Software nr 3, marzec 1998.
Zygmunt BOK
Zakłady Tworzyw Sztucznych 'NITRON' S.A.
ul. Zawadzkiego 1, 42-693 Krupski Młyn

ANALIZA MODELU LOGICZNEGO OBIEKTOWO ZORIENTOWANEJ BAZY NDS W SIECIOWYM SYSTEMIE OPERACYJNYM NOVELL NETWARE 4.1

Streszczenie. W tym artykule przedstawiono logiczny model obiektowo zorientowanej implementacji bazy informacyjnej zwanej Usługi Katalogowe NetWare (NDS), przechowującej zróżnicowane typy informacji o wszystkich obiektach znajdujących się w sieci. Przedstawiono również architekturę oraz schemat NDS'u wraz z mechanizmem dziedziczenia własności.

THE LOGICAL MODEL ANALYSIS OF THE OBJECT ORIENTED DATABASE NDS IN NOVELL NETWARE 4.1 NETWORK OPERATING SYSTEM

Summary. In this article a logical model of the object oriented implementation of the information database called NetWare Directory Services (NDS) for storage different information types about all objects in the network has been presented. Futhemore an architecture and schema of the NDS with properties heritage machanism has been presented also.

1. W P R O W A D Z E N I E

Wraz z wprowadzeniem przez firmę Novell na rynek w marcu 1993r. następnej generacji sieciowego systemu operacyjnego NetWare 4.0, wprowadzono również nowe pojęcie [1], tzw. - Usługi Katalogowe NetWare (ang. NetWare Directory Services - NDS). Przez to pojecie rozumie się [2] organizację całej sieci przedstawionej hierarchicznie, zgodnie z założeniami organizacji lub korporacji, dla której została zaimplementowana.

Usługi Katalogowe (ang. Directory Services) stanowią pewną bazę informacyjną używającą zróżnicowanych typów informacji o użytkownikach oraz jego zasobach w sieciowym środowisku obliczeniowym, natomiast NDS jest obiektowo zorientowaną implementacją tej bazy, przechowującą informacje o wszystkich obiektach znajdujących się w sieci.

Architektura NDS pozwala na globalny dostęp do wszystkich zasobów sieciowych, niezależnie od ich fizycznej lokalizacji, tworząc w ten sposób jednolity system informacyjny.

W poprzednich wersjach sieciowych systemów operacyjnych NetWare, do przechowywania informacji o pojedynczym serwerze sieciowym używano płaskiej bazy danych zwanej *bindery*, charakteryzującej się tym, że poszczególne pozycje w tej bazie nie mają bezpośrednich wzajemnych związków z innymi pozycjami.

W przeciwieństwie do bazy *bindery*, NDS zorganizowano w strukturę hierarchicznego drzewa w którym widać związki pomiędzy obiektami, która stanowi globalną, rozproszoną oraz replikowalną obiektową bazę danych przechowującą informacje o wszystkich zasobach sieciowych takich jak: użytkownicy, grupy, serwery, woluminy, drukarki, komputery, modemy itd. Przez pojecie rozproszonej bazy danych w tym opracowaniu rozumie się zbiór węzłów (serwerów) połączonych siecią komunikacyjną na których zainstalowano lokalne systemy baz danych.
Za pomocą NDS możliwe staje się zintegrowanie różnych zasobów sieciowych w jednolite i zarazem proste w wykorzystaniu środowisko sieciowe.

Dla użytkowników sieci komputerowej korzyści jakie płyną ze stosowania NDS jest wiele. Począwszy od prostego aktu logowania się do całej sieci komputerowej (a nie tylko do pojedyńczego serwera sieciowego[3]) umożliwiającego łatwiejszą nawigację po zasobach sieciowych i ich wykorzystanie przez autoryzowanych użytkowników, poprzez szerokie korzystanie z dostępnych usług sieciowych bez konieczności znajomości topologii sieci (NDS jest niezależna od platformy topologicznej [2]), protokołów, mediów transmisyjnych i połączeń komunikacyjnych, aż po możliwości wykorzystania bardzo zaawansowanych metod kryptograficznych podnoszących na wyższy poziom kwestie szeroko rozumianego bezpieczeństwa sieciowego.

2. OBIEKTOWA BAZA DANYCH NDS - WSTĘP

2.1. LOKALIZACJA OBIEKTOWEJ BAZY NDS

Jak już wspomniano, NDS jest rozproszoną i replikowalną obiektową bazą danych. W sieciach złożonych z kilku serwerów, partycja główna lub główna część logiczna bazy NDS rezyduje na pewnym wyróżnionym serwerze sieciowym, natomiast pozostałe jej partycje znajdują się na innych serwerach sieciowych.
W celu zapewnienia bezpieczeństwa i niezawodności (ang. fault tolerance), wszystkie partycje są replikowane i składowane na sąsiednich.serwerach włączonych do sieci.
W rzeczywistości jednak [5], rozproszona i replikowana na innych serwerach sieciowych jest baza o nazwie *Directory Information Base* - DIB, która opisuje NDS i jej pliki, natomiast utrzymywana i zarządzana jest przez NDS.

W sytuacji kiedy jeden z serwerów sieciowych jest wyłączony, wówczas informacje o zasobach i usługach sieciowych przez niego oferowanych odczytywane są z replik partycji NDS'u, które zeskładowane są na sąsiednich serwerach sieciowych.

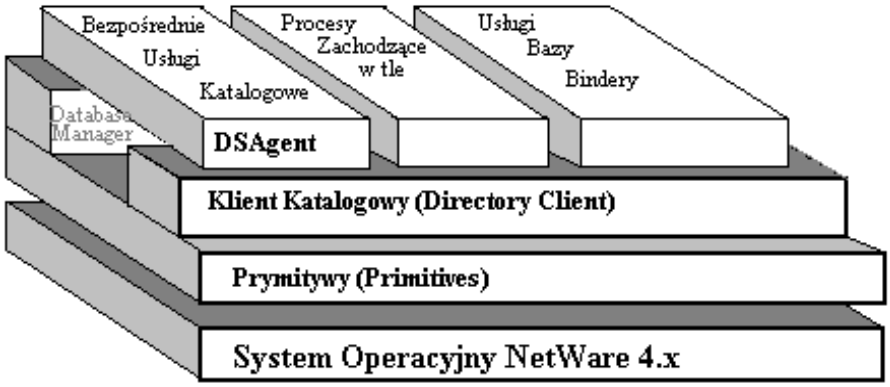
Miejsce składowania [4] plików całej bazy NDS lub poszczególnych jej partycji to katalog o nazwie *SYS:_NETWARE*, znajdujący się na każdym z serwerów sieciowych. Jest to katalog niedostępny (niewidzialny) dla użytkowników pracujących na stacjach roboczych.

Informacje o tym katalogu otrzymuje się poprzez uruchomienie programu *RCONSOLE.*
Następnie po wybraniu serwera docelowego i opcji *'Directory Scan'* oraz po podaniu ścieżki *SYS:_NETWARE*, uzyskuje się dostęp do tego katalogu.
W celu zapisania w odpowiednich plikach bazy NDS informacji o jej obiektach, NDS używa mechanizmu pod nazwą *Record Manager*, który zapewnia dostęp do bazy DIB przechowującej informacje o obiektach.

2.2 ARCHITEKTURA OBIEKTOWEJ BAZY NDS

W związku z tym, że NDS jest globalną, hierarchiczną, rozproszoną i replikowalną bazą danych, tworzy zatem złożony system bazodanowy, którego architekturę przedstawiono na rys. 1. Architektura NDS'u składa się z następujących podsystemów:

- Prymitywy (ang. Primitives)
- Klient Katalogowy (ang. Directory Client)
- Zarządca bazy danych (ang. Database Manager)
- Agent usług katalogowych (ang. DS Agent)
- Bezpośrednie Usługi Katalogowe (ang. Direct Directory Services)
- Procesy Tła (ang. Background Processes)
- Usługi bazy bindery (ang. Bindery Services)



Rys. 1 Zarys architektury NDS
Fig. 1 Layout of the NDS architecture

Prymitywy

W związku z tym, że Usługi Katalogowe Netware (NDS) mogą być dedykowane dla różnych systemów operacyjnych, muszą zatem korzystać z niektórych usług przez nie oferowanych. Ponieważ usługi te różnią się między sobą, wprowadzono zatem pewną warstwę pośrednią, ujednolicającą te usługi. Innymi słowy, warstwa prymitywów dokonuje translacji funkcji systemowych do postaci wymaganej przez NDS. Przykładowo, system operacyjny może obliczać czas w inny sposób niż to robi NDS, dlatego też warstwa prymitywów musi dokonać translacji czasu systemu operacyjnego na format NDS'u.

Klient Katalogowy

Warstwa Klienta Katalogowego pozwala serwerowi NDS działać tak jak klient i żądać usług od innego serwera NDS'u. Warstwa ta jest funkcjonalnie równoważna warstwie *Directory User Agent* (DUA) ze specyfikacji X.500.

Zarządca bazy danych

Moduł ten zapewnia między innymi następujące funkcje:
- dostarcza procedur wspierających schemat
- zawiera tzw. meta schemat (ang. meta schema), czyli zasady definiujące schemat.
- zapewnia dostęp do bazy danych NDS'u za pomocą *record manager'a*.

DSAgent

Obsługuje operacje NDS'u wykonywane na serwerze sieciowym. W przypadku gdy ta warstwa jest zamknięta, wówczas serwer nie ma dostępu do lokalnej bazy (partycji NDS'u) na nim zainstalowanej. Posiada natomiast dostęp do replik tej partycji zeskładowanych na innych serwerach sieciowych.

Bezpośrednie Usługi Katalogowe

Ten podsystem obsługuje żądania lokalnych wywołań systemowych (ang. local system calls) do agenta lokalnego. Wykonuje także lokalne operacje, jak np. tworzenie woluminu.

Procesy Tła

Zadaniem tego podsystemu jest obsługa funkcji, które wykonywane są automatycznie bez interwencji użytkownika. Należą do nich:

- synchronizacja replik partycji NDS'u
- synchronizacja schematu (ang. schema synchronization). Proces ten propaguje wszystkie zmiany jakie w nim dokonano na wszystkie repliki znajdujące się na innych serwerach sieciowych.

Usługi bazy bindery

Ten moduł NDS'u zapewnia wsteczną kompatybilność z serwerami NetWare działającymi w oparciu o bazę *bindery.*

2.3. BUDOWA OBIEKTOWEJ BAZY NDS

2.3.1. DRZEWO KATALOGOWE (ang. DIRECTORY TREE)

Baza danych NDS logicznie reprezentowana jest przez drzewiastą, hierarchiczną strukturę, zawierającą różne obiekty (pozycje). Każda pozycja w tym logicznym drzewie tj. w *Directory Information Tree* (DIT) odpowiada fizycznym pozycjom lub instancjom obiektów zawartych w *Directory Information Base* (DIB), która jest jej fizyczną reprezentacją.
NDS opracowano zatem w taki sposób [6], aby można było tworzyć hierarchiczną strukturę tzw. Drzewa Katalogowego, składającego się z jednostek organizacyjnych zawierających użytkowników i zasoby sieciowe, natomiast zasady definiujące konstrukcję drzewa katalogowego określono i zapisano w Opisie Bazy Katalogowej (ang. Directory Schema) lub inaczej - w schemacie.

2.3.1.1. WYMAGANIA KONSTRUKCYJNE DLA DRZEWA KATALOGOWEGO

Przy opracowywaniu NDS [6] przyjęto założenie aby jego konstrukcja była spójna z wymaganiami ustanowionymi w standardzie X.500.
Specyfikację tego standardu opracowano i ustalono przez IEEE (ang. Institute of Electrical and Electronic Engineers) w celu zapewnienia standardowych metod dla organizowania informacji oraz dla promowania rozwoju międzynarodowych usług katalogowych opartych o wielką liczbę agentów DSA (ang. Directory System Agents) połączonych w sieć OSI (ang. Open Systems Interconnection) używających zdefiniowanych w tym standardzie protokołów.

Przykładowo, informacje zawarte w katalogach telefonicznych, korporacyjnych strukturach organizacyjnych oraz katalogach różnych serwisów dostępne są za pomocą produktów kompatybilnych z tą specyfikacją.

3. OPIS BAZY KATALOGOWEJ NDS - SCHEMAT

3.1. OPIS BAZY KATALOGOWEJ (ang. DIRECTORY SCHEMA)

Opis Bazy Katalogowej (schemat) kontroluje strukturę drzewa katalogowego poprzez określenie zasad kontrolujących jego elementy składowe tj.:

* klasy obiektowe
* atrybuty (mogące być wykorzystywane przez różne klasy obiektowe)
* typy wartości

Schemat określa specyficzne typy informacji oraz sposób w jaki są one konstruowane i zapisywane w bazie katalogowej (ang. Directory database). Innymi słowy, schemat definiuje jaki obiekt NDS'u może istnieć, jakie atrybuty może zawierać, jakie wartości atrybutu są dozwolone oraz gdzie może istnieć w Drzewie Katalogowym NDS'u . Ponieważ struktura drzewa katalogowego zależy od schematu, dlatego też musi być składowany na każdym serwerze wchodzącym w skład drzewa katalogowego oraz synchronizowany w celu zapewnienia jego spójności.

Modyfikacji podlegają tylko definicje klas obiektowych i atrybutów. Definicje syntaktyczne lub typy składniowe zapisywane są w sposób trwały w drzewie katalogowym NDS'u

Tak więc w schemacie zdefiniowano strukturę poszczególnych obiektów, ich wzajemne relacje w drzewie katalogowym jak również:

- atrybuty (ang. attribute) - opisujące jakiego typu dodatkowe informacje mogą być stowarzyszone z obiektem NDS'u.
- dziedziczność (ang. inheritance) - określająca, który obiekt NDS'u dziedziczy własności (ang. property) oraz prawa (ang. rights) innego obiektu.
- nazewnictwo (ang. naming) - identyfikuje i pokazuje nazwę obiektu w Drzewie Katalogowym
- podrzędność (ang. subordination) - określającą pozycję obiektu NDS'u w Drzewie Katalogowym

Schemat, który dostarczany jest wraz z systemem operacyjnym NetWare 4.x zwany jest schematem podstawowym

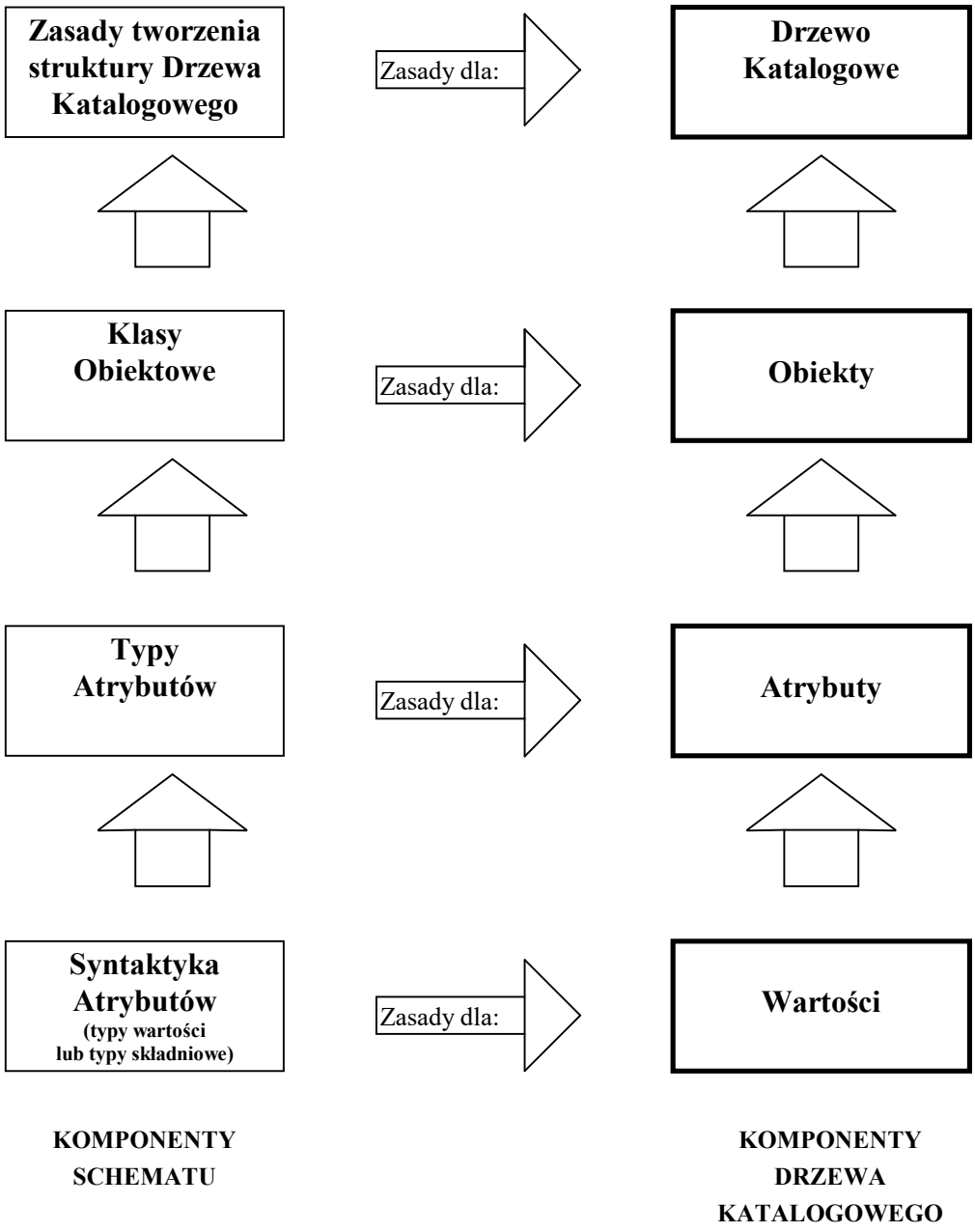
3.2. SCHEMAT PODSTAWOWY (ang. BASE SCHEMA)

Podstawą do tworzenia wszystkich pozycji w obiektowej bazie NDS jest zbiór obiektów (komponentów) tworzących klasę obiektową, który nazwany został schematem podstawowym (ang. base schema). Składa się definicji klas obiektowych i standardowych atrybutów. Wśród definicji zawartych w schemacie wyróżnić można trzy kategorie:

1. definicje syntaktyczne (ang. Syntax definition)
2. definicje atrybutów (ang. Attribute definition)

3. definicje klas obiektowych (ang. Class definition)

Poniżej, na rys. 2. przedstawiono wzajemne zależności pomiędzy komponentami wchodzącymi w skład schematu oraz drzewa katalogowego.



Rys. 2. Wzajemne zależności pomiędzy komponentami wchodzącymi w skład schematu oraz drzewa katalogowego.

Fig. 2. One-to-one relationalship between schema and directory tree components.

3.2.1. DEFINICJE SYNTAKTYCZNE (ang. Syntax definitions)

Definicje syntaktyczne definiują typy wartości opisujące naturę danych, które w dalszej kolejności mogą być przypisywane definiowanym atrybutom różnych klas obiektowych w celu zapamiętywania rzeczywistych danych. Definicje syntaktyczne są jedynym predefiniowanym i nie podlegającym rozszerzeniu oraz modyfikacji komponentem schematu NDS w którym predefiniowano 28 typów wartości (lub typów

składniowych), które mogą być przypisywane atrybutom instancji różnych typów klas obiektowych.
W przypadku gdy żaden z 28 predefiniowanych typów wartości nie jest odpowiedni do utworzenia definicji nowego atrybutu, wówczas programista może użyć następujących typów składniowych, tj. o*ctet string* lub *stream* w celu zapamiętywania rzeczywistych danych w formacie zdefiniowanym przez jego aplikację.

Predefiniowane typy wartości obejmują:
- typy łańcuchowe
- typy numeryczne (integer)
- typy logiczne (boolean)
- typy specyficzne dla NDS'u

Każdy z predefiniowanych typów wartości dostępny jest za pomocą odpowiedniej liczby z zakresu od 0 do 27, które zapisano w pliku NWDSDEFS.H, natomiast do reprezentacji tych typów wartości używa się struktur danych, które zdefiniowano i zapamiętano w pliku NDWSATTR.H.

Dla przykładu, typ wartości dostępny pod nazwą SYN_CI_STRING używa wskaźnika do znaku (ang. character pointer), natomiast typ wartości dostępny pod nazwą SYN_INTEGER używa 32-bitowej liczby ze znakiem typu integer.

3.2.2. DEFINICJE TYPÓW ATRYBUTÓW

Definicje typów atrybutów kojarzą ich nazwy z odpowiednimi definicjami syntaktycznymi. Każda definicja pewnego typu atrybutu tworzona jest na podstawie jednego z 28 predefiniowanych typów syntaktycznych oraz jednego lub więcej ograniczeń (ang. attribute constraints) w celu umożliwienia zapamiętania za jego pomocą pewnej kategorii informacji w obiekcie NDS'u. W schemacie podstawowym NDS'u zdefiniowano 142 różnych typów atrybutów. Przykładowo, typy atrybutów o nazwach *Surname, Full Name oraz Telephone Numbe*r mogą być użyte w definicji właściwych atrybutów (pól) pewnej klasy obiektowej *Person* w celu umożliwienia zapamiętania specyficznych informacji jej dotyczących.

W przeciwieństwie do klas obiektowych definiowanych np. w języku C++, gdzie dwie klasy mogą mieć pewne atrybuty o tej samej nazwie lecz o różnych typach wartości - definicje typów atrybutów w NDS'ie są niezależne od definicji atrybutów w klasach obiektowych, tzn. atrybuty o pewnej nazwie zdefiniowane w różnych klasach obiektowych zapamiętują dane o tym samym typie wartości i wykorzystywane są do definiowania dowolnych klas obiektowych.

W celu prawidłowego zdefiniowania pewnego nowego typu atrybutu należy przypisać mu także określone ograniczenia. W schemacie podstawowym NDS'u zdefiniowano 11 ograniczeniczeń atrybutów lub flag atrybutów. Przykładowo jeśli podczas definiowania pewnego typu atrybutu użyto flagi o nazwie *DS_READ_ONLY_ATTR*, wówczas użytkownik nie może go zmodyfikować, natomiast flaga o nazwie DS_NONREMOVABLE_ATTR uniemożliwia usunięcie go z definicji klasy obiektowej.

Atrybuty posiadają także stowarzyszone z nimi własności (ang. properties), które kontrolują sposób w jaki mogą być wykorzystane. Własności te określają, czy:

- atrybut może przyjmować wartości wielokrotne oraz w jakim zakresie
- atrybut powinien być np. tylko do czytania (ang. Readable) oraz w jaki sposób atrybut ma być synchronizowany.

Własności typów atrybutów ustalane są podczas ich definiowania i nie mogą być modyfikowane bez uprzedniego ich skasowania.
Definicja typu atrybutu może być skasowana tylko w tym przypadku, jeśli nie jest wykorzystywana w definicjach innych klas obiektowych.

3.2.3. KLASY OBIEKTOWE

Drzewo katalogowe tworzone jest z obiektów. Zbiór zasad [8] kontrolujących tworzenie konkretnego typu obiektowego zwany jest klasą obiektową (ang. object class), która według [10] jest opisem obiektu lub obiektów z jednolitym predefiniowanym zbiorem atrybutów i usług zawierający opis tworzenia nowych obiektów w klasie.

Każda klasa NDS'u składa się komponentów definiujących typy obiektów mogących pojawić się w drzewie katalogowym i posiadających zdefiniowane własne atrybuty (własności) oraz usługi (metody).

Każdy obiekt w drzewie katalogowym NDS'u należy do pewnej klasy obiektowej, która określa atrybuty i usługi jakie mogą być stowarzyszone z tym obiektem.
Wszystkie atrybuty obiektu oparte są o pewien zbiór wcześniej zdefiniowanych typów atrybutów, które z kolei definiowane są w oparciu o predefiniowane typy wartości.

Na definicję klasy obiektowej NDS'u składają się cztery następujące składniki:

- Reguły strukturalne (ang. Structure Rules)
- Super klasy (ang. Super Classes)
- Atrybuty obligatoryjne
- Atrybuty opcjonalne

Reguły strukturalne - Reguły strukturalne dla klas obiektowych definiują możliwe związki strukturalne pomiędzy obiektami w Drzewie Katalogowym (Directory Tree) - w szczególności atrybut każdego obiektu o nazwie ***Named By*** definiujący komponent nazwy tego obiektu, który dla większości obiektów typu „liść" (ang. leaf-node) przyjmuje wartość „CN" oraz atrybut o nazwie ***Containment*** dotyczący jego położenia w stosunku do innych obiektów (tj. w jakim kontenerze może rezydować), wykorzystywane są do zdefiniowania potencjalnych związków pomiędzy nimi.

Super klasa - Jest atrybutem służącym do określania struktury klas obiektowych w schemacie NDS'u. W szczególności określa reguły dziedziczenia, tj. określa podstawową klasę obiektową z której nowa klasa obiektowa jest wyprowadzana. Za jego pomocą następuje stowarzyszenie definicji klasy definiowanej z definicją nadklasy w wyniku czego nowa klasa obiektowa dziedziczy wszystkie własności .

Kompletna definicja każdej klasy obiektowej składa się z komponentów tej klasy oraz komponentów wszystkich nadklas znajdujących się (zapisanych explicite) na liście atrybutu super klasy.
Klasy obiektowe znajdujące się na górze hierarchii zapewniają bardziej ogólną charakterystykę, szczegółową natomiast zapewniają klasy obiektowe znajdujące się na jej dole. Super klasy służą również do wyprowadzenia kompletnego zbioru reguł dziedziczenia dla konkretnej klasy obiektowej.

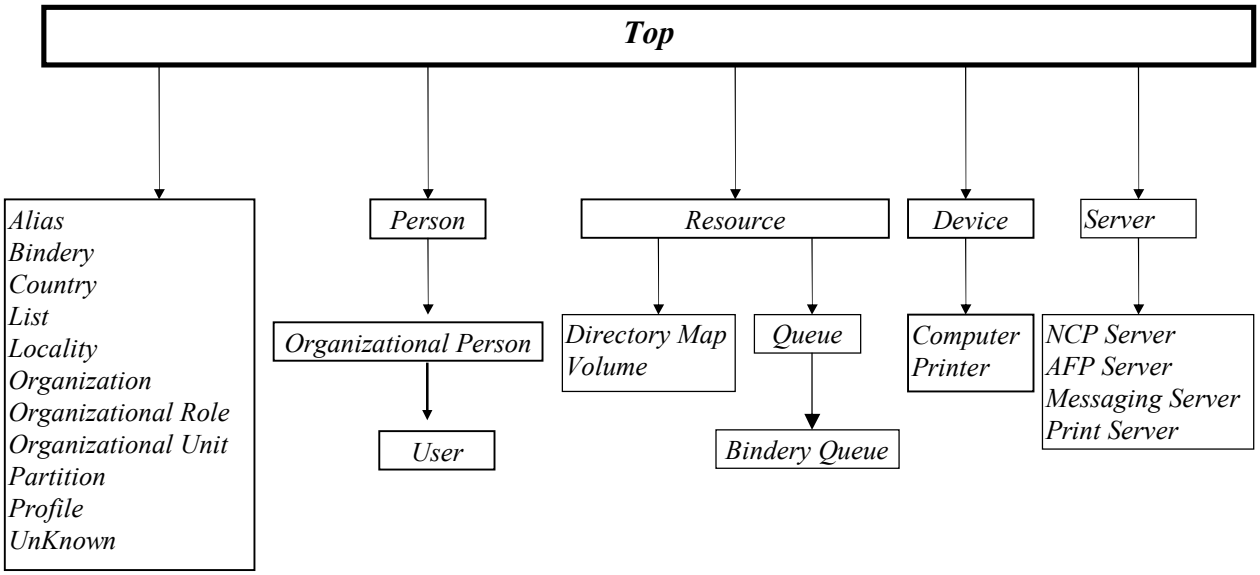
Atrybuty obligatoryjne - definiują te atrybuty których wartości muszą być określone przed utworzeniem instancji (obiektu) danej klasy obiektowej.

Atrybuty opcjonalne - definiują te atrybuty których wartości nie muszą być określone przed utworzeniem instancji (obiektu) danej klasy obiektowej. Ich określenie zwiększa użyteczność tego obiektu

3.2.3.1. HIERARCHIA KLAS

Poniżej na rys. 3. przedstawiono zarys klas obiektowych zdefiniowanych w schemacie podstawowym (*Base Schema*), pokazujący wszystkie klasy obiektowe na tle struktury hierarchii klas.

Wszystkie klasy obiektowe dziedziczą pewne wspólne własności z klasy obiektowej o nazwie *Top*. Klasy obiektowe znajdujące się nad pewną klasą są jej super klasami (nadklasami), natomiast te, które znajdują się pod nią są jej tzw. *subordinates* (podklasami).
Klasa obiektowa *Top* nie posiada swojej super klasy. Stanowi super klasę dla wszystkich innych klas obiektowych.
Każda klasa obiektowa znajdująca się niżej w hierarchii klas dziedziczy wszystkie własności ze swoich bezpośrednich nadklas.



Rys. 3. Zarys klas obiektowych zdefiniowanych w schemacie podstawowym.
Fig. 3. Layout of the object classes definied in base schema

3.2.3.2. NDS - DZIEDZICZENIE (ang. HERITAGE NDS)

Standard X.500 [9] formułuje formalne podstawy NDS. Ponieważ podstawowe obiekty i atrybuty standardu X.500 nie skalują się dobrze w środowisku biznesowym, tzn. brak jest możliwości zdefiniowania zasobów takich jak: komputery, drukarki , serwery druku, dlatego też przy wprowadzaniu tych typów obiektowych do NDS, f-ma Novell kierowała się zaleceniami CCITT dodając je w przezroczysty i przenośny sposób.

Dla zapewnienia dużej elastyczności, NDS zawiera definicje obiektów, atrybutów oraz innych wartości niewidocznych dla użytkownika, które są dla niego przezroczyste. Rezydują one w niewidzialnej części NDS'u, tj. w schemacie.

Dla zilustrowania mechanizmu dziedziczenia, poniżej na rys. 4. przedstawiono dziedziczenie klas na przykładzie klasy *Computer.* W tym przykładzie uwzględniono trzy klasy, tj. *Top, Device i Computer,* w którym widać, że atrybut super klasy jest łącznikiem pomiędzy klasami.

Jak wynika z tego rysunku - w przypadku atrybutów opcjonalnych - klasa *Computer* dziedziczy następujące własności:

- z klasy *Device* własności *Description, Owner, See Also*
- z klasy *Top* własności *ACL, Back link, Bindery Property,References*

Class	**Computer**	**Device**	**Top**
Super Classes	*Device* *Top*	*Top*	
Containment	**Organization* **Organizational Unit*	*Organization* *Organizational Unit*	
Named By	**CN*	*CN*	
Mandatory Attributes	**CN* **Object Class*	*CN* **Object Class*	*Object Class*
Optional Attributes	* ACL * Back link * Bindery Property * References * Description * Owner * See Also	* ACL * Back link * Bindery Property * References Description Owner See Also	ACL Back link Bindery Property References

Rys .4. Klasa *Computer - d*ziedziczenie własności z klas *Device* oraz *Top.*
Fig. 4. A *Computer* class - properties heritage from *Device* and *Top* classes.

Dla programisty schemat może być traktowany jako biblioteka klas (ang.Class Library), zawierającą podstawowe definicje obiektowe.
W celu utworzenia definicji nowego typu obiektowego, za podstawę bierze się definicję typu obiektowego już istniejącego a następnie dodaje się do niego nowe właściwości (atrybuty i metody).
W ten sposób klasa obiektowa z której wyprowadza się nowy obiekt zostaje nienaruszona, natomiast nowy typ obiektowy zostaje dołączony do biblioteki klas tzn. bez kodu źródłowego typu obiektowego z którego został wyprowadzony. Kod ten odczytywany jest (dziedziczony) z obiektu rodzicielskiego.
W celu wykorzystania tego nowego typu obiektowego przez aplikację użytkową, wymagane jest aby znana jej była lista jego parametrów wejściowych i wyjściowych.

3.2.3.3. DZIEDZICZENIE WIELOKROTNE

W sytuacji, kiedy klasa obiektowa NDS'u dziedziczy własności z więcej niż jednej swojej nadklasy, wówczas przypadek taki [10,11,12] zwany jest dziedziczeniem wielokrotnym. Mechanizm dziedziczenia wielokrotnego w NDS'ie realizowany poprzez wpisaniu na listę atrybutu super klasy wszystkich tych nadklas, z których mają być dziedziczone wszystkie własności.

3.3. DYNAMICZNY OPIS BAZY KATALOGOWEJ (ang. DYNAMIC DIRECTORY SCHEMA)

Opis Bazy Katalogowej (schemat) jest dynamiczny i rozszerzalny [7]. Możliwe jest zdefiniowanie nowych klas obiektowych oraz ich atrybutów ponad te, które zostały zdefiniowane w schemacie podstawowym.
Rozszerzenie opisu można dokonać poprzez modyfikację lub utworzenie nowych definicji klas obiektowych a następnie dodanie tych definicji do schematu podstawowego.
Ponieważ schemat został przeniesiony do regularnej przestrzeni obiektowej w Informacyjnym Drzewie Katalogowym (ang. Directory Information Tree - DIT), obiekty w nim zdefiniowane zachowują się jak normalne obiekty NDS'u, np. przy tworzeniu ich instancji oraz modyfikacji.
Dodatkowo, normalne funkcje NDS'u zawarte w DSAPI (*Directory Services Application Program Interface*) mogą być użyte do manipulowania definicjami zawartymi w schemacie. Istnieje również program narzędziowy pod nazwą Zarządca Schematu (ang. *Schema Manager)*, pozwalający na łatwą jego zmianę.

4. P O D S U M O W A N I E

NDS będąca obiektowo-zorientowaną implementacją usług katalogowych (*Directory Services*) umożliwiająca hierarchiczne spojrzenie na sieć komputerową i jej zasoby, jako część systemu operacyjnego NetWare 4.x zyskała sobie od czasu swojej promocji w 1993r. uznanie i dużą popularność. Usługi katalogowe są mechanizmem dostarczającym użytkownikom wszystkiego tego czego najbardziej potrzebują, tj. :

- jednokrotne logowanie się do sieci
- pojedyncze miejsce administrowania
- hierarchiczne wyświetlanie katalogów
- rozszerzalność, skalowalność
- rozproszony systemy bezpieczeństwa oraz replikacje

Usługi katalogowe doczekały się również nowych opracowań oraz oryginalnych implementacji w innych sieciowych systemach operacyjnych; przykładowo w systemie Windows NT Server istnieją w postaci Usług Katalogowych Nowej Generacji - NG.

Jedną z najważniejszych cech NDS'u jest możliwość rozszerzenia schematu za pomocą łatwych w użyciu API (*Application Program Interface*), za pomocą którego można dodać do schematu nowe klasy obiektowe.

Dostępność w ramach SDK (*SoftWare Developer's Kit*) bogatego zestawu API oraz usług wspiera rozwój aplikacji i narzędzi wykorzystujących schematy rozszerzone o nowe klasy obiektowe. Jedną z takich usług jest tzw. *Snapin Services*, pozwalająca istniejącym narzędziom administracyjnym (*NetWare Administrator Utility*) wykorzystywać schematy rozszerzone.
Jak pokazuje praktyka, NDS w ostatnim czasie coraz bardziej zyskuje na znaczeniu. Po jego rozszerzeniu użytkownicy zyskali dostęp do sieci Internet. Obecnie dostępne już są w wersji beta sterowniki NDS ODBC f-my Novell, za pomocą których projektanci wykorzystujący popularne pakiety narzędziowe Visual Basic, Delphi oraz PowerBuilder zyskują dostęp do informacji zawartych w bazie informacyjnej NDS'u.

L I T E R A T U R A

1. HERBON G.B. *„An introduction to NetWare Directory Services"* , Novell Application Notes, April 1993
2. REKOSZ D. „Cechy Funkcjonalne Systemu Sieciowego Novell NetWare 4.1 na tle różnych topologii sieciowych ", Zeszyty Naukowe Politechniki Śląskiej, INFORMATYKA z.32, Gliwice, 1997
3. JONES R.E. *„Overview of NetWare 4.0 New Features"* , Novell Application Notes, April 1993
4. *„How to Determine the Size of NDS"*, Novell Technical Information Document, document ID:TID0113464, 16 Sept. 1993

5. *„What Does DIB Stand For”*, Novell Technical Information Document, document ID:FYI.P.12612, 26 May 1993
6. *„Introduction to NetWare Directory Services”*, Novell Support Connection, NetWare 4.1 Manual Set, Novell Inc, 1996
7. CROSSEN N., WILLIAMS J., HERRIN S., *„Overview of Novell Directory Services”* , Novell Research, Feb. 1996
8. *„NDS Technical Overview”*, Novell Support Connection, version 1.3.JS, April 1997
9. KUNZE B., *„Applying X.500 Naming Conventions to NDS”* , Novell Research, Jan. 1996
10. COAD P., YOURDON E. *„Object-Oriented Design”,* PRENTICE Hall, Inc., 1991, ISBN 83-85769-18-8
11. *„NetWare Directory Services Schema Specification”*, Novell Support Connection, version 1.3.JS, April 1997
12. AUGUSTYN D., STĄPOR K., „*Obiektowo Zorientowany System Zarządzania Bazą Danych* O_2”, Zeszyty Naukowe Politechniki Śląskiej, INFORMATYKA z.31, Gliwice, 1996

Abstract

In this article a logical model of the object oriented implementation of the information database called Directory Services, for storage different information types about all objects in the network has been presented. One-to-one relationalship between schema and directory tree components has been described also. Futhemore an architecture and schema of the NDS with object classes definied in the base schema and properties heritage machanism has been presented too.