Use of national languages in X.500 Directory

Maja Górecka

and

Tomasz Wolniewicz

Torun, Poland

Introduction

In this paper we summarise work and research done for the Polish X.500 project. Problems encountered when working on making the system accessible by a Polish user, as well as solutions developed seem to be of quite universal nature.

We would like to note here the existence of the MAITS project which has goals much wider then we have put in front of ourselves. The output of that project is likely to answer a large variety of problems related to use of computers in the multilingual world, still we believe that there is place for interim solutions which can be implemented without too much effort.

We would also like to stress the point that our work is targeted at the research community which usually has a good knowledge of English, our solutions could be unacceptable in a very general approach.

The article consists of three main parts. First we identify general language related problems appearing in running and accessing an international database. Next we describe how we have addressed these problems in the Polish X.500 pilot. Our solution has the advantage of being easily implemented in a country with a single official language, without the need for any coordination on the international scale. In the third part we present an approach to the problem that could be implemented in a world-wide X.500 directory, in a universal way, providing a much larger functionality from the international user point of view.

The main assumption of our solutions was the compatibility with the currently running NameFlow PARADISE X.500 Directory service.

International Directory problems

An international Directory project like X.500 NameFlow PARADISE, has a number of quite special problems, on several levels.

1. The data is widely distributed and of international nature, consisting mainly of information about organisations and individual people.
2. The data is accessed from a large number of environments, ranging from 7-bit text terminals to windows applications of various kind, including WWW browsers.
3. The data is accessed by international users.

We have identified that expectations, as far as Directory format and information are concerned, divide users into two groups:

- Local users, typically the inhabitants of the country which the data concerns. They comprehend the native language of the data and they appreciate if the spelling is correct (including all accented letters).
- Foreign users, typically do not understand the language of the data, they do not care about the fine details of spelling. These users, especially within the research community will be satisfied with getting the information in English.

Local user requirements

Our experience is based on the Polish language, but we know that similar problems arise in many countries and languages. The local user will require that his *local* data is in his own language, secondly he is likely to be equipped with a computer system that can display his national characters correctly and he will want the data appearing in its correct spelling. This may depend from one language to another, but there are instances, where accents are no mere *decorations*, as foreigners like to see them, but carry a lot of information. The lack of a single accent may change a word into another that is not *comparable* to the original. Local user will not want to bothered with other versions of names (for instance accent-less versions) if he already has a correct one.

Foreign user requirements

A foreign user does not (typically) have the knowledge of the local language and hence the understanding of the data is his priority. Only seldom this understanding can come from looking at the name in its local language. To understand that one only needs to realise that a Polish name *Wyzsza Szkola Rolnicza* means *College of Agriculture* (the Polish name was given in its accentless version). A foreign user may need to have an original version of the name but not for the reason of understanding its content. Finally a foreign user, nowadays will not probably be equipped with necessary software or fonts, to display names in their accentfull spelling. The emerging standards of representing various characters without the use of 8-bit code pages are likely to change this situation, but there is considerable expense behind implementing them and English speaking users do not feel the need strongly enough to make enough pressure on vendors, the fact that Western Europe is essentially covered by the standard ISO-Latin-1 code page also takes off a lot of pressure.

In our opinion a foreign user, while probably preferring names appearing in his own language, does not make it a requirement. Definitely in research community the English language is seen as a universal medium of communication. Political reasons force use of multiple languages officially, non official projects may use simplifications that will be universally accepted by users.

Technical limitations

We have already mentioned that there are technical limitations due to a lack of implementation of certain character sets, but it often happens that in case where no fixed standards existed, intermediate local pseudo-standards were developed. These standards were sometimes hard-coded into software, or even hardware, ad it will take time until they are eliminated. To make things more complicated even the `standard' representations may have different formats, ISO-Latin-2 is not the same as Latin-2 (CP 852) used by Microsoft, which in turn is not like Microsoft Windows EE (Eastern Europe) code page.

Having this in mind we see that there is an obvious requirement that the database holds data in a unique format and the user interface makes necessary translations. This assumption will again not be true since users in the international community will use all sorts of user interfaces, often locally developed, which will not always have enough translation capabilities. It may be thus necessary to have the database server supply the data in an already transliterated format.

International use of X.500 directory

X.500 directory of a global type (like the NameFlow PARADISE project) attracts international users. This is confirmed by the logs of the system. It is also very typical, that users of the directory do not have enough information about the target of their search to enable the system to return a single result. Very often the browsing method is being used, simply going from one level of information to another, until the interesting entry is located, indeed users will often not use the directory to find a specific answer, but to gain some sort of overview. The straight browsing method will be impractical with large data sets, but even then, users will probably form general search criteria and browse in the results. Hence it is crucial that the results are given in a format that is comprehensible to a foreign user. By a comprehensible format we mean character sets in the first place, using wrong display fonts with a wrong character sets will generate results that are not only simply lacking some information, but are often false as well.

A local user will often have more specific information on the subject she searches for, but will often use browsing techniques as well. For such a user it is a critical requirement that no irrelevant information is being shown (like accentless versions next to accented ones). There is also usually no need to supply other language forms of the data, unless the user specifically requires. The system should be capable of using all popular ‘pseudo standards’ for displaying extended character sets or provide a reduction to 7-bit ASCII if there is no other option.

X.500 specifics

X.500 Directory standard defines a distributed database organised into a tree hierarchy. On each level of the tree entries are named by a relative distinguished name - RDN. RDN is formed as a collection of

(attributeType, attributeValue)

pairs (usually just a single pair). The information in the entry is kept in attributes in the form of unordered sets. The ordered sequence of RDNs starting from the root of the tree, ending in the entry’s RDN forms a distinguished name - DN, which uniquely identifies the entry in the tree.

One large advantage of the X.500 design is its flexibility - local administrators can modify standard data structures, add their own attribute types and so on. Such local extensions can be made visible outside when the description of data structures (schema) is published and installed. It is in this way possible for each country to have its own extensions which will not have influence onto the international look of their data.

Whenever a list or a search operation is performed by the X.500 directory, a list of results will be returned consisting of DNs of entries found. It is possible to return certain (or all) entry attributes together with the names. The fact that results are given as DN lists makes it difficult to present those lists in various language and character formats. This is due to the assumption that the RDNs should be restricted to the 7bit ASCII. On the other hand translation of these lists into the local language is exactly the goal we need to achieve.

Another important point is the way of connecting to the Directory. This is done by connecting to some local directory server a special client user agent and from this point the system navigates itself through the tree completely hiding the distribution of information from the user. To simplify access, X.500 to WWW gateways exist and then users can use standard WWW browsers by connecting to the gateway which will handle all the X.500 protocol. It is assumed that a local user will use a local access point to the directory, which implies that foreign user will use foreign access points. This assumption tells us that foreign access points must not suffer from local modifications. We must also realise that it is impossible to expect that foreign servers will be modified to facilitate this. We will, later on, suggest that world-wide changes should be made, but even then we insist that backward compatibility has to be maintained.

The X.500 (‘88) standard supports extended character sets via the T.61 encoding standard. Accented letters are encoded by an accent code followed by the letter, special characters have their own codes. This representation covers most Latin based languages in Europe. It is up to the user interface to convert T.61 codes into screen characters. It must be mentioned, that most of available software will only cover conversion to characters within the ISO-Latin-1 character table, usually behaving unexpectedly at characters from outside of this table. This deficiency implies that T.61 characters should be avoided within RDNs since bad handling of those can make browsing impossible. It has become standard in the NameFlow PARADISE project that local language is used for RDNs (but only 7-bit ASCII). This makes RDNs unsuitable for both local and foreign users, except for countries which can well cope with the reduced character set.

The introduction of WWW gateways should make it possible to provide simple multilanguage support, via translation of T.61 codes into HTML entities. This approach, even using standard code-page oriented fonts allows to display characters from various code pages within one line of text. In fact this is implemented in the X.500 interface of the Euromath system [1]. The new version (‘93) of the X.500 standard [2] supports the Unicode encoding as well, we expect however that it will take a considerable amount of time to convert the entire Directory, since there is currently no freely available software supporting the new standard and on top of that any global conversion of the data is a slow process.

The main goal

After presenting the main aspects of the problem we can now easily specify what needs to be achieved.

Local user must have full local language support, i.e.:

1. attribute names are presented in the local language,
2. attribute values of local entries appear in local language only, unless specifically requested otherwise,
3. search and list results must be returned in the local language i.e., the DNs need to be converted into local names if necessary,

4. popular local character representations need to be supported including the 7-bit ASCII representation,
5. input in local language including accents needs to be supported.

The minimum support of foreign users must consist of:

1. RDNs must be restricted to the 7-bit ASCII character set,
2. Entries, especially at the higher levels of the tree must contain enough information in English to provide some understanding of the object identity.

The extended support for foreign users would require that certain client software is available (WWW gateways, for instance) and it would then provide:

1. search and list results returned in English,
2. attribute values given in their English form,
3. attribute values in their local language transliterated to a chosen standard character set.

Solutions in the Polish X.500 project

Localised attributes

In order to support the local user requirements it is necessary to store all entry information in its full form as T.61 strings. It is a standard practice in the PARADISE project to store various versions of one attribute value as multiple values of this attribute. Various language forms, 7bit ASCII as well as T.61 versions of an attribute value will be represented, it is then up to the user to 'filter out' the information himself. The software cannot make any decisions since attribute values are not marked in any way.

The only solution to distinguish between the information targeted at the local user and that at the foreign one is to have a special attribute type replacing the standard one. The value held in the localised attribute may also normally be present in the corresponding standard one, this data duplication is unavoidable overhead of the proposed system.

The Directory interface needs to know whether a localised version of a given standard attribute type is present to make a decision whether to hide the standard attribute. To facilitate this there must be a simple mapping between standard attributes and their localised versions. Such a mapping can be done either on name or the corresponding OID value basis; we have chosen the first approach. This way to an attribute type of commonName there corresponds one of polishCommonName, and so on. Localised attributes may be multivalued the same way as the standard attributes are.

In order to support foreign users we have issued a recommendation to use English forms of the data as alternative attribute values. This way a show entry operation will provide some information that should be comprehensible to foreign users. No English information will be present in listings, but it should be possible to search for English strings.

Localised RDNs

Localised attributes solve the problem of showing correct language forms of entries, but they are of no help when list or search results are being returned. Such search results will consist of a list of DNs, and possibly some attributes of the entries found. As an example, when a search for a person is run under an organisation the directory may return a DN that will contain the person's RDN as well as the RDN's of departments where this person was found. Depending on the depth of the organisational structure there may be many levels of such organisational units above the person's entry.

The user interface needs to convert this DN into a list of names that will correspond to the RDNs appearing in this DN. The interface will need to read entries of the organisational units above the person and get correct forms of the RDNs. Since localised attributes may be multivalued there is a need to have a localised copy of the RDN attribute value. In our solution a single attribute type polishRDN of caseIgnoreString syntax is being used for this purpose. The interface will read this attribute value, which needs to be unique, and display it in place of the RDN. We have not found the solution of using the single syntax limiting so far.

In its simplest form the described approach will result in a serious loss of efficiency. Together with every search operation a number of read operations will need to be performed to obtain the RDN translations. In practice however, those read operations will very often address the same RDNs and if a translation cache is kept on the server side, it is possible to significantly reduce the number of read operations. We have implemented this mechanism in a WWW gateway. HTTP requests are handled by a single daemon which can build up the cache. Since the server handles the incoming requests by forking off children the cache space needs to be common to all those processes, in our case shared memory is being used for efficiency reasons.

Alternative approach - multiattribute RDNs

We have also considered another approach that would make the RDN remapping operations unnecessary. As we have mentioned before, the standard allows that RDNs are formed as a collection of (attributeType, attributeValue) pairs. If we have changed the naming and instead of

organizationalUnitName=Wyzsza Szkola Rolnicza

used something like

organizationalUnitName=Wyzsza Szkola Rolnicza%polishOrganizationalUnitName={T.61}Wy\cazsza Szko\f8a Rolnicza

we could have the correct form present in the RDN next to the standard 7bit ASCII one. We could in fact hold an English form of the name next to the full Polish form. Client software that is aware of such naming could then pick up only a portion of the name, depending on the user needs. The disadvantage of this approach is that software that is not aware of this system will not be able to understand the local addition to the RDN and will obscure the listing of the directory by raw ASN.1 data. We have decided that only a major decision on the international level could make this system usable by forcing it onto a large community, hence this system is completely unsuitable for local extensions.

Implementation

As we have already mentioned the presented solutions have been implemented in a WWW - X.500 interface. We have extensively modified the web500gw interface by Frank Richter from Technical University in Chemnitz-Zwickau to make it aware of the special naming system. The RDN to local translations need to be cached, the introduction of that made the system considerably faster. All server instances use the same cache space (shared memory), this way a cache update which was done for a particular request is immediately usable by another server instance.

Another important factor was the existence of multiple coding standards in Poland. Since the users still access WWW via text terminals, sometimes based on simple PCs that some character coding standard put into the hardware we felt that we need to support at least four encoding: 7bit ASCII, Microsoft code page 852, Microsoft Windows CE code page, ISO 8859-2. To make the system easy to modify we have chosen a common internal representation of the data, using SGML entities. Only the final form of the data is transliterated to the code page that a user has chosen. This common form was also necessary to hold a common cache space. The choice of SGML entities was motivated by the expectation that WWW browsers should soon be able to handle them in a straightforward fashion, without any conversion (as is already true for the ISO 8859-1 range), when this happens it will be possible to switch off the conversion in the gateway completely. One disadvantage that was observed is the difficulty in sorting. It is possible that a switch from SGML entities to Unicode representation will be made in future releases.

The global approach

We would now like to discuss the limitations of the presented approach.

The main goal of our implementation was to provide the Polish user with full language support, this was achieved completely. The second goal was to provide the basic support for the foreign user, this again was done. The extended support of the foreign user is much more difficult and would require changes in international recommendations.

Another limitation of the Polish system is no support for multiple local languages. This would be a requirement in many countries. We also realise that in certain situations a foreign user may prefer to use the original language of the data rather then its internationalised (English) form.

International, localised and standard attributes

Simple extension

The simplest extension of the presented Polish system is by an addition of yet another copy of standard attributes - international attributes, those would hold English versions of the names and information. Local versions could be held in localised attributes, so that in place of polishCommonName we would have localizedCommonName. Standard attributes would still need to be kept for compatibility. Both localizedRDN and internationalRDN would be needed, unless it was decided that real RDN will all be kept in their international form (we do not suggest that they are kept in the localised form to provide the basic foreign user support without any specialised software). Access software would allow users to switch between foreign and local forms, but could also make automatic decisions based on the data location and user access point. Such system should be quite useful in all cases where there is no real need to hold several language versions of the localised attributes and would then provide both the full local and extended foreign user support. Since this is a requirement in many countries we assume that the simplified international scheme is not general enough.

Generalised extension

The generalised system keeps the international (English) version of attributes, and the internationalRDN in particular, but also allows for multiple language forms of localizedAttributes. Such multiple languages values will appear together in a result of the show entry operation (together with the international version if the user so desires), since they will be normally grouped by the attribute type such presentation should be quite readable. To provide correct versions of listings and search results we need to introduce an extension to the localizedRDN attributes. We suggest that their actual value is proceeded by a language code. It must be assumed that if a localised RDN for a given language exists then also most of the attributes will have their versions in this language.

A user accessing the Directory specifies a language of his choice. If the system finds that a localised RDN for the chosen language exists it displays the localised version, otherwise the international version is printed. Since we require that together with a certain language version of the RDN also most of the attributes will appear in this language amongst the localised version, we will to have a situation where the user will be given the information in a totally incomprehensible format. The listings and search results will normally be translated in full into the chosen language or into English if the particular data forms do not exist.

It should be stressed that we would only expect official languages of the region being used to keep the uniformity of the presentation.

We cannot expect to put language tags onto attributes other then localizedRDNs as this would make various search operations complicated, we think however that multiple languages when displaying a single entry will not make the system overloaded.

The presented system does not loose anything in the case of a single-language country.

Bibliography

[1] M. Gorecka, R.M. Timoney, T.M. Wolniewicz, „The Euromath Interface to X.500 Directory Services", Euromath Bulletin Vol. 2, No. 1, 1996

[2] D. Chadwick, „Understanding X.500 The Directory", Chapman & Hall, 1994