# DOSTOSOWANIE BAZY X.500 DO SPECYFIKI JĘZYKA LOKALNEGO *

**Maja Górecka**
Maja.Gorecka@cc.uni.torun.pl

*Ogólnouczelniany Ośrodek Obliczeniowy*

**Tomasz Wolniewicz**
twoln@mat.uni.torun.pl

*Wydział Matematyki i Informatyki*

## 1 Wstęp

Celem niniejszego artykułu jest przedstawienie problemów związanych z eksploatacją bazy danych o międzynarodowym zasięgu. Od kilku lat koordynujemy usługę X.500 Directory w Polsce i przedstawiamy doświadczenia i przemyślenia zdobyte w trakcie tej pracy.

Artykuł prezentuje założenia teoretyczne, jak również opisuje sposób zaimplementowania ich w praktyce w interfejsie WWW.

## 2 Specyfika bazy międzynarodowej

Baza typu X.500 jest systemem pozwalającym na gromadzenie szeroko rozumianej informacji adresowej o osobach i instytucjach. W założeniach informacja jest udostępniana całemu światu (w pewnych sytuacjach z ograniczeniami narzuconymi m.in. koniecznością ochrony lokalnych danych). Główny problem związany z dostępem do bazy pojawia się po zauważeniu, że użytkownicy dzielą się na dwie kategorie: użytkowników posługujących się tym samym językiem co obszar, którego dotyczą dane (język ten nazywamy lokalnym) i użytkowników obcojęzycznych.

Podstawowe zagadnienie, z którym mamy do czynienia to język udostępnienia danych. Problemem jest zarówno sam język, jak i specyficzne znaki stosowane w pisowni.

### 2.1 Udostępnianie informacji w języku lokalnym

Dane w języku lokalnym są zazwyczaj wystarczające w celu zaadresowania listu lub wybrania innej formy kontaktu. Dotyczy to zarówno użytkownika lokalnego jak i obcojęzycznego. Problemem może być jednak graficzna prezentacja informacji (znaków diakrytycznych). Użytkownik obcojęzyczny na ogół zakłada, że znaki diakrytyczne są zbędnym balastem, utrudniają mu odczytanie, napisanie i zapamiętanie nazwy. Użycie nazwy bez tych znaków zazwyczaj

*Autorzy pragną w tym miejscu podziękować mgr. inż. Jerzemu Żenkiewiczowi, który z ramienia NASK koordynuje prace w zakresie X.500, za kilkuletnią współpracę i wsparcie naszych działań. Panu prof. Jerzemu Brzezińskiemu dziękujemy za pomoc w zakresie teorii baz danych i uspójnienie naszej wiedzy. Poparcie NASK umożliwiało nam prowadzenie prac oraz branie udziału w europejskich spotkaniach koordynacyjnych X.500

i tak pozwoli na skomunikowanie się. Powodów takiego stanu rzeczy należy szukać w niedostatecznym rozwoju technik komputerowych, przyczynia się do tego dominacja języka angielskiego. Stosowanie stron kodowych o pojemności zaledwie 256 znaków i kodowanie znaków zgodnie z rozkładem wybranej strony doprowadza do sytuacji, w której prawidłowa interpretacja tekstu może być niemożliwa. Brak odpowiedniej standaryzacji, na przyklad wielość formatów Latin II, wprowadza dodatkowe utrudnienia.

Niewątpliwie docelowo komputerowe środowisko pracy będzie udostępniało pełny zestaw znaków, służą temu różne metody kodowania niezależne od strony kodowej (Unicode, SGML entities, standard T.61). Wydaje się jednak, że chwila ta jest od nas odległa jeszcze o kilka lat i należy starać się, aby już teraz poprawić sytuację tam, gdzie to jest możliwe za pomocą dostępnych środków.

Pamiętając o specyfice wymagań użytkownika obcojęzycznego, musimy jednocześnie zwrócić uwagę, że dla użytkownika lokalnego znaki diakrytyczne są absolutnie niezbędne. Baza danych, która będzie myliła nazwiska *Makowski* i *Mąkowski* nie będzie mogła być w Polsce uznana za profesjonalne narzędzie. Uznacza to, że baza musi niezbędnie zawierać informacje w prawidłowym formacie, ale być może niekoniecznie udostępniać je wszystkim, gdyż, przykładowo, generowanie znaków zakodowanych w Latin II na ekranie Latin I, produkuje dane zafałszowane bardziej niż gdyby zostały one po prostu pozbawione znaków diakrytycznych.

### 2.2 Udostępnianie informacji obcojęzycznej

Poza nazwami własnymi, baza zawiera również informacje opisowe. Zresztą nawet same nazwy własne niosą na ogół pewną informację typu opisowego. Ta informacja musi być udostępniana użytkownikowi obcojęzycznemu, nie chodzi tutaj o sposób kodowania, a o język. Prowadzone są prace nad internacjonalizacją X.500 w taki sposób, aby serwery, wyposażone w odpowiednie oprogramowanie same generowały informację we właściwym języku. U postaw tego projektu leży założenie, ze dane byłyby wprowadzone w jednym formacie, prezentacja byłaby generowana na ich podstawie przez serwer. Wydaje się, że jest to interesujący pomysł, ale po pierwsze jest on dopiero w trakcie opracowywania, a po drugie nie jest oczywiste, czy oferowane przez niego rozwiązania będą zadowalające. Na przykład dowolne tłumacznie nazwy firmy na obcy język może być niedopuszczalne dla danej firmy, gdyż na ogół posiada ona ustaloną nazwę nie tylko lokalną, ale i w kilku obcych językach.

Wydaje się, że w obecnej sytuacji praktycznym rozwiązaniem jest dublowanie informacji w języku lokalnym i języku angielskim. Takie założenie zostało zaimplementowane w polskiej bazie X.500.

## 3 Specyfika struktury bazy danych X.500 a nazewnictwo

### 3.1 Hierarchia

Obiekty bazy X.500 uporządkowane są w strukturę hierarchiczną. Na szczycie tej struktury typowo umieszcza się państwa, na niższym poziomie instytucje, następnie ich podjednostki itd, aż do poziomu osób, pojedynczych obiektów (np. aplikacji komputerowych) itp.

## 3.2 Nazewnictwo

Każdy obiekt opisywany jest przez zestaw atrybutów, na ogół wielowartościowych. W typowej sytuacji jedna z wartości pewnego atrybutu jest wyróżniana jako właściwa nazwa obiektu. Taka nazwa musi być jedyna na danym poziomie hierarchii, ale może się pojawiać ponownie na innych poziomach. Obiekt opisywany jest za pomocą ciągu nazw kolejnych poziomów hierarchii bazy. W taki sposób pełna nazwa jest unikalna i zapewnia szybką lokalizaję obiektu w bazie. Tak skonstruowaną nazwę określa się terminem *nazwy wyróżnionej.*

## 3.3 Sposoby dostępu do danych a język prezentacji

Typowy dostęp do adresowej bazy danych realizowany jest na jeden z dwóch sposobów:

1. przeglądanie
2. wyszukiwanie

Pierwsza metoda jest bardzo często preferowana. Użytkownik przegląda kolejne poziomy hierarchii organizacji aż dojdzie do danych, które go interesują. W tym podejściu użytkownik nie zawsze wie co chce znaleźć, przeglada informację, zatrzymując się w miejscach, które go zaciekawią.

Metoda przeszukiwania stosowana jest przez użytkownika, który stosunkowo dobrze wie czego szuka i zależy mu na szybkim otrzymaniu wyniku. Jeżeli dane do przeszukania są niedokładne, to możemy mieć do czynienia z połączeniem metody wyszukiwania i przeglądania, gdyż użytkownik będzie przeglądał listę wyników przeszukania, próbując odnaleźć dane, których potrzebuje.

Użytkownik stosujący metodę przeglądania ma dostęp jedynie do głównych nazw obiektów i na ich podstawie musi sobie wyrobić zdanie o kolejnym etapie poszukiwań. W sytuacji kiedy mamy do czynienia z użytkownikiem obcojęzycznym, a nazwy główne są prezentowane w języku lokalnym, można spodziewać się problemów w zrozumieniu nazw, a prezentowanie ich z uwzględnieniem znaków diakrytycznych i za pomocą nieznanej strony kodowej może całkowicie uniemożliwić korzystanie z bazy.

Stosowanie metody przeszukania daje dostęp do większej gamy atrybutów, przykładowo Uniwersytet Mikołaja Kopernika w Toruniu jest reprezentowany przez wartość główną atrybutu o nazwie `organizationalName` równą `Uniwersytet Mikolaja Kopernika w Toruniu`, ale atrybut ten posiada również wartości:
`Uniwersytet Mikolaja Kopernika w Toruniu`, `Uniwersytet Mikolaja Kopernika`, `UMK`, `University of Torun`, `Nicholas Copernicus University in Torun`,
`Nicholas Copernicus University`.
Oznacza to, że przeszukanie ze względu na `organizationalName=umk` odnajdzie ten obiekt, pomimo, że wartość `UMK` nigdy nie pojawi sie przy listach otrzymywanych z przeglądania.

Jako wynik wypisywania obiektów danego poziomu hierarchii lub przeszukania poddrzewa zaczynającego się od pewnego obiektu bazowego otrzymujemy listę zawierającą ciąg nazw wyróżnionych znalezionych obiektów. Na ogół z nazw wyróżnionych odcina się przedrostek będący nazwą obiektu bazowego. Wyprowadzenie ciągu nazw zwracanych przez operacje przeszukania lub wypisywanie danych bieżącego poziomu jest realizowane na podstawie algorytmów wbudowanych w eksploatowany system, bardzo to ogranicza możliwości stosowania różnych sposobów nazewnictwa, czy przetworzenia tej listy na zestaw odpowiednich łańcuchów czytelnych lokalnie.

# 4 Dostosowanie bazy X.500 do pracy dwujęzycznej

## 4.1 Założenia

Polski projekt usługi X.500 bazuje na oprogramowaniu QUIPU, które należy do pakietu ISODE, będącego implementacją protokołów OSI. QUIPU realizuje wszystkie protokoły zdefiniowane w ramach standardu X.500, dostarcza serwer bazy danych oraz oprogramowanie dające użytkowy dostęp do X.500.

Dopuszcza się pracę polskich serwerów X.500 w oparciu o oprogramowanie QUIPU w wersji ISODE-8.0 (edycja *public domain*) oraz IC (edycja dystrybuowana przez organizację Isode Consortium, udostępniana instytucjom akademickim na podstawie nieodpłatnej licencji). Wymaganie to jest niezbędne m.in. z powodu konieczności modyfikacji oprogramowania obsługi X.500 (serwer DSA oraz moduły biblioteczne) do potrzeb obsługi strony kodowej Latin II.

Zakłada się, że w celu dostosowania bazy X.500 do pracy w języku polskim stosowane będą wspólne tablice definiujące obiekty w ramach polskiego poddrzewa.

Dodatkowo przyjmujemy, że polscy użytkownicy będą korzystać z bazy za pomocą specjalnie przystosowanych interfejsów.

## 4.2 Struktura bazy

### 4.2.1 Model informacyjny

W bazie X.500 stosuje się specyfikacje CCITT ASN.1 (*Abstract Syntax Notation*) do definiowania struktur danych. Obiekty grupowane są w klasy i opisywane za pomocą ustalonych administracyjnie atybutów. W momencie definiowania atrybutu określa się reguły syntaktyczne, którym dany atrybut podlega. Pozwala to na kontrolę typu informacji przechowywanej w danym atrubucie, jak również właściwe jej interpretowanie.

Elementy bazy X.500, takie jak klasa obiektu czy atrybut, definiowane są w ASN.1 za pomocą typu zwanego identyfikatorem obiektu (`OBJECT IDENTIFIER`), który określa autorytatywnie nazwany obiekt. Identyfikatory te uporządkowane są w strukturę drzewa, mają postać ciągu nieujemnych liczb całkowitych, z których każda jest etykietą kolejnego węzła drzewa na drodze od korzenia do bieżącej pozycji. Postaci numerycznej opisującej węzeł towarzyszy na ogół krótki opis tekstowy.

System przyporządkowania identyfikatorów jest zestandaryzowany. Obecnie drzewo identyfikatorów obiektów ma trzy podstawowe gałęzie:

- `ccitt(0)`, administrowany przez CCITT;
- `iso(1)`, administrowany przez ISO/IEC;
- `joint-iso-ccitt(2)`, administrowany wspólnie przez ISO/IEC i CCITT.

co ASN.1 opisuje następująco:

```
ccitt           OBJECT IDENTIFIER ::= { 0 }
iso             OBJECT IDENTIFIER ::= { 1 }
joint-iso-ccitt OBJECT IDENTIFIER ::= { 2 }
```

Dalej np. na gałęzi `ccitt` zdefiniowano:

```
data            OBJECT IDENTIFIER ::= { ccitt 9 }
pss             OBJECT IDENTIFIER ::= { data  2342}
quipu           OBJECT IDENTIFIER ::= { pss 19200300 }
```

System taki pozwala na jednoznaczne przyporządkowanie identyfikatorów i delegowanie odpowiedzialności za administrowanie gałęziami drzewa identyfikatorów do organizacji krajowych i dalej do instytucji.

Po przydzieleniu gałęzi drzewa, instytucja lub kraj, ma pełną swobodę w definiowaniu swoich własnych elementów. Oczywiście lokalnie dodane klasy (czy atrybuty) nie będą interpetowane przez użytkowników nie mających dostępu do właściwych definicji (dzięki jednoznaczności nie zachodzi jednak niebezpieczeństwo interpretacji fałszywej). W standardzie X.500 (88) podstawowe identyfikatory rozpowszechniane są za pomocą statycznych tablic, w wersji 93 przewiduje się umieszczenie definicji w samym drzewie informacji. Takie rozwiązanie spowoduje, że niezdefiniowane obiekty nie będą się w ogóle pojawiały.

### 4.2.2 Standard T.61

Jedną z reguł syntaktycznych stosowanych w X.500 jest wykorzystanie standardu telekomunikacyjnego T.61 do kodowania wszelkich znaków diakrytycznych. Łańcuchy zawierające wyłącznie znaki czytelne mają przypisany syntaks `PrintableString`, takie w których dopuszcza się znaki specjalne określa się jako `T61String`. Tego typu łańcuch charakteryzuje się nagłówkiem w postaci identyfikatora {T.61} i zawiera, poza standardowymi znakami ASCII kody znaków diakrytycznych. Rozszerzeniem syntaksu `T61String` jest `CaseIgnoreString`, który dopuszcza pełne łańcuchy T.61, a jednocześnie zakłada, że przy porównaniach nie będzie się rozróżniać małych i wielkich liter. Jest to jeden z najszerzej stosowanych syntaksów.

Narodowe znaki specjalne z akcentami mają w standardzie T.61 postać dwóch bajtów: pierwszy to kod akcentu, drugi kod znaku podstawowego, inne są reprezentowane jako jeden bajt i występują w tabeli na pozycjach powyżej 127. Bardzo istotne jest, że T.61 jednoznacznie koduje znaki bez wykorzystania pojęcia strony tabeli kodowej.

Poniżej przedstawiono fragment tabeli kodów T.61:

| Kod (hex) | Opis | Znak |
|---|---|---|
| c1 | grave accent | ` |
| c2 | accute accent | ´ |
| c3 | circumflex | ˆ |
| c4 | tilde | ˜ |
| c5 | macron | ¯ |
| c7 | dot | ˙ |
| c8 | diaresis | ¨ |
| c9 | umlaut | ¨ |
| ca | ring | ˚ |
| cb | cedilla | ¸ |
| cc | underline | _ |
| cd | umlaut | ˝ |
| ce | ogonek | ˛ |
| e8 | L z kreską (Ł) | Ł |
| f8 | l z kreską (ł) | ł |

### 4.2.3 Rozszerzenia w celu dostosowania bazy do lokalnych potrzeb

W celu przystosowania bazy X.500 do zapamiętywania w niej polskich nazw zostały ustalone niezbędne rozszerzenia, związane z definicją nowych klas obiektów i atrybutów.

Do tablic konfiguracyjnych wprowadzono polskie identyfikatory:

```
nask:                   data.2602
umk:                    nask.1
polishQuipu:            umk.1
polishObjectClass:      polishQuipu.1
polishAttributeType:    polishQuipu.2
```

Zdefinowano nowe atrybuty, m.in.:

```
polishCnName:   polishAttributeType.1 :caseIgnoreString (imie i nazwisko)
polishOName:    polishAttributeType.3 :caseIgnoreString (nazwa instytucji)
polishRDN:      polishAttributeType.19:caseIgnoreString (polska nazwa wyróżniona)
```

Atrybuty te zwierają łańcuchy w poprawnej polskiej pisowni, zapisane przy pomocy standardu T.61.

Konieczne było również zdefiniowanie nowych klas obiektów dopuszczających polskie atrybuty.

Polskie klasy obiektów i atrybuty powstały w celu umożliwienia wprowadzenia podwójnego nazewnictwa obiektów. Przyjęto, że jako wyróżnioną nazwę obiektu stosować się będzie określenia pozbawione znaków diakrytycznych, natomiast dodatkowo lokalne dane będą opisywane za pomocą specjalnych atrybutów, uwzględniających polskie znaki specjalne. Mamy wówczas do czynienia z pewnego rodzaju dublowaniem informacji, ale jednocześnie realizowane jest nazywanie obiektów zgodnie z potrzebą polskich użytkowników i w taki sposób, że odbiorca niezainteresowany narodową pisownią danych może ją bez problemu zaniedbać.

Dzięki oparciu definicji nowych atrybutów o istniejące syntaksy nie było potrzebne tworzenie dodatkowych funkcji obsługi struktur danych.

Przyjęte rozwiązanie nazywania polskich obiektów wymusiło wprowadzenie polskiej nazwy wyróżnionej, tak by na każdym poziomie można było jednoznacznie wybrać prawidłową lokalną nazwę, atrybut `polishRDN` jest obowiązkowy dla każdego polskiego obiektu, reprezentowanego jako `newPolishObject`.

## 4.3 RFC 1617

Polska baza X.500 jest fragmentem projektu PARADISE-NameFlow koordynowanego przez DANTE i mającego na celu tworzenie bazy X.500 dla środowiska naukowego. Sprawna praca takiej międzynarodowej bazy danych wymaga ustaleń co do sposobu nazewnictwa. Krajowe systemy X.500 działające w ramach PARADISE powinny stosować zalecenia zawarte w RFC 1617. Jednym z zaleceń jest zakaz stosowania łańcuchów T.61 w głównych nazwach obiektów. Wynika to z konieczności zapewnienia czytelnego wyświetlania nazw wyróżnionych.

Zalecenie RFC 1617 przewidujące wpisywanie nazw w poprawnej pisowni do kolejnych wartości atrybutów jest całkowicie nie do przyjęcia w przypadku, gdy chcemy stworzyć system, który polskiemu użytkownikowi będzie wyświetlał **wyłącznie** prawidłowe formy nazw.

Stosowanie wielokrotnych wartości atrybutów powoduje, że nie ma żadej możliwości automatycznego decydowania o tym, która wartość powinna być wyświetlona, a która nie.

Na podstawie doświadczeń z polską bazą przedstawiliśmy PARADISE projekt nazewnictwa uwzględniający pracę dwujęzyczną, w sposób nie ograniczony do pojedynczego kraju ([4]). W projekcie tym przewidujemy stosowanie atrybutów w wersji lokalnej i miedzynarodowej, pozostawiając interfejsowi użytkownika decyzję o tym, z której wersji korzystać. Przyjęcie tego typu zaleceń jest sprawą nadzwyczaj trudną, gdyż wymaga znacznych modyfikacji bazy danych. W tej sytuacji na razie zmuszeni jesteśmy korzystać z mniej optymalnych rozwiązań lokalnych.

## 4.4 Prezentacja wyników

Wiekszość aktualnie funkcjonujących aplikacji komputerowych stosuje pojęcie strony kodowej. W ten sposób użytkownik ustala, z jakiego zestawu czcionek będzie korzystał. Zmiana strony kodowej powoduje wyświetlanie innych symboli na wysokich miejscach (powyżej 127) tabeli kodów. Bezpośrednim efektem takiego podejścia jest niemożność wyświetlania na jednym ekranie jednocześnie znaków z różnych języków. Stosowanie tabeli kodowych jest szczególnie uciążliwe w przypadkach niejednoznaczych standardów (np. tabela Latin II (852) stosowana przez Microsoft i tabela ISO8859-2)

Wykorzystanie w X.500 standardu T.61 pozwala na jednoznaczne zakodowanie znaków łacińskich wszystkich języków europejskich, problemem pozostaje jednak ich wyświetlanie. Oczywiście nic nie stoi na przeszkodzie, aby aplikacja stosowała różne czcionki w ramach jednego ekranu i w ten sposób była całkowicie uniwersalna, ale nie jest to jak na razie powszechne podejście. Głównym powodem takiego stanu jest najprawdopodobniej fakt, że większość zachodnioeuropejskich języków mieści się w ramach podstawowej tabeli ISO8859-1.

Ponieważ obecnie typowy użytkownik na świecie ma do dyspozycji interfejs wyświetlający znaki zgodnie z tabelą Latin I należy założyć, że nie ma on dostępu do czcionek prezentujących polskie znaki i zostaną one zastąpione znakami nieczytelnymi. Dlatego niezbędne jest, by użytkownik obcojęzyczny bazował na nazwach pozbawionych polskich akcentów, a więc w podstawowych atrybutach bazy X.500 należy stosować nazwy bez polskich znaków diakrytycznych.

Interfejsy użytkowe muszą mieć wbudowaną możliwość wyboru strony kodowej, z którą współpracują. Takie też było założenie twórców oprogramowania QUIPU, rodzaj stosowanego zestawu znaków może zostać określony poprzez zmienną środowiskową lub za pomocą odpowiednich ustawień w użytkowych plikach konfiguracyjnych, przy czym, jak dotychczas, QUIPU dopuszcza wyłącznie ASCII i Latin I. Dokonana przez nas modyfikacja oprogramowania QUIPU w tym zakresie dopuściła stosowanie również strony Latin II w aplikacjach użytkowych.

## 4.5 Problemy efektywnościowe

Podstawowym problemem, z którym mamy do czynienia przy przyjętym rozwiązaniu realizacji dwujęzycznej bazy X.500 jest wydajność systemu. Nieuchronną konsekwencją proponowanego modelu jest konieczność wielokrotnego odczytu danych w celu dotarcia do informacji zgodnej z prawidłową polską pisownią. Jak wcześniej wspomnieliśmy, obiekty w bazie X.500 są jednoznacznie identyfikowane poprzez swoje nazwy wyróżnione (*Distinguished Names*). Nazwy te występują w wersji międzynarodowej, w której polskie znaki diakrytyczne są pozbawione akcentów.

Użytkownik podając wzorzec przeszukania lub na podstawie przeglądania kolejnych poziomów drzewa informacji dociera do interesujących go danych. Niestety, hasło odczytywane posiada wyłącznie atrybut będący relatwną polską nazwą wyróżnioną, nie są znane polskie odpowiedniki nazw obiektów leżących na ścieżce drzewa informacji prowadzącej do docelowego obiektu. Oznacza to m.in., że wyszukując osobę reprezentowaną w bazie pod nazwą podstawową: `Wladyslaw Lopuszanski` jesteśmy w stanie odczytać lokalną nazwę obiektu: `Władysław Łopuszański`. Przyjmijmy, że nazwa wyróżniona obiektu ma postać:

```
Wladyslaw Lopuszanski, Dzial Techniczny, Wydzial Chemii,
Wyzsza Szkola Inzynierska w Zielonej Gorze, Polska
```

Chcąc wyprowadzić informację dotyczącą nazwy organizacji (zgodnie z polską pisownią), w których zlokalizowano obiekt konieczne jest odczytanie danych dotyczących lokalnych nazw haseł powyżej bieżącego (a wiec haseł trzech poziomów), tak by dotrzeć do polskich relatywnych nazw wyróżnionych i móc wyświetlić pełne dane o obiekcie:

```
Władysław Łopuszański, Dział Techniczny, Wydział Chemii,
Wyższa Szkoła Inżynierska w Zielonej Górze, Polska
```

Oczywiście taka technika znacznie wydłuża proces docierania do całości interesującej użytkownika informacji. W rozdziale 5 przedstawimy, w jaki sposób można zminimalizować koszt przedstawionej propozycji dostępu do lokalnych danych.

# 5 Implementacja

W bieżącym rozdziale opisujemy implementację modyfikacji interfejsów użytkowych X.500 w celu ich dostosowania do wykorzystania polskich danych umieszczonych w bazie.

Prace związane z tym tematem rozpoczęliśmy dwa lata temu. Początkowo zajęliśmy się programem `de`, który był w tym czasie najczęściej stosowanym interfejsem użytkowymi X.500. Następnie, gdy dynamicznie rosnąca popularność usługi World Wide Web przyczyniła sie do powstania programu umożliwiającego korzystanie z bazy X.500 poprzez interfejs kliencki WWW (tzw. WWW-browser, jak np. *Lynx, Netscape, Mosaic*), przedmiotem prac stał się moduł `web500gw`, pełniący funkcję tzw. *gateway'a* pomiędzy WWW a bazą X.500, obecnie jeden z powszechnie wykorzystywanych interfejsów X.500.

## 5.1 Program `de`

`DE` (*Directory Enquiries*) to bardzo prosty interfejs użytkowy X.500, rozwijany w ramach europejskich projektów COSINE i VALUE, autorstwa P. Barkera z Uniwersytetu Londyńskiego. Program oferuje dane X.500 Directory poprzez terminal liniowy. Dostępne są dwie wersje `de`:

1. działająca według standardowego dla X.500 protokołu DAP (*Directory Access Protocol*), zapewniającego komunikację pomiędzy aplikacją użytkową X.500 a serwerem DSA,

2. oparta o uproszczony protokół LDAP (*Lightweight Directory Access Protocol*), w którym kontakt z serwerem X.500 następuje poprzez serwer LDAP, komunikujący się następnie z DSA.

Prace modyfikacyjne polegały na dołączeniu w programie odczytu, w przypadku dostępu do obiektów polskiego poddrzewa X.500, wartości lokalnych atrybutów oraz prezentacji wyników zgodnie z obowiązującą stroną kodową.

W programie `de` wybór zestawu znaków odbywa się przy starcie aplikacji, decydują tu parametry konfiguracyjne interfejsu użytkowego, które są pobierane ze specjalnego pliku użytkownika, albo z pliku systemowego.

W przypadku `de` opartego na protokole DAP dostosowanie interfejsu do współpracy ze stroną Latin II wymagało zmian w oprogramowaniu QUIPU. W modułach bibliotecznych QUIPU dołączono interpretację kodów znaków w ramach obsługi syntaksów konkretnych atrybutów.

W LDAP `de` było możliwe ograniczenie zmian do samego kodu źródłowego programu, w którym zostały dołączone funkcje translujące zestaw T.61 do postaci Latin II.

„Spolszczone" `de` wykorzystuje polskie opisy umieszczone w lokalnych atrybutach i prezentuje je na ekranie. Uzyskanie pełnej informacji na temat hasła, zgodnie z tym co opisaliśmy w rozdziale 4.5, pociąga za sobą odczyt polskich danych obiektów położonych na ścieżce prowadzącej do bieżącego hasła w drzewie informacji, co zostało zaimplementowane w programie.

`De` pozwala również wyszukiwać informacje z bazy X.500 za pomocą wzorców zawierających polskie znaki diakrytyczne.

Modyfikacja programu `de` odegrała bardzo ważna rolę w zrozumieniu wielu problemów związanych z interpretacją i prezentacją znaków narodowych w aplikacjach X.500. Prace te zmusiły nas do poznania tajników oprogramowania QUIPU i znacznie usprawniły dalsze działania dotyczące polskich interfejsów X.500.

## 5.2 Program `web500gw`

Autorem *gateway'a* pomiędzy WWW a X.500 jest Frank Richter z Uniwersytetu w Chemnitz-Zwickau.

Modyfikacja tego programu zmierzała w dwóch kierunkach. Podstawowym celem było wykorzystanie do prezentacji informacji zawartej w ramach dodatkowych, lokalnych atrybutów i „spolszczenie" programu poprzez zastąpienie angielskich tekstów objaśniających polskimi. Kolejnym zadaniem było wbudowanie w program możliwości wysyłania formularza zawierającego dane aktualizujące informacje zamieszczone w X.500.

`Web500gw` pozwala uzyskać dostęp do danych X.500 Directory. Jego działanie bazuje na protokole LDAP, będącym uproszczoną wersją standardowego protokołu X.500, zwanego DAP (*Directory Access Protocol*), zapewniającego komunikację pomiędzy aplikacją użytkową X.500 a serwerem.

Zgodnie z przyjętym założeniem uwzględniono możliwość stosowania następujących stron kodowych: ASCII, Latin I, Latin II, IBM (CP852), Windows-EE. Wybrany zestaw znaków może zostać zmieniony w trakcie korzystania z programu, przekodowanie do odpowiedniego formatu następuje dynamicznie.

Ponieważ program `web500gw` jest typową aplikacją kliencką World Wide Web, jego dokumenty, czyli teksty wyświetlane przez program, są przygotowywane w języku HTML (*HyperText Markup Language*). HTML jest podzbiorem bardziej ogólnego języka — SGML (*Standard Generalized Markup Language*). Znaki specjalne są przedstawiane w HTML'u, podobnie jak w SGML'u, jako tzw. *SGML-entities*.

Oto przykładowe odpowiedniki:

| | | |
|---|---|---|
| > | — | `>` |
| Ą | — | `Ą` |
| Œ | — | `Œ` |

Program `web500gw` zmodyfikowano w taki sposób, że wszelkie znaki o kodach powyżej 127 zapamiętywane są wewnętrznie jako tzw. *SGML-entities*, również informacje pobrane z bazy X.500 translowane są z kodu T.61 do postaci SGML. Następnie, w trakcie wyprowadzania danych na ekran następuje konwersja do postaci aktualnej strony kodowej poprzez zastosowanie odpowiedniego filtru przekodowującego. Obecnie graficzne programy klienckie WWW akceptują wyłącznie *SGML-entities* z zakresu alfabetu Latin I, ale ponieważ sam standard SGML zawiera formalną definicję nazw znaków innych stron kodowych, należy się spodziewać, że wkrótce będą one interpretowane w interfejsach graficznych WWW. Zastosowane w `web500gw` podejście do kodowania znaków diakrytycznych pozwoli w tym momencie w prosty sposób zmodyfikować kod źródłowy i pozbawić program balastu translacji z postaci SGML.

Polska wersja programu `web500gw` wykorzystuje możliwość umieszczania w bazie X.500 obiektów opisanych za pomocą atrybutów lokalnych, o których była mowa w rozdziale 4.2.3.

W przypadku odczytu hasła w ramach polskiego poddrzewa informacji sprawdzamy, czy obiekt należy do klasy `newPolishObject`, jeżeli tak, odczytywana jest wartość atrybutu `polishRDN` (jest to atrybut obowiązkowy dla tej klasy i jednowartościowy), która jednoznacznie decyduje o nazwie głównej obiektu. Pobierane są również informacje umieszczone w innych lokalnych atrybutach (m.in. `polishDescription`, `polishTitle`, `polishPostalAddress` itp.)

Z przyczyn efektywnościowych przedstawionych w rozdziale 4.5 konieczne było zaimplementowanie algorytmu *cachowania*, czyli przechowywania w podręcznej pamięci danych przemapowujących nazwę wyróżnioną obiektu w jego postać lokalną. Zasada działania programu `web500gw`, który startując uruchamia proces macierzysty oczekujący na połączenia, a każde odwołanie typu *HREF* generuje nowy proces wymusiła potrzebę stosowania pamięci współdzielonej jako *cache*'a. Konieczna jest kontrola zajętości pamięci podręcznej, jej wielkość jest programowo ograniczona poprzez ustawienie zgodne z argumentem wywołania programu lub w przypadku jego braku przyjmowana jest domyślnie pojemność niezbędna do przechowania 800 przemapowań. Przyjęto rotacyjne umieszczanie kolejnych elementów w pamięci. *Cache* ma postać par:

(nazwa wyróżniona obiektu, wartość atrybutu `polishRDN`)

przy czym `polishRDN` jest przechowywany w postaci SGML, tzn. zawiera *SGML-entities* w miejscu znaków diakrytycznych. Optymalizuje to wykorzystanie pamięci podręcznej i stwarza warunki do dynamicznego przekodowywania nazw do postaci obowiązującej strony kodowej.

Drugą istotną modyfikacją zaimplementowaną w `web500gw` było udostępnienie, w przypadku odczytu danych osobowych, formularza aktualizacji. W swojej oryginalnej postaci `web500gw` udostępnia eksperymentalnie funkcję aktualizacji danych bezpośrednio w bazie X.500. Uprawniono do tego właścicieli obiektów klasy *Person*, znających swoje hasło zabezpieczające (*password*). W ramach polskiej bazy X.500 zrezygnowaliśmy z funkcji bezpośredniej aktualizacji, w zamian oferujemy możliwość przesłania za pomocą poczty elektronicznej specjalnego formularza z nowymi danymi. Ponieważ tematyka związana z aktualizacją danych w ramach X.500 znacznie wykracza poza ramy naszego artykułu, problem ten zostanie opisany w odrębnym raporcie. W tym miejscu warto jedynie nadmienić, że dane w formularzu mogą zawierać polskie znaki diakrytyczne, które niezależnie od stosowanej strony kodowej zostaną prawidłowo dostarczone do administratora (w standardzie T.61).

## Bibliografia


[1] M. Górecka, T. Wolniewicz, *X.500 - standard i usługi katalogowe*, kwiecień 1995, materiały konferencyjne Polman.

[2] M. Górecka, T. Wolniewicz, *Stosowanie znaków diakrytycznych w systemach baz danych X.500*, wrzesień 1994, raport NASK.

[3] P. Barker, S. Kille, T. Lenggenhager (May 1994) *Naming and Structuring Guidelines for X.500 Directory Pilots*, Request for Comments RFC1618.

[4] M. Górecka, T. Wolniewicz, *"Naming in the X.500 Directory"*, 1995, propozycja zmiany nazewnictwa obiektow w projekcie Paradise przesłana do Dante i koordynatorow X.500.