować prawidłowo z odbiorcą danych, jak i z ich źródłem.

„Warto, przed wyborem serwera OPC sprawdzić, jakie specyfikacje OPC obsługuje.”

SERWERY OPC

Czym jest serwer OPC?

Serwer OPC jest aplikacją, która realizuje funkcjonalność sterownika dla źródła danych, napisaną w taki sposób, aby był zgodny z jedną lub kilkoma specyfikacjami OPC. Słowo „serwer” w wyrażeniu „serwer OPC” nie odnosi się do rodzaju używanego komputera, ale odzwierciedla związek z jego odpowiednikiem w ramach OPC czyli klientem OPC.

Za co odpowiedzialne są serwery OPC?

Serwery OPC to aplikacje, które mogą być traktowane jako tłumacze pomiędzy światem OPC, a protokołem komunikacji natywnej lub interfejsem źródła danych. Ponieważ OPC obsługuje komunikację dwukierunkową, oznacza to, że serwery OPC mogą zarówno pobierać dane ze źródła, jak i dane do niego zapisywać. Relacja klienta i serwera OPC jest relacją typu Master/Slave, co oznacza, że serwer OPC będzie przekazywał dane z/do źródła danych tylko wtedy kiedy klient OPC tego sobie zażyczy.

Z jakimi rodzajami źródeł danych (urządzeń) mogą komunikować się serwery OPC?

Serwery OPC mogą komunikować się prawie z każdym źródłem danych, którego dane wyjściowe mogą być odczytywane lub zapisywane za pośrednictwem interfejsów komunikacyjnych. Krótka lista możliwych źródeł danych obejmuje: urządzenia, PLC, DCS, RTU, wagi elektroniczne, bazy danych, archiwizatory danych. Aby komunikować się z którymkolwiek z tych źródeł, wymagane jest tylko użycie serwera OPC, który korzysta z odpowiedniego protokołu natywnego lub interfejsu. Gdy tylko taki serwer OPC zostanie skonfigurowany, aplikacja wyposażona w interfejs klienta OPC (i posiadająca odpowiednie uprawnienia) może rozpocząć komunikację ze źródłem danych bez względu na to, jaki protokół/interfejs obsługuje źródło danych.

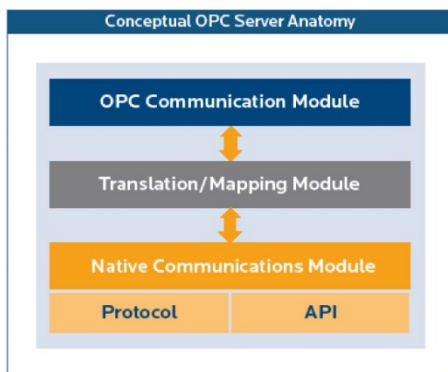
Gdzie mogę znaleźć serwer OPC dla urządzenia X?

Wielu producentów urządzeń dostarcza serwery OPC dla ich sterowników, kontrolerów czy aplikacji. Jednak są tacy, którzy tego nie robią. MatrikonOPC jest największym na świecie dostawcą wysokiej jakości serwerów OPC dla setek źródeł. Dobrym miejscem do rozpoczęcia poszukiwań serwera jest strona internetowa MatrikonOPC lub wykonanie telefonu bezpośrednio do przedstawiciela MatrikonOPC.

Jak działają serwery OPC?

Podczas gdy użytkownicy serwerów OPC, aby móc z nich korzystać, nie muszą znać szczegółów związanych z ich działaniem, zrozumienie koncepcji tego, co się dzieje wewnątrz może rzucić światło na kwestię, dlaczego jakość i wydajność serwerów OPC różnych dostawców znacznie się różni.

„Serwery OPC mogą komunikować się praktycznie z dowolnym źródłem danych.”



Rysunek 4: Architektura serwera OPC – serwer OPC można zasadniczo podzielić na trzy moduły: moduł komunikacji OPC, moduł tłumaczenia/mapowania, a także moduł komunikacji natywnej.

Architektura serwera OPC wygląda następująco:

- **Moduł komunikacji OPC:** Ta część serwera OPC jest odpowiedzialna za prawidłową komunikację z klientem OPC. Dobrze napisane serwery OPC muszą być w pełni zgodne ze specyfikacją/-ami OPC, które implementują, aby zapewnić odpowiednią komunikację z klientami OPC.
- **Moduł komunikacji natywnej:** Serwer OPC powinien korzystać z najbardziej efektywnej metody komunikacji ze źródłem danych. W niektórych przypadkach oznacza to bezpośrednie połączenie ze źródłem danych za pośrednictwem jego protokołu natywnego, podczas gdy w innych przypadkach oznacza to komunikowanie się ze źródłem danych za pośrednictwem jego sterownika niestandardowego przez Interfejs Programowania Aplikacji (API). Zazwyczaj im większe doświadczenie dostawcy serwera OPC z urządzeniem, tym lepiej serwer OPC będzie korzystał z możliwości komunikacyjnych urządzenia.
- **Moduł tłumaczenia/mapowania:** To miejsce gdzie dzieje się cała „magia” w serwerze OPC. Moduł ten ma za zadanie odpowiednio interpretować nadchodzące zapytania ze strony klienta OPC, a także zamieniać je na odpowiednie natywne zapytania, które są wysyłane do źródła danych i odwrotnie. Jeżeli odbywa się to sprawnie, dostawca OPC może utrzymać obciążenie źródła danych na poziomie minimalnym, podczas gdy przepustowość danych będzie maksymalna.

Czy serwer OPC od jednego dostawcy może komunikować się z klientami OPC innych dostawców?

Tak, zakładając, że zarówno klient OPC, jak i serwer OPC są zgodne z tymi samymi specyfikacjami OPC, powinna być możliwa ich wzajemna komunikacja, niezależnie od tego, od którego dostawcy pochodzi dany komponent OPC.

Czy serwery OPC mogą przekazywać informacje innym serwerom OPC?

Serwery OPC nie komunikują się bezpośrednio ze sobą. Zostały one zaprojektowane do komunikacji z klientami OPC. Jednakże, istnieją narzędzia OPC, takie jak MatrikonOPC Data Manager (<http://www.matrikonopc.com/products/opc-data-management/opc-data-manager.aspx>), opracowane w celu umożliwienia wymiany danych pomiędzy serwerami OPC. Proces konfiguracji takiej komunikacji w oparciu o to narzędzie jest trywialny.

KLIENCI OPC

Czym jest klient OPC?

Klient OPC to oprogramowanie napisane do komunikowania się z serwerami OPC. Wykorzystuje on sposób komunikacji zdefiniowany przez konkretną specyfikację OPC Foundation.

„Dobrze napisane serwery OPC muszą być w pełni zgodne ze specyfikacjami OPC, które implementują.”

Co robi Klient OPC?

Od strony architektury: Klient OPC reprezentuje odbiorcę danych. Inicjuje on i kontroluje komunikację z serwerem OPC w oparciu o to, czego wymaga aplikacja, której są częścią. Klient OPC przekłada zapytania komunikacyjne danej aplikacji na zapytania zgodne z OPC i wysyła je do odpowiedniego serwera OPC w celu ich przetworzenia. Z kolei, gdy serwer OPC odpowiada, klient OPC ponownie przekłada je na natywny format aplikacji, tak aby aplikacja mogła w sposób prawidłowy wykorzystać dane.

Od strony technicznej: Klient OPC, to moduł oprogramowania używany przez aplikację, który umożliwia jej komunikację z dowolnym, dostępnym w sieci i zgodnym ze specyfikacją serwerem OPC. Zazwyczaj interfejs klienta OPC jest implementowany w aplikacjach, takich jak HMI, trender, archiwizator, a także aplikacjach raportujących umożliwiając im dostęp do danych z serwerów OPC.

Częste jest określanie aplikacji wyposażonej w interfejs klienta OPC jako „klient OPC”, mimo że implementacja interfejsu OPC to mały fragment całej aplikacji.

Czy klient OPC może jednocześnie komunikować się z wieloma urządzeniami (serwerami OPC)?

Istnieją dwie części odpowiedzi:

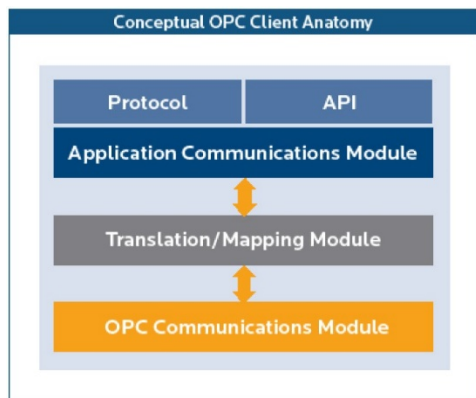
1. Po pierwsze, semantyka: Należy pamiętać, że klient OPC z punktu widzenia specyfikacji, może komunikować się tylko z serwerem OPC, a nie z urządzeniami końcowymi, czy innymi źródłami danych. Jest to konieczne, aby klient OPC mógł pozostać niezależny od protokołu. W przeciwnym razie rozwiązanie to przypominałoby to z przeszłości czyli aplikację z dedykowanym interfejsem do urządzenia.
2. Tak, klient OPC może komunikować się z wieloma serwerami OPC jednocześnie. W rzeczywistości oznacza to, że klient OPC może odczytywać i zapisywać dane z wielu źródeł danych za pośrednictwem wykorzystywanych serwerów OPC.

Jak działa klient OPC?

Podobnie jak w przypadku serwera OPC, klient OPC może być podzielony z punktu widzenia architektury na trzy moduły, które obejmują: moduł komunikacji aplikacji, moduł tłumaczenia/mapowania, a także moduł komunikacji OPC. Każdą z tych funkcji opisano poniżej:

Moduł komunikacji OPC: Choć nie jest tak mocno rozbudowany jak serwer OPC (implementacja serwera OPC jest znacznie bardziej skomplikowana), nadal kluczowe dla klienta OPC jest prawidłowe zachowanie podczas połączenia z serwerem OPC, wymiany danych z nim, a także rozłączanie się bez destabilizowania pracy serwera OPC.

Moduł natywnej komunikacji aplikacji: Klient OPC jest zazwyczaj napisany w taki sposób, aby pracować w ramach danej aplikacji, więc opiera się na połączeniach z Interfejsem Programowania Aplikacji (API), aby umożliwić przekazywanie danych z aplikacji do serwera OPC/źródła. Możliwe jest również dla podstawowego klienta OPC komunikowanie się z aplikacją za pośrednictwem protokołu, a nie API, jeżeli aplikacja obsługuje taki protokół. (Przykładem tego jest klient MatrikonOPC dla ODBC, który korzysta z instrukcji SQL dla ODBC w celu komunikacji z relacyjną bazą danych).



Rysunek 5: Architektura klienta OPC - odzwierciedlając strukturę serwera OPC, klient może być również traktowany, jako składający się z trzech modułów: moduł obsługujący natywną komunikację aplikacji, moduł tłumaczenia / mapowania oraz moduł komunikacji.

"Klienci OPC mogą komunikować się z wieloma serwerami OPC jednocześnie."

„... nie istnieje żadne teoretyczne ograniczenie liczby Serwerów OPC, z którymi może połączyć się Klient OPC.”

„Podejście „plug-and-play” zastosowane w OPC spowodowało gwałtowny wzrost popularności tego rozwiązania.”

Moduł tłumaczenia /mapowania: Kluczową funkcją klienta OPC jest Dwukierunkowa translacja informacji, gdyż aplikacja, którą reprezentuje prosi o dane do odczytu lub zleca zapis danych w źródle.

Do ilu serwerów OPC może podłączyć się klient OPC?

Krótką odpowiedź - do tyłu, do ilu potrzebuje. W ramach OPC nie istnieje żadne teoretyczne ograniczenie liczby serwerów OPC, z którymi może połączyć się klient OPC.

Czy klienci OPC mogą komunikować się bezpośrednio z innymi klientami OPC?

Nie. Komunikacja pomiędzy klientami OPC nie została zdefiniowana w OPC. Obsługiwana jest wyłącznie architektura klient-serwer OPC. Jednakże, jeżeli aplikacja powinna udostępniać dane innym klientom OPC, musi zostać wyposażona we własny serwer OPC. Interfejs serwera OPC pozwoli wtedy klientom OPC z innych aplikacji na korzystanie z tej aplikacji jako źródła danych OPC.

Gdzie jest zainstalowany klient OPC?

Klienci OPC są zazwyczaj wbudowani w aplikację, która z nich korzysta, jak na przykład HMI lub archiwizator danych procesowych. Jeżeli z jakiegoś powodu dana aplikacja nie posiada wbudowanego interfejsu klienta OPC, może on być dostępny jako opcjonalna aplikacja tego producenta lub od producenta zewnętrznego, takiego jak MatrikonOPC. Zewnętrzny klient OPC dla aplikacji zazwyczaj komunikowałby się z aplikacją za pośrednictwem jednego z jej protokołów natywnych. W tym przypadku klient OPC nie musiałby znajdować się na tym samym komputerze, co aplikacja.

PODSUMOWANIE

W niniejszym artykule wyjaśniono, w jaki sposób OPC rozwiązuje rosnące wyzwanie nowoczesnego przemysłu pod względem dostępu i wymiany danych pomiędzy urządzeniami, kontrolerami i aplikacjami, niezależnie od tego, czym jest ich natywna komunikacja i od jakiego dostawcy pochodzą. W szczegółowym opisie zilustrowano, czym jest OPC, a następnie wyjaśniono, czym są klienci OPC i serwery OPC, a także przedstawiono role, jakie odgrywają w komunikacji OPC. Podczas gdy dogłębna znajomość wewnętrznego funkcjonowania OPC nie jest konieczna do jego stosowania, nawet skrótowa znajomość jej głównych koncepcji może być przydatna.

Będące znakiem rozpoznawczym OPC podejście „plug-and-play”, spowodowało gwałtowny wzrost popularności tego rozwiązania, a to z kolei sprawiło, że jest ono najpopularniejszym interfejsem poprzez który aplikacje wymieniają dane. Dzięki wszechstronności tej technologii, OPC stosuje się na wszystkich szczeblach przedsiębiorstwa i we wszystkich branżach przemysłowych.

Wszystkie prawa zastrzeżone © Matrikon Inc. 2015