Multilingual implementations of OSI applications

C. Bouras[1,2] D. Fotakis[2] V. Kapoulas[1,2]
S. Kontogiannis[2] P. Lampsas[1,2] P. Spirakis[1,2]
A. Tatakis[2]

[1]*Computer Technology Institute,*
[2]*Department of Computer Engineering and Informatics*
University of Patras

Contact author: Vaggelis Kapoulas, Computer Technology Institute, Kolokotroni 3, 262 21 Patras, Greece. Tel.: +30 61 220112 (ext. 380), Fax: +30 61 222086, E-mail: kapoulas@cti.gr

Abstract

In this work the problem of supporting multilingual environments over OSI applications is addressed. Initially a solution based on the idea of using *multilingual structures* and transcription techniques is presented. Another solution based on the major principles of the former follows, which exploits the coding of all of character sets uniformally.

Keywords: ISO/OSI, Character Sets, Transcription, ISO 10646.
Technical session: Computer Networks

1 Introduction

The need for developing multilingual distributed applications is imposed by the aspects of user friendliness and ease of use. Just a multilingual interface would not be a solution to the problem because a multilingual application is expected not only to represent, but also to process either latin or non-latin information. An example is the multilingual version of the OSI Directory, where a DUA[1] should be able to look for a non-latin object by using the specific object's alphabet besides the latin one. Furthermore, a DSA[1] is expected to take into account the special characteristics of the corresponding language for latin or non-latin information in operations such as approximate or case insensitive matching.

[1] The acronyms DUA and DSA stand for Directory User Agent and Directory System Agent respectively.

The current work presents two independent solutions for the multilingual versions of OSI applications. The first one adds many character sets in an OSI application by defining new structures in ASN.1 for handling non-latin information. The second solution is based on some character set coding techniques and tries to restrict the necessary changes in the augmented implementations of the application, leaving the communication protocols as they are.

Throughout this paper the proposed solutions are applied to the X.500 Directory System and X.400 MHS, in order to give examples of how these solutions may be used in OSI applications.

2 Non-latin characters at the ASN.1 level

The handling of textual information by OSI applications should take into account the special features of the specific character set, as well as the features of the corresponding language. The major problems of this new way of handling textual information have to do with the Presentation Layer of the communication process, such as character set incompatibility. Some other problems such as functions that depend on the special characteristics of a language (i.e. phonetic matching) must be overcome in the Application Layer. So, our interest is focused on the necessary changes in these two layers, and especially in the Presentation Layer because the changes of the second category are application-dependent.

The definition of an ISO 8859 string (besides the Latin-1), is not included in the ASN.1. This means that either an existing type (such as the EXTERNAL type) will be used for embedding values corresponding to types from different abstract and/or transfer syntax, or new types must be defined.

Initially a new presentation context, suitable for the representation of the values of the abstract syntax is required. The next step is the detailed specification of the syntaxes. In this way the abstract syntax of this new presentation context will be specified by the declaration in ASN.1, showing that ISO 8859 characters are accepted as a sequence of bytes (octets).

Consequently the transfer syntax is specified, which will be used in combination with the previously defined abstract syntax and which is going to accept only sequences of octets as inputs, in the following way: Every octet to be transferred is a character coded in an ISO 8859 character set (i.e. ISO 8859-5 Cyrillic).

Finally, values are given to the defined syntaxes, as object identifiers which are registered in the Object Identifier Tree (OIT).

3 A first solution to the problem

In this solution, the key objective is to handle multilingual text and provide some complementary information about the corresponding language of the character set in use.

A new structure in ASN.1 is defined for holding strings written in various character sets, that will include information about the corresponding language. If an OSI application attempts to retrieve some information written in a specific character set, it must be able to process and/or represent it properly to the user.

In the case that the application does not support the specific character set, it should either reject the results of the request, or try to give the user an approximate representation, at user's will. The latter is achieved by the transcription techniques. There are two kinds of transcription. The first is used for coding the characters of a non-latin alphabet with characters of the latin one in a reversible way. The second transcription technique attempts to achieve readability of the specific non-latin string with latin characters and it is not reversible[2]. The kind of transcription that should be used depends on the nature and the needs of the specific OSI application. The new structure in ASN.1, called *multilingualString*, is shown below:

```
multilingualString ::= SET {
    alphabet         [0]  Alphabet OPTIONAL DEFAULT latin-1,
    origAlphabet     [1]  Alphabet OPTIONAL,
    value            [2]  Value }
Alphabet ::= INTEGER {
    latin-1(1), latin-2(2), latin-3(3), latin-4(4), cyrillic(5),
    arabic(6), greek(7), hebrew(8), turkish(9), sami(10) }
    −−other ISO 8859 character sets may be included in the future
Value ::= CHOICE {
    T61String,
    PrintableString,
    GeneralString }
    −− any other string type, defined in ASN.1
```

The *value* field holds the string itself. This structure is constructed so as to use the ISO 8859 Coding Series. When the *value* field contains a string of an ISO 8859 character set, the *Alphabet* field is used to declare this character set. In this case the *origAlphabet* field holds no valid information. In the case that the OSI application does not support the specific character set of the textual information, and the user has declared his strong interest for it, the information is transcripted[3] and while the *Alphabet* field declares the latin alphabet, the *origAlphabet* field determines the original alphabet of the string before the transcription.

The *multilingualText* structure defined below, is a collection of *multilingualStrings* for supporting multilingual textual information:

[2] Many national organizations have standardized these two transcription techniques.

[3] The first kind of transcription is used if reversibility is required, otherwise the second kind is preferred.

```
multilingualText ::= SET {
    alphabetSet    [0]  SET OF Alphabet
    text           [1]  SEQUENCE OF multilingualString }
```

The *alphabetSet* field declares the various alphabets used in the multilingual text for handshaking before the communication process. In the case that there is an alphabet incompatibility between the requester of some information and the character sets used in the result of the request, then either the whole information is discarded or the incompatible strings are transcripted, at user's will. For example in the OSI Directory, the *multilingualString* structure is embedded in the informational model by defining the corresponding syntax as follows:

```
multilingualStringSyntax ATTRIBUTE−SYNTAX
    multilingualString
    MATCHES FOR EQUALITY SUBSTRINGS
    ::={attributeSyntax 13}
```

A rational hypothesis is that an OSI application supports a specific subset, and not all the existing alphabets, according to the possible users' languages. For example in a Directory System, a DUA is expected to support at most one non-latin alphabet, besides the ISO 8859 Latin-1. On the other hand a DSA should support many alphabets, but again it is unlikely that it will support all the existing alphabets.

So during the phase of connection establishment, a handshaking process will take place before the two communicating parts begin their interaction. In this process the counterparts of the communication will declare the supported character sets and if they are interested in the transcripted information in case of alphabet incompatibility. So the structures used for the connection establishment will have to be augmented.

In our specific example of the OSI Directory, some changes are recommended in the *DirectoryBind* operation in order to enforce the declaration of the supported alphabet set by a process (DUA or DSA) that requests a service by another.

```
DirectoryBindArgument ::= SET {
    credentials           [0]  Credentials OPTIONAL,
    versions              [1]  Versions DEFAULT v1988,
    supportedAlphabets    [2]  SET OF Alphabet }
```

The remote DSA accepts the request only if the set of alphabets that are declared in the *DirectoryBindArguments* is a subset of the set of alphabets that it supports. With this convention we avoid the case of having a request that cannot be processed by the DSA that serves it or, a reply that cannot be handled by the requester because of alphabet incompatibility.

A multilingual OSI application should also be able to choose the textual information according to the alphabet that it is written in. For example if a description of an object is available in the greek, the cyrillic and the english alphabet, a greek user would like to see the greek version while an english user would prefer the english one. On the other hand, a greek user that his application cannot support the greek character set (ISO 8859-7), would like to see the greek message transcripted with the second technique and store its transcripted version with the first technique so as to be able to restore the initial greek description.

Returning to the example of the Directory System, the previous decision is made by the *foreign* flag of the *EntryInformationSelection* structure. The value of the *foreign* flag is taken into account only if the attribute's values are asked to be retrieved and the attribute's syntax is *multilingualStringSyntax*. The augmented *EntryInformationSelection* structure is described below:

```
EntryInformationSelection ::= SET {
    attributeType
        CHOICE {
            allAttributes      [0]  NULL,
            select             [1]  SET OF AttributeTypes,
            typeAndAlphabet    [2]  SET OF AttrTypeAndAlphabet
            -- empty sets imply no attribute is requested
        DEFAULT allAttributes NULL,
    InfoTypes INTEGER {
        attributeTypesOnly (0), attributeTypesAndValues (1)}
        DEFAULT attributeTypesAndValues }
AttrTypeAndAlphabet ::= SEQUENCE {
    type    AttributeType,
    foreign BOOLEAN DEFAULT TRUE }
```

All the previously described changes affect also the communicating protocols.

Evaluation of the proposed solution

This solution describes all the necessary changes for universal support of multiple alphabets by OSI applications and gives the potential to intermediate nodes to handle the non-latin information according to its special characteristics. These changes must be adopted by the corresponding standardization committees.

The main problem of the proposed model is the incompatibility with the existing implementations. But this model suggests only some extensions to the existing structures and only the multilingualString is a new structure. Yet, even this structure is based on existing string structures defined in ASN.1. So, provided that a conventional version of the application can ignore the extra information, there will be no problems in cooperating with augmented versions, in the standard latin alphabet.

4 An alternative model

This solution is based on the following observation:

Since an OSI application can transfer only specific types of coded text, it should be given only these types of coded character sets, for transferring.

An efficient way for this would be the encoding of non-latin text to the appropriate type before being transferred via the transfer system used by the OSI application (i.e. the X.400 Message Handling System), and the decoding of a text to its initial format when it is received at the destination (end-to-end processing).

The encoded text must be transferable (i.e. IA5String, T61String, etc.) and the conversion must be reversible. The encoding of non-latin characters is achieved using the T.61 standard.

During the communication process the sender will convert the initial text in T61String or generalString, send it through a transfer system and the recipient will restore it. The above described model requires to augment only the User Agents of the existing implementations of OSI applications (i.e. UAs for X.400, DUAs for X.500, etc.). So, no changes are required at the ASN.1 level. Moreover, intermediate conventional nodes can forward appropriately an IA5String, T61String, or generalString. Finally, the proposed encoding already exists for non-latin character sets.

There is a standard methodology for the selection of a graphic character set, the so-called *ISO 2022 world*. This standardization defines a special way for the selection of characters from a number of character sets. In this way the capability of using more than 94 or 96 graphic characters is achieved. The *ISO 2022* standard describes an automaton, which can use up to four character sets (G zone) coming from different code tables and up to two control character sets (C zone)[4]. Using the control character <ESC>, the designation of up to four character sets is achieved, to one of the G zones. The invocation and utilization of these zones is achieved with the control characters LS0, LS1, LS2, LS3.

An OSI application may introduce a non-latin character set in one of the G zones and in this way overcomes the problem of using non-latin characters in a text. For example, in the Message Handling System (MHS) a User Agent may use the internationally standardized non-latin character sets that are available[5].

The composition of a multilingual text to be sent is achieved by placing the basic latin character set IR-102 in the G0 zone and the supplementary set IR-103 (accents and symbols) in the G2 zone (initial configuration according to the X.209 Recommendation). The G0 zone is initially invoked in the active left set and the G2 zone in active right set. For example, in a multilingual MHS, if the message contains non-latin characters then the corresponding standardized character set

[4] These zones are referred to as G0, G1, G2, G3 for the G zone and C0, C1 for the C zone.

[5] For the greek alphabet the IR-150, IR-70, or the IR-126 which is the part of the ISO 8859-7 that contains the greek characters.

is designated in the G0 zone and the basic latin character set is designated in the G3 zone[6]. Then, whenever required, LS0 and LS3 are used to invoke in the active left set either the non-latin or the latin character set. In message reception any designation and invocation is acceptable, if it follows the *ISO 2022 world*.

The conversion of the non-latin text in the appropriate format for transmission and the reverse operation is done in two stages. At message reception, initially the conversion from T.61 encoding to character mnemonics (as specified in ISO 10646 standard) is performed. ISO 10646 mnemonic encoding is used only for internal representation of text in the OSI application. In this way the application can accept and store any textual information in any language. For display purposes, the mnemonic encoding is converted to the locally used character set (i.e. one of the ISO 8859 character sets), in a second stage.

It is worth noting that the conversion from T.61 to an ISO 8859 character set in two stages and the use of mnemonic representation, internally, has the advantage that the ISO 10646 encoding is expected to be widely adopted, thus the multilingual versions of an OSI application can be easily enhanced to accept directly ISO 10646 encoded text. Moreover, a multilingual implementation can easily be adapted to strings using any character set by creating mapping tables between this encoding and ISO 10646 encoding, suffices.

5 Conclusions - Future work

In this work, the first solution is a complete model for multilingual OSI applications and defines some new structures in ASN.1 that might be used for supporting multilingual textual information. The major drawback of the proposed model is that these augmented structures must be standardized. This solution also allows the textual information processing to take into account the special characteristics of the specific language and intermediate processing of information before being returned to the requester (i.e. greek descriptions to greek users).

In the second solution the main objective was the compliance with the existing Recommendations Series for the OSI applications and the communication protocols used by them. The corresponding implementations are based on the ISO 10646 standard, which is expected to be used by OSI applications. This idea is compliant with the suggestions made by the RARE project ([2]). The proposed model is used mainly for multilingual textual information transfer and for end-to-end processing of this information.

In future, our estimation is that a global solution like the first model should be standardized and be applied to all the OSI applications, but until then some intermediate solutions, such as the second model may be used.

[6] The specific designation is based on the recommendations of the Greek PTT, but is not limiting for the solution.

Acknowledgments: We would like to thank the unknown referee for the valuable comments.

References

[1] Henshall, John and Sandly Shaw, *“OSI Explained: End-to-End Computer Communication Standards”*, Chischester, England.

[2] Harald Alvestrand UNINETT, *“RARE Technical Report 7, X.400 Use of Extended Character Sets”*, August 1993.

[3] FASCICLE VIII.4 - Data communication networks: Open Systems Interconnection (OSI). Model and notation, service definition. Recommendations X.200-X.219 (Study Group VII).

[4] FASCICLE VIII.5 - Data communication networks: Open Systems Interconnection (OSI). Protocol specifications, conformance testing. Recommendations X.220-X.290 (Study Group VII).

[5] Hellenic Organization for Standardization (ELOT), *“Hellenic Standard 743. Transcription of Hellenic alphabet with latin characters”*, Athens, Greece, 1993.

[6] ISO 646, *“Information Processing - 7-bit coded Character Set for Information Interchange”*, 1983.

[7] ISO 2022, *“Information Processing - 7-bit and 8-bit coded Character Sets. Code extention techniques”*, 1986.

[8] ISO 8824, *“Information Processing Systems - Open Systems Interconnection (OSI) - Specification of ASN.1”*, 1987.

[9] ISO 8825, *“Information Processing Systems - Open Systems Interconnection (OSI) - Specification of Basic Encoding Rules for ASN.1”*, 1987.

[10] ISO 8859, *“Information Processing - 8-bit single-byte Coded Graphic character sets”*, 1987.

[11] International Telegraph and Telephone Consultative Committee, *“Character repertoire and coded character sets for the International Teletex service”*, Recommendation T.61, 1988.

[12] C. Bouras, D. Fotakis, V. Kapoulas, S. Kontogiannis, P. Spirakis, *“Hellenization of the Recommendations Series X.500”* CTI-TR 95.4.14, April 1995.

[13] C.Bouras, V. Kapoulas, P. Lampsas, G. Papoutsopoulos, P. Spirakis, A. Tatakis, G. Theodoropoulos, *"Helios: An implementation of X.400 supporting Hellenic characters"* CTI-TR 95.4.12, April 1995.