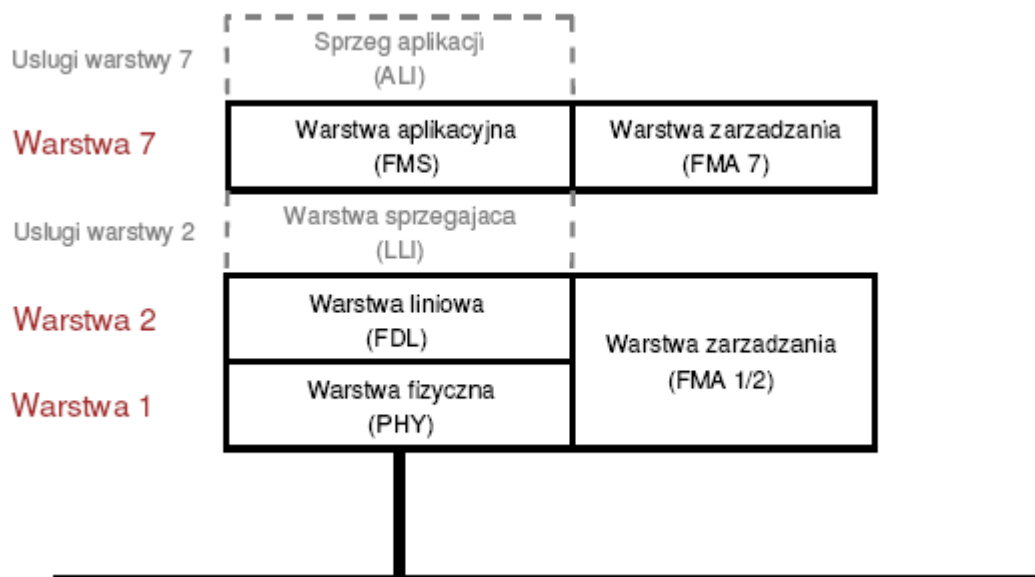


1 Opis wstępny:

Warstwa aplikacji FMS w przeciwieństwie do warstwy łącza definiuje formaty danych przesyłanych pomiędzy węzłami sieci. Jest zbudowana ponad warstwą FDL, jej działanie dominują te same ustawienia szyny (**Bus Parameters**). Sieć FMS jest typu multimaster, wyróżnia się węzły nadrzędne i podrzędne. Sieć FMS definiuje różne obiekty programowe, do których dostęp mogą mieć inne węzły sieci. Norma FMS, podobnie jak pozostałe komponenty sieci PROFIBUS definiuje jedynie usługi sieci w sposób bardzo ogólny, co pozwala na dowolny sposób jej realizacji przez producentów komponentów sieciowych, dotyczy to oczywiście również sprzęgów programowych (APIs).



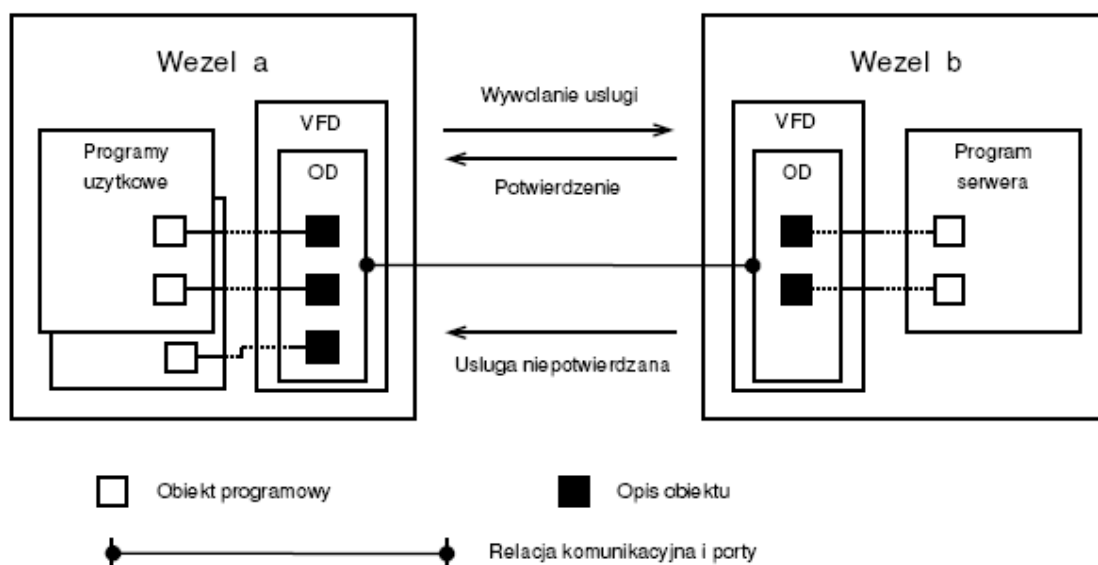
Usługi warstwy liniowej umożliwiają węzłom sieci cykliczne lub niecykliczne przekazywanie komunikatów zawierających dane o nieokreślonej strukturze wewnętrznej. Dane te są traktowane jako ciągi pewnej liczby bajtów, których treść nie jest w żaden sposób interpretowana przez programy komunikacyjne. Określenia logicznej struktury danych dokonują programy użytkowe. Uniemożliwia to automatyczną kontrolę poprawności treści komunikatów i utrudnia współpracę węzłów sieci posługujących się różnymi formatami danych.

Usługi warstwy aplikacyjnej zdefiniowane w specyfikacji FMS (*Fieldbus Message Specification*) umożliwiają użytkownikom dostęp do obiektów programowych, takich jak: zmienne, tablice lub rekordy, istniejących w innych węzłach sieci. Udostępnione do komunikacji obiekty mogą przyjmować wartości należące do ściśle określonych typów, a format transmisji wartości wszystkich typów jest dokładnie zdefiniowany. Sposób opisu usług warstwy aplikacyjnej jest bardzo ogólny i obejmuje tylko określenie ich funkcji, bez odniesienia się do sposobu ich realizacji. Dzięki temu możliwe jest efektywne wykorzystanie standardu do współpracy węzłów sieci zrealizowanych w oparciu o bardzo różne architektury sprzętowe i programowe.

Użytkownikiem sieci jest całość programów, zarówno systemowych, jak i aplikacyjnych, korzystająca w danym węźle z usług sieciowych. Definicja sieci Profibus nie odnosi się w żaden sposób do struktury programów narzuconej przez system operacyjny komputera i nie rozróżnia ani procesów wykonywanych w środowisku wielozadaniowym, ani różnych użytkowników zarejestrowanych w systemie. Każdy wykonywany program ma te same prawa dostępu do obiektów udostępnianych w sieci.

2 Rozwiązanie problemu komunikacji:

Część funkcjonalności węzła FMS związana z udostępnianiem obiektów związana jest z abstrakcyjnym VFD (Virtual Field Device). Jego zasadniczym elementem jest słownik obiektów. (Object Directory). Udostępniać obiekty może zarówno węzeł podrzędny jak i nadrzędny. Dopuszczalne jest zdefiniowanie więcej niż jednego urządzenia sieciowego w węźle. Wirtualne urządzenie sieciowe (VFD) zdefiniowane w specyfikacji FMS jest odpowiednikiem wirtualnego urządzenia produkcyjnego (VMD) zdefiniowanego w specyfikacji MMS.



Użytkownik sieci, tzn. wykonywane w danym węźle programy, definiuje różne obiekty programowe, takie jak zmienne proste lub tablice, i udostępnia niektóre z nich użytkownikom innych węzłów. Ta część funkcjonalności użytkownika, która jest związana z udostępnianiem obiektów innym użytkownikom jest w abstrakcyjny sposób reprezentowana przez wirtualne urządzenie sieciowe (*virtual field device - VFD*). Zasadniczym elementem wirtualnego urządzenia sieciowego jest słownik obiektów (*object directory - OD*), opisujący wszystkie zdefiniowane w danym węźle obiekty programowe, które mogą stać się przedmiotem komunikacji.

Usługi warstwy aplikacyjnej umożliwiają użytkownikom innych węzłów dostęp do obiektów zdefiniowanych w słowniku obiektów OD danego węzła. Usługi są wykonywane zgodnie ze schematem klient-serwer, w którym serwer udostępnia swoje obiekty dla działań klienta. Większość usług stanowią usługi potwierdzone, których wywołanie przez klienta przekazuje do serwera komunikat zawierający żądanie wykonania operacji, np. odczytu lub zapisu wskazanego obiektu, a potwierdzenie wysłane przez serwer przesyła do klienta rezultat wykonania tej operacji. Stosunkowo niewielką grupę stanowią usługi nie potwierdzone, których wykonanie przesyła od serwera do klienta komunikat informujący o stanie lub wartości jakiegoś obiektu serwera. Te same programy mogą wywoływać różne usługi, pełniąc w nich rolę klienta lub serwera.

Komunikacyjną strukturę sieci, widzianą przez wirtualne urządzenie sieciowe, wyznacza zdefiniowana w tym węźle lista relacji komunikacyjnych. Każda relacja łączy parę portów w dwóch współpracujących węzłach i tworzy odrębny kanał komunikacyjny, poprzez który wykonywane w jednym węźle programy mogą wywoływać usługi adresowane do drugiego węzła. Wykonanie każdej usługi warstwy aplikacyjnej polega na przesyłaniu lub wymianie pojedynczych komunikatów. Maksymalna długość użytecznych danych zawartych w jednym komunikacie wynosi ok. 220 bajtów.

Specyfikacja FMS dopuszcza możliwość zdefiniowania kilku wirtualnych urządzeń sieciowych w tym samym węźle sieci. Każda relacja komunikacyjna musi być związana z tylko jednym urządzeniem VFD.

3 Połączenia oraz relacje komunikacyjne:

Warstwa liniowa implementuje w każdym węźle sieci zestaw portów, przez który programy wykonywane w tym węźle mogą wysyłać i odbierać dane do i od programów wykonywanych w innych węzłach. Każdy przekaz danych między dwoma węzłami sieci musi przejść przez jakiś port w jednym i w drugim węźle. Para portów po jednym w każdym z dwóch współpracujących węzłów — tworzy w warstwie aplikacyjnej relację komunikacyjną (*communication relationship*) dostępną dla wykonywanych w tych węzłach programów. Definicja relacji komunikacyjnej obejmuje: numer portu własnego węzła, adres i numer portu węzła docelowego oraz opis wszystkich usług warstwy aplikacyjnej, które mogą być w tej relacji wywoływane.

Relacje komunikacyjne definiuje się w każdym węźle oddzielnie, na ogół podczas konfigurowania węzła do pracy w sieci. Zestaw wszystkich relacji zdefiniowanych w danym węźle tworzy listę relacji komunikacyjnych (*communication relationship list — CRL*) tego węzła. Numer na tej liście (*communication reference — CREF*) jest identyfikatorem relacji używanym jako parametr adresowy w wywołaniach wszystkich usług warstwy aplikacyjnej. Numer CREF=0 ma znaczenie specjalne i jest zarezerwowany dla nagłówka listy. Numer CREF=1 jest zarezerwowany dla relacji obsługującej odległe funkcje zarządzające. Relacje używane do

komunikowania się programów użytkowników zaczynają się od numeru CREF=2.

Niemal wszystkie relacje komunikacyjne zdefiniowane w warstwie aplikacyjnej są relacjami połączeniowymi. Nieliczne relacje bezpołączeniowe są używane do rozgłaszania. Relacje połączeniowe (*connection-oriented*) gwarantują poprawność nadawania i odbierania danych przekazywanych w sieci podczas wykonania usługi. Wykorzystanie relacji połączeniowej przez współpracujące programy musi obowiązkowo rozpoczynać się od nawiązania w tej relacji połączenia (*connection*), którego funkcjonowanie jest nadzorowane przez oprogramowanie komunikacyjne. Połączenie w relacji komunikacyjnej, nawiązywane jest poprzez usługę *Initiate*. W każdej chwili można je przerwać usługą *Abort*. Konfigurowanie sieci FMS polega na ustaleniu zbiorów atrybutów dla wszystkich relacji komunikacyjnych w sieci.

Nadzór nad funkcjonowaniem połączenia obejmuje: sprawdzanie poprawności przekazywania komunikatów, inicjowanie retransmisji w razie wystąpienia błędów oraz kontrolę sprawności połączenia w okresach, w których połączenie to nie jest wykorzystywane. Raz nawiązane połączenie jest symetryczne i dwukierunkowe, tzn. każdy ze współpracujących programów może pełnić rolę klienta lub serwera usług wywoływanych w tym połączeniu. Połączenie istnieje od chwili nawiązania do chwili usunięcia.

Relacje bezpołączeniowe (*connectionless*) nie gwarantują poprawności przekazu danych. Wykonanie usługi sprowadza się tu do nadania komunikatu przenoszącego dane, bez jakiegokolwiek kontroli poprawności odbioru. Relacje bezpołączeniowe wykorzystuje się wyłącznie do przekazywania danych skierowanych do odbiorców w wielu węzłach sieci. Relacje bezpołączeniowe są niesymetryczne w tym sensie, że rola - klient lub serwer - jaką wykonywane programy mogą pełnić w tej relacji jest ustalona na stałe podczas konfigurowania listy CRL.

Różne tryby pracy i komunikowania się węzłów sieci, cykliczny lub niecykliczny stwarza różne wymagania na sposób zorganizowania komunikacji, co znajduje wyraz wysterowaniu różnych typów relacji komunikacyjnych. W relacjach tych mogą być wywoływane te same usługi, jednak sposób ich wykonywania może być w różnych realizacjach różny.

Specyfikacja FMS definiuje następujące typy realizacji komunikacyjnych:

MSCY, MSCY_SI - relacje cykliczne, łączące węzeł nadrzędny z podrzędnym
MSAC, MSAC_SI - relacje niecykliczne, łączące węzeł nadrzędny z podrzędnym
MMAC - relacje niecykliczne, łączące dwa węzły nadrzędne
MULT, BRCT - relacje niecykliczne, używane do rozgłaszania w sieci

W relacjach typu **MS**. wszystkie usługi potwierdzone oraz usługi niepotwierdzone o niskim priorytecie są realizowane za pomocą usługi CSRD warstwy liniowej. Usługi niepotwierdzone o wysokim priorytecie są realizowane za pomocą usługi SRD. Adresy węzłów podrzędnych i numery portów wykorzystywanych w tych węzłach we wszystkich relacjach tego typu muszą być umieszczone na liście odpytywania węzła nadrzędnego. Cała lista odpytywania musi być zdefiniowana w jednym porcie węzła, który musi być wykorzystywany przez wszystkie relacje tego typu w danym węźle.

W relacjach typu **MMAC** wszystkie usługi FMS, zarówno potwierdzone, jak niepotwierdzone, realizuje się za pomocą usługi SDA warstwy liniowej. Warto zauważyć, że dzięki temu wykonanie usług niepotwierdzanych odbywa się ze zwrotnym potwierdzeniem poprawności odebrania komunikatu sieciowego przez warstwę liniową.

W relacjach typu **MULT** i **BRCT** dozwolone są tylko niepotwierdzone usługi FMS, które realizuje się za pomocą usługi SDN warstwy liniowej.

Relacje cykliczne **MSCY** i **MSCY_SI**

Relacje **MSCY** (*Master Slave Cyclic*) i **MSCY_SI** (*Master Slave Cyclic with Slave Initiative*) są relacjami połączeniowymi. Wymiana danych między współpracującymi węzłami jest możliwa w takiej relacji dopiero po nawiązaniu połączenia za pomocą usługi *Initiate*. Po nawiązaniu połączenia program wykonywany w węźle nadrzędnym może wywołać usługi potwierdzone *Read* i *Write* oraz wszystkie usługi niepotwierdzone. Kolejna usługa potwierdzana może być wywołana dopiero po zakończeniu wykonywania usługi poprzedniej. Program wykonywany w węźle podrzędnym może w relacji **MSCY** jedynie usunąć połączenie za pomocą usługi *Abort*, w e relacji **MSCY_SI** może ponadto wywoływać wszystkie usługi niepotwierdzone. Usługę *Initiate* może wywołać tylko program wykonywany w węźle nadrzędnym. Wykonanie tej usługi uruchamia proces odpytywania współpracującego węzła podrzędnego (usługa CSRD warstwy liniowej), kontynuowany aż do chwili usunięcia połączenia. Dzięki ciągłemu odpytywaniu, węzeł podrzędny może w relacji **MSCY_SI** przekazywać do węzła nadrzędnego komunikaty przenoszące dane usług niepotwierdzanych. W relacji **MSCY** wywołanie usług niepotwierdzanych w węźle podporządkowanym jest zabronione.

Węzeł nadrzędny (<i>master</i>)			Węzeł podrzędny (<i>slave</i>)	
FMS	FDL	sieć	FDL	FMS
→ Read.req	Fdl_Send_Update.req	→ SRD (dane) → ← SRD (ϕ) ←	Fdl_Data_Reply.ind	Read.ind → Read.res ←
		→ SRD (ϕ) → ← SRD (dane) ←	Fdl_Reply_Update.req	
← Read.con	Fdl_Cyc_Data_Reply.con		Fdl_Data_Reply.ind	Read.ind → Read.res ←
→ Read.req ← Read.con		→ SRD (ϕ) → ← SRD (dane) ←	Fdl_Reply_Update.req	
	Fdl_Cyc_Data_Reply.con		Fdl_Data_Reply.ind	Read.ind →
→ Read.req ← Read.con		→ SRD (ϕ) → ← SRD (ϕ) ←		

Harmonogram realizacji usługi *Read* w relacji cyklicznej MSCY jest pokazany w tablicy powyżej. Pierwsze wywołanie operacji *Read.req* przez program klienta przenosi do węzła podrzędnego indeks żądanego obiektu. Odpowiedź węzła podrzędnego, przekazana w tym samym cyklu odpytywania, nie zawiera żadnych danych. Oprogramowanie węzła podrzędnego zapamiętuje indeks w obszarze buforowym (*Image Data Memory - IDM*) i za pomocą operacji *Read.ind* zawiadamia program użytkownika o wywołaniu usługi. Wartość odczytywanego obiektu, przekazana przez program użytkownika za pomocą operacji *Read.res*, jest zapisywana w buforze wyjściowym portu (operacja *Fdl_Reply_Update.req*) i przenoszona w sieci w następnym cyklu odpytywania. Oprogramowanie węzła nadrzędnego zapamiętuje odebraną wartość w obszarze buforowym IDM i przekazuje ją do programu klienta, za pomocą operacji *Read.con*. Wysłanie do węzła nadrzędnego danych, przekazanych przez operację *Read.res*, ponownie inicjuje w węźle podrzędnym operację *Read.ind*, w odpowiedzi, na którą program użytkownika może - za pomocą operacji *Read.res* - przekazać nową wartość obiektu. Tak jak poprzednio, nowe dane są przenoszone w sieci w najbliższym cyklu odpytywania i zapisywane w obszarze IDM węzła nadrzędnego. Proces ten jest powtarzany permanentnie bez względu na działania programu klienta. Kolejne wywołania operacji *Read.req* przez program klienta nie są już bezpośrednio związane z przesyłaniem jakichkolwiek komunikatów sieciowych. Każde następne wywołanie powoduje odczytanie danych zapisanych w obszarze IDM własnego węzła i przekazanie ich do programu za pomocą operacji *Read.con*. Sposób wykonania usług niepotwierdzanych wywołanych w węźle nadrzędnym jest analogiczny do sposobu wykonania operacji *.req* i *.ind* pierwszego wywołania usługi *Read*. Oczywiście, nie występują tu operacje *.res* i *.con*. Każde wywołanie usługi

powoduje przekazanie danych w sieci i zapisanie ich w buforze wyjściowym portu węzła odbierającego. Priorytet transakcji odpytywania realizującej usługę niepotwierdzaną jest równy priorytetowi wywołania usługi. Sposób wykonania usługi niepotwierdzanych wywołanych w węźle podrzędnym jest analogiczny do sposobu wykonania operacji *.res* lub *.con* pierwszego wywołania usługi *Read*. To znaczy, wywołanie usługi powoduje zapisanie danych w odpowiednim buforze wyjściowym portu. Dane te będą przesłane do węzła nadrzędnego w jednym z następných cykli odpytywania. Zgodnie z regułami odpytywania dane zapisane w buforze wyjściowym o wysokim priorytecie (związane z usługami o wysokim priorytecie) są przekazywane przed danymi zapisanymi w buforze o priorytecie niskim. Sposób wykonania usług potwierdzanych w relacjach cyklicznych odpowiada koncepcyjnie sieciowej implementacji pamięci wspólnej, w której proces odpytywania zapewnia spójność kopii, przechowywanej w węźle nadrzędnym, z wartościami zmiennych istniejących w węźle podrzędnym. Definicja usług stwarza jednak niebezpieczeństwo niewykrywalnej dezaktualizacji kopii w wypadku przerwania komunikacji. Dla uniknięcia tego niebezpieczeństwa standard narzuca obowiązek monitorowania połączeń w relacjach cyklicznych, z zadaniem okresem kontroli (*Control Interval - ci*). W tym celu oprogramowanie węzła nadrzędnego musi zagwarantować wykonywanie co najmniej jednego odpytania w każdym okresie *ci*. Niedotrzymanie tego warunku jest wykrywalne przez obydwie węzły i powoduje usunięcie połączenia. Czas *ci* jest jednym z parametrów konfiguracyjnych relacji komunikacyjnej. Wartość tego parametru musi być na obu końcach relacji taka sama.

Relacje MSAC i MSAC_SI

Relacje MSAC (*Master Slave Acyclic*) i MSAC_SI (*Master Slave Acyclic with Slave Initiative*) są relacjami połączeniowymi. Wymiana danych między współpracującymi węzłami musi być w takiej relacji poprzedzona nawiązaniem połączenia za pomocą usługi *Initiate*. Po nawiązaniu połączenia program wykonywany w węźle nadrzędnym może wywoływać wszystkie usługi potwierdzone i niepotwierdzone. Usługi mogą być wykonywane równolegle, tzn. program może wywołać następną usługę zanim otrzyma potwierdzenie wykonania poprzedniej. Program wykonywany w węźle podrzędnym może w relacji MSAC jedynie usunąć połączenie za pomocą usługi *Abort*, w relacji MSAC_SI może ponadto wywoływać usługi niepotwierdzone. Usługę *Initiate* może wywołać tylko program wykonywany w węźle nadrzędnym. Wykonywanie tej usługi w relacji MSAC_SI uruchamia proces odpytywania współpracującego węzła podrzędnego (usługa CSRD warstwy liniowej), który może dzięki temu przekazywać do węzła nadrzędnego komunikaty przenoszące dane usług niepotwierdzanych. W relacji MSAC proces odpytywania jest uruchamiany w chwili wywołania dowolnej usługi potwierdzonej (w tym również usługi *Initiate*) i jest kontynuowany tylko przez czas niezbędny do wykonania tej usługi. Wykonanie usług potwierdzanych przebiega podobnie do wykonania pierwszego wywołania usługi *Read* w relacji MSCY, tzn. każdemu wywołaniu usługi odpowiada w sieci odrębną parą przesyłać żądanie - potwierdzenie. Z każdym wywołaniem usługi potwierdzonej jest związany

indywidualny identyfikator (*Invoke ID*), przekazywany w każdym komunikacie sieciowym. Identyfikator ten umożliwia jednoznaczne przyporządkowanie komunikatów do równolegle realizowanych usług potwierdzanych.

Węzeł nadrzędny (<i>master</i>)		sieć	Węzeł podrzędny (<i>slave</i>)	
FMS	FDL		FDL	FMS
← InfoR.ind	Fdl_Cyc_Data_Reply.con Fdl_Send_Update.req Fdl_Send_Update.con Fdl_Data_Reply.con	→ SRD (ϕ) → ← SRD (dane) ← → SRD (ack) → ← SRD (ϕ) ←	Fdl_Reply_Update.req Fdl_Data_Reply	InfoR.req ←

Harmonogram realizacji usługi *Read* w relacji cyklicznej MSCY jest pokazany w tablicy powyżej. Pierwsze wywołanie operacji *Read.req* przez program klienta przenosi do węzła podrzędnego indeks żądanego obiektu. Odpowiedź węzła podrzędnego, przekazana w tym samym cyklu odpytywania, nie zawiera żadnych danych. Oprogramowanie węzła podrzędnego zapamiętuje indeks w obszarze buforowym (*Image Data Memory - IDM*) i za pomocą operacji *Read.ind* zawiadamia program użytkownika o wywołaniu usługi. Wartość odczytywanego obiektu, przekazana przez program użytkownika za pomocą operacji *Read.res*, jest zapisywana w buforze wyjściowym portu (operacja *Fdl_Reply_Update.req*) i przenoszona w sieci w następnym cyklu odpytywania. Oprogramowanie węzła nadrzędnego zapamiętuje odebraną wartość w obszarze buforowym IDM i przekazuje ją do programu klienta za pomocą operacji *Read.con*. Wyśłanie do węzła nadrzędnego danych, przekazanych przez operację *Read.res*, ponownie inicjuje w węźle podrzędnym operację *Read.ind*, w odpowiedzi na którą program użytkownika może za pomocą operacji *Read.res* - przekazać nową wartość obiektu. Tak jak poprzednio, nowe dane są przenoszone w sieci w najbliższym cyklu odpytywania i zapisywane w obszarze IDM węzła nadrzędnego. Proces ten jest powtarzany permanentnie bez względu na działania programu klienta. Kolejne wywołania operacji *Read.req* przez program klienta nie są już bezpośrednio związane z przesyłaniem jakichkolwiek komunikatów sieciowych. Każde następne wywołanie powoduje odczytanie danych zapisanych w obszarze IDM własnego węzła i przekazanie ich do programu za pomocą operacji *Read.con*. Sposób wykonania usług niepotwierdzanych wywołanych w węźle nadrzędnym jest analogiczny do sposobu wykonania operacji *.req* i *.ind* pierwszego wywołania usługi *Read*. Oczywiście, nie występują tu operacje *.res* i *.con*. Każde wywołanie usługi powoduje przekazanie danych w sieci i zapisanie ich w buforze wejściowym portu węzła odbierającego. Priorytet transakcji odpytywania realizującej usługę niepotwierdzoną jest równy priorytetowi wywołania usługi. Sposób wykonania usługi niepotwierdzanych wywołanych w węźle podrzędnym jest analogiczny do sposobu wykonania operacji *.res* lub *.con* pierwszego wywołania usługi *Read*. To

znaczy, wywołanie usługi powoduje zapisanie danych w odpowiednim buforze wyjściowym portu. Dane te będą przesyłane do węzła nadrzędnego w jednym z następujących cykli odpytywania. Zgodnie z regułami odpytywania dane zapisane w buforze wyjściowym o wysokim priorytecie (związane z usługami o wysokim priorytecie) są przekazywane przed danymi zapisanymi w buforze o priorytecie niskim. Sposób wykonania usług potwierdzanych w relacjach cyklicznych odpowiada koncepcyjnie sieciowej implementacji pamięci wspólnej, w której proces odpytywania zapewnia spójność kopii, przechowywanej w węźle nadrzędnym, z wartościami zmiennych istniejących w węźle podrzędnym. Definicja usług stwarza jednak niebezpieczeństwo niewykrywalnej dezaktualizacji kopii w wypadku przerwania komunikacji. Dla uniknięcia tego niebezpieczeństwa standard narzuca obowiązek monitorowania połączeń w relacjach cyklicznych, z zadaniem okresem kontroli (*Control Interval - ci*). W tym celu oprogramowanie węzła nadrzędnego musi zagwarantować wykonywanie co najmniej jednego odpytania w każdym okresie *ci*. Niedotrzymanie tego warunku jest wykrywalne przez obydwa węzły i powoduje usunięcie połączenia. Czas *ci* jest jednym z parametrów konfiguracyjnych relacji komunikacyjnej. Wartość tego parametru musi być na obu końcach relacji taka sama.

Relacje MSAC i MSAC_SI

Relacje MSAC (*Master Slave Acyclic*) i MSAC_SI (*Master Slave Acyclic with Slave Initiative*) są relacjami połączeniowymi. Wymiana danych między współpracującymi węzłami musi być w takiej relacji poprzedzona nawiązaniem połączenia za pomocą usługi *Initiate*. Po nawiązaniu połączenia program wykonywany w węźle nadrzędnym może wywoływać wszystkie usługi potwierdzane i niepotwierdzone. Usługi mogą być wykonywane równolegle, tzn. program może wywołać następną usługę zanim otrzyma potwierdzenie wykonania poprzedniej. Program wykonywany w węźle podrzędnym może w relacji MSAC jedynie usunąć połączenie za pomocą usługi *Abort*, w relacji MSAC_SI może ponadto wywoływać usługi niepotwierdzone. Usługę *Initiate* może wywołać tylko program wykonywany w węźle nadrzędnym. Wykonywanie tej usługi w relacji MSAC_SI uruchamia proces odpytywania współpracującego węzła podrzędnego (usługa CSRD warstwy liniowej), który może dzięki temu przekazywać do węzła nadrzędnego komunikaty przenoszące dane usług niepotwierdzonych. W relacji MSAC proces odpytywania jest uruchamiany w chwili wywołania dowolnej usługi potwierdzanej (w tym również usługi *Initiate*) i jest kontynuowany tylko przez czas niezbędny do wykonania tej usługi. Wykonanie usług potwierdzanych przebiega podobnie do wykonania pierwszego wywołania usługi *Read* w relacji MSCY, tzn. każdemu wywołaniu usługi odpowiada w sieci odrębna para przesyłać żądanie - potwierdzenie. Z każdym wywołaniem usługi potwierdzanej jest związany indywidualny identyfikator (*Invoke ID*), przekazywany w każdym komunikacie sieciowym. Identyfikator ten umożliwia jednoznaczne przyporządkowanie komunikatów do równolegle realizowanych usług potwierdzanych zwrotnym potwierdzeniem odbioru na poziomie warstwy liniowej.

Węzeł nadrzędny (<i>master</i>)			Węzeł podrzędny (<i>slave</i>)	
FMS	FDL	sieć	FDL	FMS
→ Read.req	Fdl_Data_Ack.req	→ SDA (dane) →		
	Fdl_Data_Ack.req	← SDA (ack) ←	Fdl_Data_Ack.ind	Read.ind →
			Fdl_Data_Ack.req	Read.res ←
		← SRD (data) ←		
		→ SRD (ack) →	Fdl_Data_Ack.con	
← Read.con	Fdl_Data_Ack.req			

Relacje MULT i BRCT

Relacje MULT i BRCT są relacjami bezpołączeniowymi, z których każda może łączyć wiele węzłów nadrzędnych. Węzły te mogą wywoływać tylko usługi niepotwierdzone. Wykonanie usługi sprowadza się do wystania komunikatu rozgłaszanego w sieci za pomocą usługi SDN warstwy liniowej.

4 Obiekty:

Warstwa aplikacyjna separuje użytkowników od problemów związanych z techniką komunikacji w sieci. Usługi implementowane przez tę warstwę nie odwołują się do pojęcia „komunikatu sieciowego” lecz określają działania na obiektach (*objects*), definiowanych w dziedzinie programu. Wszystkie obiekty, które mogą być przetwarzane za pomocą usług warstwy aplikacyjnej, muszą być z góry określone, a ich opisy muszą być wcześniej wprowadzone do słownika obiektów węzła, w którym te obiekty istnieją. Rodzaje obiektów FMS:

- zmienna (prosta, tablica, rekord, lista zmiennych) . obiekty te służą do wymiany zmiennych pomiędzy uczestnikami procesu technologicznego. Mogą występować w różnych typach, również zdefiniowanych w słowniku OD
- typ danych (skalary lub struktura rekordu)
- zdarzenie (*event*) . obiekt reprezentujący wyjątek lub stan szczególny zaistniały we współpracującym programie
- domena (*domain*) . to obszar (*segment*) pamięci, do którego można załadować np. zestaw parametrów konfiguracyjnych lub program
- program . np. załadowany do domeny, który można zatrzymać lub uruchomić.
- obiekt fizyczny o nieznanym strukturze wewnętrznej

Na podanych obiektach można realizować usługi warstwy aplikacyjnej. Z punktu

widzenia naszego źródła danych interesują nas jedynie zmienne (*variables*) i typy danych.

Obiekty mogą być adresowane na dwa sposoby:

- przez indeks . określenie położenia w OD
- przez nazwę

Dostęp do obiektów może być autoryzowany, poprzez zdefiniowanie grup i parametrów

dostępu, użytkownicy zainteresowani pełnym dostępem muszą podać hasło. W naszym źródle

danych nie będziemy jednak korzystać z mechanizmów autoryzacji.

Słownik obiektów OD ma ustaloną strukturę, dzieli się on na 5 segmentów:

OD-ODES nagłówek wraz z opisem słownika

ST-OD opisy typów danych

Implementacja źródła danych systemu SCADA dla sieci PROFIBUS

S-OD opisy zmiennych, domen i zdarzeń

DV-OD opisy list zmiennych

DP-OD opisy programów

Standard FMS definiuje 14 standardowych typów danych, zatem najczęściej pierwsze 14 wpisów w sekcji **ST-OD** to ich opisy (choć nie jest to konieczność).

Oprócz działań na obiektach, takich jak np. zapisanie lub odczytanie wartości zmiennej, usługi warstwy aplikacyjnej umożliwiają również odczytywanie i zapisywanie opisów obiektów. Opis obiektu obejmuje następujące elementy: rodzaj obiektu, typ danych związanych z tym obiektem, adres i długość danych w pamięci oraz atrybuty praw dostępu. Wszystkie obiekty powinny być zdefiniowane lokalnie i opisane w słowniku obiektów podczas konfigurowania węzła sieci (wirtualnego urządzenia sieciowego). Wyjątkiem są listy zmiennych i programy, które mogą być tworzone i usuwane tylko zdalnie, przez wywołanie odpowiednich usług warstwy aplikacyjnej.

Adresowanie obiektów. Podstawowym trybem adresowania obiektów jest wskazywanie ich przez podanie indeksu (numeru) obiektu w słowniku obiektów. Indeks obiektu jest 16 bitową liczbą bez znaku. Adresowanie indeksowe stosuje się do wszystkich obiektów, z wyjątkiem obiektów fizycznych, które mogą być adresowane wyłącznie za pomocą 32 bitowych adresów, określających ich położenie w pamięci.

Oprócz adresowania za pomocą indeksów, obiekty mogą być opcjonalnie adresowane przez nazwy. Jeżeli urządzenie VFD realizuje tę opcję, to nazwą obiektu może być dowolny ciąg znaków, o długości określonej w nagłówku słownika obiektów. Maksymalna długość nazwy nie może nigdy przekraczać 32 znaków. Ciąg złożony z pojedynczej spacji ma znaczenie szczególne i oznacza brak nazwy.

Typy danych. Specyfikacja FMS definiuje 14 standardowych typów danych. Zmienne proste, elementy tablic i składowe rekordów mogą przyjmować wartości tylko wymienionych typów. Oprócz typów standardowych, użytkownik może definiować struktury rekordowe, traktowane jako typy danych rekordów. Definicja struktury rekordu obejmuje następujące atrybuty:

- rozmiar rekordu (liczba składowych),
- lista opisów kolejnych składowych: indeks typu wartości i długość (liczba bajtów).

Standardowe typy danych w sieci FMS

Nazwa	Indeks	Długość
Boolean (OD_BOOL)	1	1
Integer8 (OD_INT8)	2	1
Integer16 (OD_INT16)	3	2
Integer32 (OD_INT32)	4	4
Unsigned8 (OD_USIGN8)	5	1
Unsigned16 (OD_USIGN16)	6	2
Unsigned32 (OD_USIGN32)	7	4
Floating Point (OD_FLOAT)	8	4
Visible String (OD_VSTRING)	9	n
Octet String (OD_OSTRING)	10	n
Date (OD_DATE_TYPE)	11	7
Time Of Day (OD_TIME_OF_DAY)	12	4 lub 6
Time Of Day (OD_TIME_OF_DAY)	13	4 lub 6
Bit String (OD_BSTRING)	14	n

Autoryzacja dostępu. Programy będące klientami usług warstwy aplikacyjnej są podzielone na osiem grup różniących się prawami dostępu do obiektów zdefiniowanych w serwerze. Ten sam program może należeć do więcej niż jednej grupy — przynależność programu opisuje ośmiobitowa maska, w której bity o wartości 1 wskazują te grupy, do których dany program należy. Maska dostępu nie jest stałą cechą programu, który może legitymować się różnymi maskami w różnych połączeniach.

Każda próba dostępu do obiektu zdefiniowanego po stronie serwera jest poprzedzona kontrolą uprawnień klienta. Ograniczenia, którym te uprawnienia podlegają są określone przez atrybuty praw dostępu zapisane w opisie obiektu w słowniku OD. Atrybuty te dzielą potencjalnych klientów na trzy kategorie: użytkowników uprzywilejowanych, użytkowników należących do uprawnionych grup dostępu i pozostałych. Z każdą kategorią jest związany odrębny zestaw

znaczników uprawniających do grup dostępu i pozostałych. Z każdą kategorią jest związany odrębny zestaw znaczników uprawniających do zapisu, odczytu i wykonywania.

W skład atrybutów praw dostępu wchodzi:

- ośmiobitowe hasło, znane tylko użytkownikom uprzywilejowanym,
- ośmiobitowa maska wskazująca uprawnione grupy dostępu,
- wskaźniki praw dostępu do użytkowników należących do poszczególnych

kategorii.

Identyfikacja użytkowników następuje w chwili nawiązania połączenia za pomocą usługi *Initiate*. W toku realizacji tej usługi, użytkownik przesyła do partnera znane sobie hasło oraz własną maskę, określającą grupy, do których należy. Obydwa atrybuty są zapamiętywane wśród parametrów połączenia i przy próbie dostępu (poprzez to połączenie) porównane z atrybutami obiektu. Zgodność hasła lub przynależność do jednej z grup użytkowników uprawnionych, decydują o przyznaniu lub nie przyznaniu klientowi prawa dostępu do żadanego obiektu.

Mechanizm kontroli dostępu jest opcjonalny. Sprawdzanie uprawnień klientów następuje tylko wtedy, gdy znacznik *Ochrona dostępu obecna* ma wartość *true*.