

Michał SPENDEL

PWPT WASKO Sp. z o.o., Zakład Systemów Telekomunikacyjnych

MODELOWANIE APLIKACJI ZARZĄDCY W OPARCIU O INFORMACJĘ ZARZĄDZANIA W SIECIACH TMN

Streszczenie. Artykuł podejmuje próbę wyjaśnienia metodologii przygotowywania aplikacji zarządzających zasobami sieci telekomunikacyjnych. Jako odniesienie przyjmuje standardy zarządzania zdefiniowane przez ITU-T. Stosując metodologię przyjętą w niniejszym opracowaniu, można integrować heterogeniczne zasoby sieciowe z aplikacją specjalizowanego zarządcy.

MODELLING OF MANAGER APPLICATION IN TMN NETWORKS

Summary. The article explains the methodology of preparing the manager applications in TMN networks. Relying upon ITU-T standards it gives a conception of how to integrate the heterogenous network resources with the application of the specialized manager.

1. Wprowadzenie

Modelowanie aplikacji specjalizowanego zarządcy w sieciach TMN jest procesem wymagającym znajomości zagadnień opisanych w standardach ITU-T. Kolejne rozdziały przybliżą najważniejsze postulaty wypływające z tych standardów i posłużą do przygotowania modelu informacyjnego zarządcy.

1.1. Czym jest sieć TMN?

TMN (*ang. Telecommunication Management Network*) to model zarządzania siecią telekomunikacyjną. Standardy ISO określają sposoby zarządzania i współpracy systemów otwartych OSI (*ang. Open Systems Interconnection*). Pojęcie sieci nie jest w tych standardach

bezpośrednio używane - jedynie można zakładać, że systemy otwarte tworzą sieć. ITU-T, której prace koncentrują się na zarządzaniu siecią telekomunikacyjną, przyjęła od ISO model zarządzania OSI i ustanowiła serię zaleceń, które w 1988 roku przyjęły nazwę TMN. Celem opracowania zaleceń TMN było zdefiniowanie architektury zarządzania, która pozwala za pośrednictwem sieci transmisji danych DCN (*ang. Data Communication Network*) połączyć systemy zarządzania OS (*ang. Operating System*) z elementami sieci NE (*ang. Network Elements*). Elementy sieci NE mają z kolei dostęp do fizycznych zasobów sieci. Zalecenia określają również zasady współpracy systemów zarządzania OS między sobą.

Funkcje, jakie ma zapewniać sieć TMN [1]:

- wymiana informacji zarządzania między siecią telekomunikacyjną a siecią zarządzania TMN,
- przesyłanie informacji zarządzania między komponentami sieci TMN,
- konwersję formatu informacji zarządzania między komponentami sieci TMN,
- konwersję formatu informacji zarządzania przesyłanej wewnątrz sieci TMN tak, aby były zrozumiałe w środowisku heterogenicznym,
- analizę uzyskanej informacji zarządzania i reagowania na nią (przetwarzanie informacji zarządzania),
- dostarczanie informacji zarządzania do jej użytkownika,
- zapewnienie ochrony dostępu do informacji zarządzania.

Czym jest informacja zarządzania?

Informacja zarządzania jest umownym sposobem opisu zasobów infrastruktury telekomunikacyjnej w postaci zrozumiałej dla heterogenicznych systemów. Językiem przyjętym do opisu wspomnianych zasobów jest GDMO (*ang. Guidelines for Definition of Managed Object*) [6], a typy danych, jakimi się on posługuje, są prezentowane za pomocą dokumentów ASN.1 (*ang. Abstract Syntax Notation One*).

1.2. Modelowanie informacji zarządzania

Modelowanie informacji zarządzania sprowadza się zazwyczaj do procesu serwera (agenta), który ma dostęp do fizycznych zasobów w sieci. Agenci znajdują się najczęściej w tzw. elementach sieci NE, chociaż nie jest to regułą. Komunikują się z procesem zarządcy za pośrednictwem sieci DCN. Sieć DCN może być częścią sieci telekomunikacyjnej (np. SS7, ECC w SDH) lub zupełnie odrębną siecią mającą własną infrastrukturę.

Proces zarządcy umiejscowiony jest najczęściej w elemencie OS. Nie jest on w stanie zarządzać elementami sieci za pomocą agentów, jeśli nie przekaże się mu odpowiedniej

informacji zarządzania. Informacja zarządzania znana agentowi musi być przekazana zarządcy, by mógł wykonywać na niej operacje zarządzania [3][4].

2. Proces tworzenia aplikacji zarządcy

Aplikacja zarządcy TMN jest tworzona w celu integracji różnych zasobów w heterogenicznym środowisku telekomunikacyjnym, kiedy to jednemu systemowi należy powierzyć określone funkcje kontrolne nad innym systemem [2]. Integracja ma sens jedynie w przypadku znajomości modelu informacyjnego zasobu.

2.1. Opis algorytmu

Tworzenie aplikacji zarządcy można przedstawić za pomocą algorytmu zaprezentowanego na rysunku 1.

2.1.1. KROK 1. Określenie dokumentu wymagań

Czynność poprzedza właściwe przystąpienie do procesu tworzenia aplikacji.

Ideą tej czynności jest przygotowanie tzw. dokumentu wymagań, opisującego miejsce zarządcy w sieci zarządzania. Dokument powinien zawierać:

- rodzaje usług (zdarzeń) w komunikacji pomiędzy zarządcą i agentem,
- elementy sieciowe, którymi zarządca może zarządzać.

Rysunek 2 przedstawia obrazowo powstawanie dokumentu wymagań.

2.1.2. KROK 2. Modelowanie fizycznych urządzeń

Proces modelowania fizycznych zasobów infrastruktury telekomunikacyjnej polega na zdefiniowaniu obiektów zarządzanych MO (*ang. Managed Objects*), którymi posługuje się zarządca. MO są niczym innym, jak abstrakcyjną reprezentacją fizycznych zasobów. Obiekty MO składają się na bazę informacji zarządzania MIB (*ang. Management Information Base*). Każdy obiekt w bazie ma nadany unikalny w skali globalnej identyfikator.

Pojęcie obiekty zarządzane MO jest w pewnym sensie uproszczeniem. W rzeczywistości obiekt w bazie MIB jest wystąpieniem¹ klasy zarządzanego obiektu MOC (*ang. Management Object Class*). Obiekt w bazie opisywany jest szczegółowo przez definicję klasy, która została omówiona w rozdziale następnym.

¹ „Wystąpienie” klasy obiektu jest pojęciem ogólnie stosowanym w literaturze fachowej. Innymi słowy, „wystąpienie” to obiekt z nadaną nazwą umieszczony w bazie MIB.

2.1.2.1. Klasa zarządzanego obiektu MOC

Zalecenie ITU-T X.720 proponuje sposób definiowania obiektów, a tym samym sposób definiowania ich klas.

W definicji klasy MOC znajdują się następujące elementy:

- atrybuty określają statyczne cechy obiektu. Na definicję atrybutu składają się nazwa, typ i wartość. Typ atrybutu definiuje się poprzez podanie dziedziny jego wartości za pomocą notacji ASN.1 oraz operacji i działań, jakie mogą zostać przeprowadzone na wartościach przez zarządcę. Przy ustanawianiu obiektu (wystąpienia klasy) atrybutowi nadaje się wartość; wartości atrybutów stanowią element rozróżniający obiekty będące wystąpieniami tej samej klasy. Przykładem atrybutu może być atrybut o nazwie „data uruchomienia urządzenia”, typie „ciąg znaków” i wartości „27.05.1999”;
- operacje, akcje i meldunki są elementami definicji obiektu, które określają jego dynamiczne właściwości. Zgodnie z modelem zarządcy-agent [1], zarządca zleca agentowi wykonanie operacji dotyczących zarządzanych zasobów. Z kolei rola agenta polega na wykonywaniu zleconych operacji, a także na samodzielnym przesyłaniu zarządcy meldunków o zarządzanych zasobach. Definicja klasy zawiera z jednej strony dopuszczalne operacje i akcje, a z drugiej meldunki, których składnia jest definiowana za pomocą notacji ASN.1. Operacje można podzielić na te, które przeprowadza się na atrybutach obiektu i te, które dotyczą obiektu jako całości. Tych, które dotyczą atrybutu, jest pięć:
 - Get attribute value – odczytać wartość atrybutu,
 - Replace attribute value – zmienić wartość atrybutu,
 - Replace with default value – zamienić wartość atrybutu na wartość domyślną,
 - Add member – dodać wartość do zbioru wartości, jakie może przyjmować atrybut,
 - Remove member – usunąć wartość ze zbioru wartości, jakie może przyjmować atrybut.

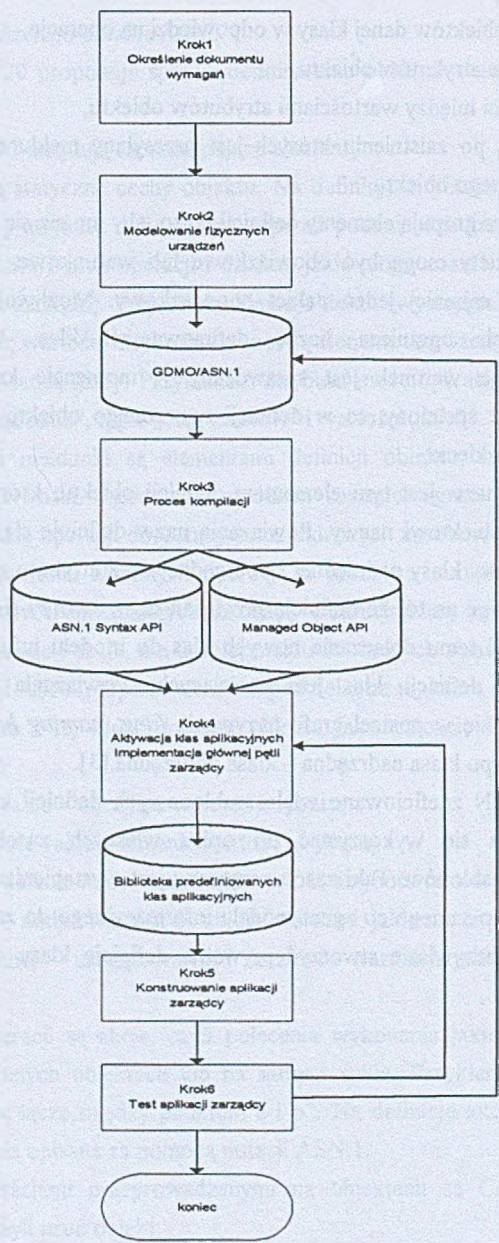
Innym typem operacji są akcje, czyli polecenia wykonania jakiegoś działania przez dany obiekt na innych obiektach lub na samym sobie. Przykładem akcji może być polecenie „zestaw łączy między punktem a i b”. Na definicję akcji składa się przede wszystkim składnia opisana za pomocą notacji ASN.1.

Pozostałymi operacjami przeprowadzanymi na obiektach są *Create*, czyli utwórz obiekt i *Delete*, czyli usuń obiekt;

- zachowanie, które jest najmniej rygorystycznie zdefiniowanym elementem definicji klasy obiektu. Określa ono przede wszystkim semantykę (znaczenie) obiektów będących wystąpieniami danej klasy, a także związki tych obiektów z obiektami innych klas. Zachowanie może również określać:

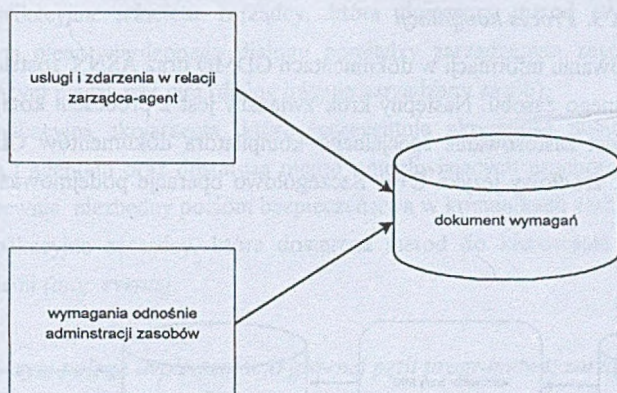
- reakcje obiektów danej klasy w odpowiedzi na operacje,
 - znaczenie atrybutów obiektu,
 - zależności między wartościami atrybutów obiektu,
 - warunki, po zaistnieniu których jest przesyłany meldunek określony w definicji klasy danego obiektu;
- pakiety, które grupują elementy definicji po to, aby można się było odwoływać do nich łącznie. Pakiety mogą być obowiązkowe lub warunkowe, przy czym każda klasa zawiera co najmniej jeden pakiet obowiązkowy. Możliwość stosowania pakietów warunkowych ogranicza liczbę definiowanych klas. W przypadku pakietu warunkowego warunek jest sprawdzany w momencie kreacji wystąpienia. Jeśli warunek jest spełniony, to w definicji tworzonego obiektu umieszcza się elementy zawarte w pakiecie;
- powiązania nazw jest tym elementem definicji obiektu, który wykorzystuje się przy nadawaniu obiektowi nazwy. Powiązania nazw definiuje się na poziomie klas przez podanie nazwy klasy nadrzędnej i podrzędnej oraz atrybutu tworzenia nazwy. Należy zwrócić uwagę na to, że definicję powiązań nazw tworzy się niezależnie od definicji klasy. Dzięki temu dołączanie nowych klas do modelu informacyjnego nie wymaga modyfikacji definicji klas już istniejących. Powiązania nazw danego modelu przedstawia się w postaci grafu nazywania (*ang. naming hierarchy*), który ilustruje zależności typu klasa nadrzędna – klasa podrzędna [3].

W standardach TMN zdefiniowano wiele szablonowych definicji klas, a nawet całych baz MIB, które zaleca się wykorzystać do opisu własnych zasobów w przypadku ich podobieństwa do szablonów. Taki zasób nazywany jest wystąpieniem klasy predefiniowanej. Zarządca używa tego samego co agent modelu informacyjnego do zarządzania interakcjami z zasobami. Można oczywiście stworzyć prywatną definicję klasy specjalnie utworzoną na własne potrzeby.



Rys. 1. Algorytm tworzenia aplikacji zarządcy

Fig. 1. Algorithm of creation the management application

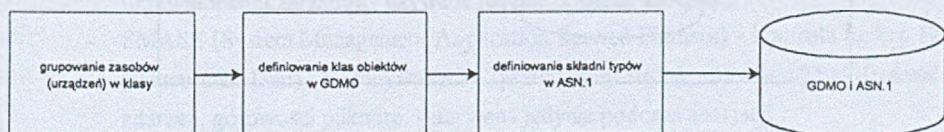


Rys. 2.Tworzenie dokumentu wymagań

Fig. 2. Developing a requirements document

Rysunek 3 przedstawia w uproszczeniu tworzenie własnej bazy MIB, która nazwana jest bazą modelującą. Powiązane dokumenty GDMO/ASN.1 tworzą bazę zarządzania MIB.

Po określeniu klasy MOC odpowiedniej do danego zasobu należy zastosować jej dokument GDMO, który jest formalnym opisem zasobu.



Rys. 3.Tworzenie bazy MIB

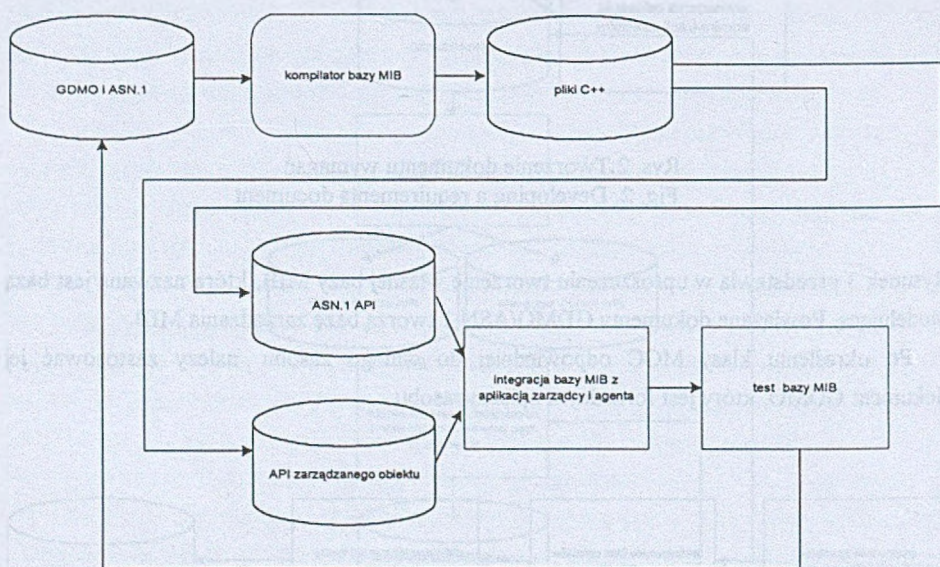
Fig. 3. MIB modeling

Dokument GDMO opisuje zasób, ale nie posiada informacji na temat typów zmiennych, którymi się posługuje.

Z modelem zasobu powinien być również skojarzony dokument ASN.1, w którym znajduje się informacja o używanych typach zmiennych. Dopiero tak przygotowany model może być wykorzystany w dalszych etapach tworzenia aplikacji zarządzania.

2.1.3. KROK 3. Proces kompilacji

Po skompletowaniu informacji w dokumentach GDMO oraz ASN.1 została przygotowana baza MIB dla danego zasobu. Następny krok związany jest z procesem kompilacji. Do tego celu niezbędne jest zastosowanie specjalnego kompilatora dokumentów GDMO, który na wyjściu daje plik źródłowy języka C++. Szczegółowo operacje podejmowane w tym kroku oddaje rysunek 4.



Rys. 4. Uruchomienie kompilatora i test bazy MIB

Fig. 4. Running the Object Compiler and testing the MIB

2.1.4. KROK 4. Aktywacja klas aplikacyjnych i implementacja głównej pętli programowej

W tym kroku implementowana jest główna pętla programowa zarządzcy, która rozpoczyna i kontroluje zarządzane procesy. Inicjowane są również następujące klasy aplikacyjne:

- klasa aplikacyjna zarządzanego zasobu, która dostarcza metod wywoływanych w sytuacjach potwierdzonego dialogu pomiędzy zarządzanym zasobem a agentem sprawującym pieczę nad nim (dialog inicjuje zarządzany zasób),

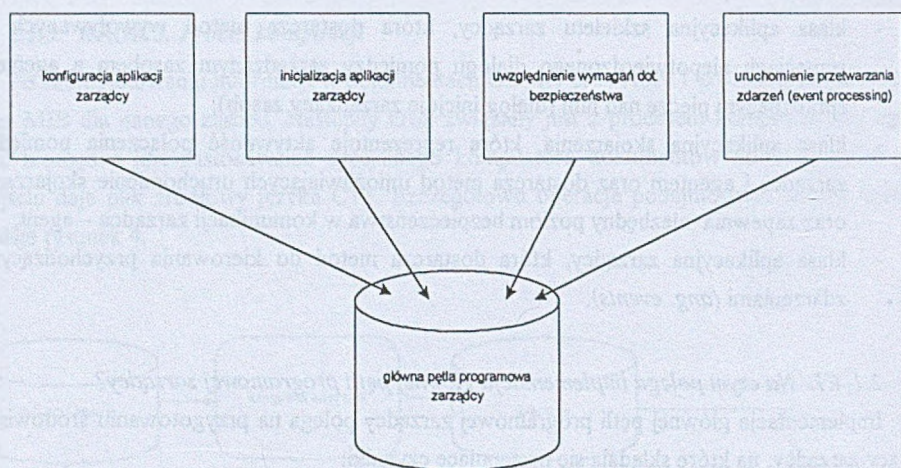
- klasa aplikacyjna szkieletu zarządcy, która dostarcza metod wywoływanych w sytuacjach niepotwierdzonego dialogu pomiędzy zarządzanym zasobem a agentem sprawującym pieczę nad nim (dialog inicjuje zarządzany zasób),
- klasa aplikacyjna skojarzenia, która reprezentuje aktywność połączenia pomiędzy zarządcą i agentem oraz dostarcza metod umożliwiających uruchomienie skojarzenia oraz zapewnia niezbędny poziom bezpieczeństwa w komunikacji zarządcą – agent,
- klasa aplikacyjna zarządcy, która dostarcza metod do kierowania przychodzącymi zdarzeniami (*ang. events*).

2.1.4.1. Na czym polega implementacja głównej pętli programowej zarządcy?

Implementacja głównej pętli programowej zarządcy polega na przygotowaniu środowiska pracy zarządcy, na które składają się następujące czynniki:

- ustawienie stałych określających m.in. maksymalną liczbę połączeń utrzymywanych przez zarządcę z agentem, liczbę operacji w ramach jednego połączenia itp.,
- określenie poziomu bezpieczeństwa podczas wymiany informacji między zarządcą i agentem. Istnieją trzy mechanizmy zapewniające bezpieczną wymianę informacji:
 - Access Control - kontrola identyfikatora agenta oraz jego praw do wykonania zamierzonych działań - używana podczas nawiązywania połączenia (asocjacja) oraz podczas wykonywania operacji na zasobach zarządzanych,
 - User Information - kontrola specyficznych danych użytkownika znanych tylko uprawnionemu zarządcy - używana jedynie podczas asocjacji,
 - SMASE (System Management Application Service Element) - kontrola funkcji FU (Functional Unit) wykonywanych przez obie strony, np. kontrola dziennika zdarzeń, gotowości pliku itp. - używana jedynie podczas asocjacji,
- wbudowanie modułu przetwarzającego przychodzące zdarzenia z sieci - każdy przychodzący raport o zdarzeniu w sieci powinien być skojarzony w module przetwarzającym z funkcją tzw. callback, dającą informację zwrotną o fizycznym zasobie. Podobnie rzecz się ma w przypadku tzw. timerów, czyli zdarzeń zachodzących lokalnie na platformie zarządcy po zakończeniu ustalonego odcinka czasowego.

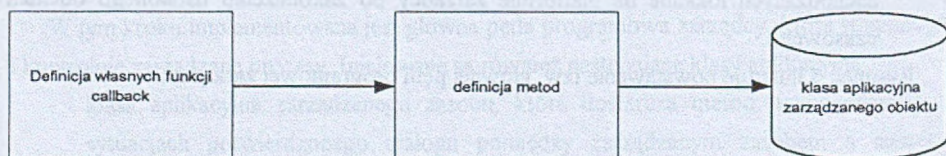
Rysunek 5 ilustruje powstawanie tzw. głównej pętli programowej zarządcy.



Rys. 5. Implementacja głównej pętli programowej zarządcy
 Fig. 5. Implementing the manager main loop

2.1.4.2. Uzyskiwanie klas aplikacyjnych

Uzyskiwanie klasy aplikacyjnej zarządzanego obiektu. Klasa aplikacyjna zarządzanego obiektu dostarcza metod wykonywanych na zarządzanych zasobach i dlatego niezbędne jest zagwarantowanie aktywności agenta, który będzie dawał odpowiedzi na zapytania kierowane od zarządcy. Proces definiowania klasy aplikacyjnej zarządzanego zasobu polega na określeniu zbioru własnych funkcji callback i skojarzeniu ich z metodami. Metody są elementem składowym definicji klasy aplikacyjnej zarządzanego obiektu. Są elementem biblioteki API (*ang. Application Program Interface*) C++. Z biblioteki API należy także wybrać własną klasę aplikacyjną spośród predefiniowanych i wprowadzić do niej metody. Rysunek 6 obrazowo przedstawia kolejność działań przy ustalaniu klasy aplikacyjnej.



Rys. 6. Uzyskiwanie klasy aplikacyjnej zarządzanego obiektu
 Fig. 6. Deriving the application managed object class

Uzyskiwanie klasy aplikacyjnej szkieletu zarządzcy. Klasa aplikacyjna szkieletu dostarcza metod, które są wywoływane w przypadku braku jakiegokolwiek odpowiedzi ze strony zasobu będącego pod pieczęcią agenta. Istotne jest, że sam agent funkcjonuje i zwraca odpowiedzi zarządzcy. Proces ten polega na wybraniu jednej z predefiniowanych klas aplikacyjnych szkieletu zarządzcy i połączeniu jej z własnymi funkcjami callbacks poprzez metody - tak jak w przykładzie poprzednim.

Uzyskiwanie klasy aplikacyjnej skojarzenia. Ta klasa dostarcza metod do zarządzania połączeniem pomiędzy aplikacjami zarządzcy i agenta. Podobnie jak w dwu poprzednich przypadkach klasę aplikacyjną skojarzenia wybiera się spośród klas predefiniowanych w API. Przed wyborem klasy należy określić, jakie wymagania odnośnie bezpieczeństwa powinna spełniać sesja pomiędzy zarządzcą i agentem, po czym należy określić wymagania SMASE, User Information oraz Access Control. Przygotowanie tych kryteriów uprawnia do zdefiniowania metod klasy aplikacyjnej skojarzenia. Tak przygotowana klasa pozwala wyłuskiwać argumenty z informacji przychodzącej od agenta i odpowiedzieć na nie w sposób autoryzowany.

Uzyskiwanie klasy aplikacyjnej zarządzcy. Z pomocą tej predefiniowanej klasy otrzymujemy metody do przetwarzania i odpowiadania na zdarzenia przychodzące od agentów.

2.1.5. KROK 5. Konstruowanie aplikacji zarządzcy

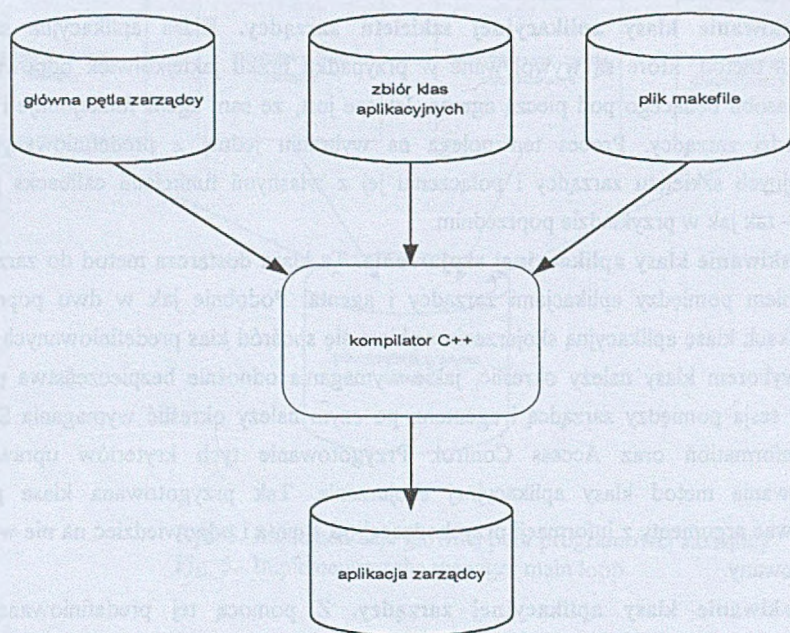
Rysunek 7 w skrócie przedstawia proces konstruowania aplikacji zarządzcy pod warunkiem poprawnego przygotowania klas z kroku poprzedniego.

Właściwe przygotowanie programu makefile poprzedza kompilację, która jest głównym procesem realizowanym na tym etapie.

Makefile jest programem wywołującym kompilator i ustawiającym środowisko kompilacji.

2.1.6. KROK 6. Testowanie aplikacji zarządzcy

Ostatnim etapem w procesie tworzenia własnej aplikacji zarządzania jest procedura testowania. Test polega na uruchomieniu dwóch aplikacji: zarządzającej oraz agenta lub specjalnej aplikacji testującej podszywającej się pod prawdziwego agenta (tzw. symulator)[5]. Należy kolejno dokładnie sprawdzić wszystkie operacje zarządzcy (od ustanowienia połączenia poprzez akcje wykonywane na zarządzanych zasobach). Należy tak ustawić agenta, by wygenerował wszystkie komunikaty, jakie zapisane są w bazie MIB, po czym należy obserwować zachowanie zarządzcy. Operacje zarządzcy powinny dotyczyć wszystkich obiektów w bazie MIB.



Rys. 7. Konstruowanie aplikacji zarządcy

Fig. 7. Building the management application

Program zarządcy powinien być wywołany z opcjami pozwalającymi na wykonanie dokładnych trace'ów. Analiza trace'ów w wielu przypadkach pozwala zidentyfikować i wyeliminować błędy. Jeśli aplikacja zarządcy nie działa w sposób zamierzony, należy wrócić albo do kroku 4, jeśli przyczyna błędów leży w źle zastosowanej klasie aplikacyjnej, albo do kroku 2, jeśli zastosowano zły model informacyjny i powtórzyć kolejne kroki algorytmu.

3. Podsumowanie

Standardy ISO dokładnie opisują przebieg komunikacji na poszczególnych warstwach stosu OSI. Programista modelując aplikację zarządcy korzysta z narzędzi, które przysyłają złożoność niższych warstw. Zgodność aplikacji zarządcy z zaleceniem X.720 omówionym na wstępie artykułu w zasadniczej mierze zależy od poprawności przygotowanego dokumentu GDMO/ASN.1. Zadaniem specjalizowanego API jest dostarczenie elastycznej funkcjonalności z zachowaniem zgodności ze standardami. Dokument opracowano wg dokumentacji firmowej do produktu TMN MDE v.3.2 firmy Vertel. Odwołania do predefiniowanych klas

aplikacyjnych muszą być poprzedzone wcześniejszym zainstalowaniem odpowiednich bibliotek programowych.

Idea projektowania aplikacji zarządzających jest zawsze taka sama. Mogą ulegać zmianie jedynie środki służące do realizacji zamierzenia. Istnieje kilku producentów oprogramowania do zarządzania w sieciach telekomunikacyjnych i każdy dostarcza swój zbiór bibliotek programowych. Należy zatem wykorzystać najodpowiedniejszy pakiet oprogramowania i postępować wg niniejszego algorytmu.

LITERATURA

1. Czarnecki P., Jajszczyk A., Lubacz J.: Standardy zarządzania sieciami OSI/NM, TMN. Wydawnictwo EFP, Poznań, 1995.
2. TMN C++ Manager Application Developer's Guide, 1999.
3. Czarnecki P., Jajszczyk A.: Modelowanie informacji zarządzania, czyli jak systemy zarządzania widzą zarządzane zasoby. Przegląd Telekomunikacyjny nr 10/1995.
4. Czarnecki P., Jajszczyk A.: Zastosowanie notacji GDMO do tworzenia modeli informacyjnych zarządzania sieci. Przegląd Telekomunikacyjny nr 3/1996.
5. TMNTelecore Agent Simulator User Guide, 1999.
6. Solstice TMN Agent Developer's Handbook, 1998.

Recenzent: Dr inż. Rafał Cupek

Wpłynęło do Redakcji 4 kwietnia 2000 r.

Abstract

This article explains the methodology of creating the management application in TMN (Telecommunication Management Network) environment. It relies upon ITU-T standards concerning the management model at telecommunication network. It introduces to creating the management information and uses it to modeling the application. The most important item in this article is presentation of the preceedings algorithm (Fig. 1) during creation own application. The next sections give the explanation of the individual steps of the algorithm. Conformity between ITU-T standards and application is guaranteed with the specialist software. The article gives the universal method it can be used with various software library.