Unicode: a tool for system interoperability and human communication

Cathie Jilovsky
Information Services Manager
CAVAL Collaborative Solutions
cathiej@caval.edu.au

Abstract:
This paper describes the development of the Unicode standard, how it is being used by computer systems generally and by the library community in particular. The complexities of implementing a system which supports a multi-language environment are explored, using the implementation of the Ex Libris Aleph 500 library system at CAVAL Collaborative Solutions as a case study. CAVAL provides cataloguing services to customers in over fifty languages, so that the ability to handle Unicode was a key criterion in the selection of the system. Unicode is complex to implement but is becoming an essential component for library systems.

Introduction

Unicode is an international character set developed for the scripts of all the languages of the world. It is becoming a fundamental tool for communicating and exchanging data between computer systems. The power of Unicode is that it enables human communication across languages, cultures and geographic areas. However, although it is complex to understand and to implement, it provides for system interoperability that is essential in the electronic age. Unicode must be embraced by libraries and information services in order to remain relevant in the wider community.

CAVAL Collaborative Solutions, a consortium of the University Libraries in Victoria and the State Library of Victoria, provides a range of services to the library and information communities throughout Australasia. One of these services is the provision of multi-language cataloguing. As a result, when CAVAL staff surveyed the library system marketplace in 2001, the ability to use the Unicode format was a key factor in the selection of a system. The system selected and subsequently purchased was the Aleph 500 system from Ex Libris.

The implementation process proved to be quite complex and more protracted than originally envisaged. The Implementation Team had to grapple with a range of issues - ranging from understanding the MARC21 and Unicode standards; to developing internal workflows and procedures along with processes for the transfer and conversion of records to and from a variety of external systems. At this time many of the external systems either do not support Unicode at all or only support a portion of it. This presented the team with a range of challenges to be addressed. In particular, reference is made to the National Library of Australia's Kinetica[i] system, as like most Australian libraries, CAVAL sources from and contributes high quality records to the National Bibliographic Database.

The development of Unicode

In the early days of computer development, most usage was scientific and most of the data was numeric which could easily be represented by only 4 bits. The ASCII encoding system, using 7 bits, is sufficient to represent the ordinary Latin alphabet and was developed in the late 1950s. Today most computers use 8 bits (one byte) to represent a character. As this is not sufficient for the representation of additional characters, such as diacritics (accent marks) and non-Latin alphabets, a variety of other encoding systems were developed to meet the needs of different languages. [Erickson 1997]

Unicode was devised so that one unique code is used to represent each character, even if that character is used in multiple languages. It is a complex standard and is not possible to describe without some technical detail. The major computer companies were the drivers in the development of Unicode, and its incorporation into many commonly used software products today provides a satisfactory platform for the support of multiple languages. [Wells 2000]

Since the introduction of the Unicode standard in 1991, its scope has developed considerably. The original 2 byte (16 bit) encoding has expanded to 4 bytes (32 bits), allowing for the representation of over 1 million characters. However, in order to facilitate data transmission, a series of Unicode Transformation Formats (UTFs) have been designed, the most common being the single-byte (8 bit) UTF-8. [Felici 2002] The big advantage of UTF-8 being that

ASCII and its equivalent Unicode characters have the same value, making it more compatible with existing software. [Tull 2002]

The Unicode Consortium is an international organisation responsible for the development and promotion of the standard. [Aliprand 2000] Today Unicode is incorporated into many software products, including Apple, Java (SUN), Microsoft Windows (NT, 2000 and XP), Internet Explorer, Netscape Navigator, and Oracle. Unicode has enabled system developers to create products which are independent of the encoding, and can therefore perform consistently regardless of the language of the user interface. Although this system development took place primarily for maintenance rationalisation and economic reasons, it has provided a platform for multi-language systems. [Felici 2002]

Understanding the associated terminology is essential - for example, the distinctions between a character and a glyph, and between fonts and character sets. A character is the semantic value of a letter, symbol or number, and glyphs are the shapes of these that appear on the screen or printed page. Fonts contain glyphs not characters. The paper by Erickson contains a very useful glossary. [Erickson 1997] The Unicode site [http://www.unicode.org] contains a wide range of documentation about the standard and its usage. Version 4.0.0 is the latest version of the Unicode standard and was released in April 2003.

There are a number of websites which provide practical information about Unicode and how to use it to produce standardised multilingual and technical documents. [Kass 2003, Wood 2003]. Cunningham's paper discusses the use of Unicode on multi-lingual websites [Cunningham, 1999]. Unicode is also not without its detractors. [Holmes 2003]

The implementation of software which supports multiple languages entails the consideration of a range of issues beyond the data being stored in Unicode. These include sorting and display issues. Sorting data containing more than one language is difficult, as the sorting order of characters varies between languages. Understanding why a system is unable to display a particular character is often complex, e.g. it may have no information about that character, or may lack the right tool (such as a font) to display that character. Aliprand recommends that systems provide a helpful message in these situations. [Aliprand 2000]

Usage of Unicode in libraries

A review of the library literature identified a number of papers discussing and describing the Unicode standard but very few with details of library-specific implementations.

Tull reviews the use of Unicode and its incorporation into library systems. She notes that the crucial issue for libraries is now how Unicode will affect the exchange of records and that Unicode is just one aspect of support for languages in a library system. Other considerations are "are language translation tools, sorting mechanisms, dealing with word parsing in specific languages and developing character mapping tools during the transition phase from older character sets". Tull reports on a survey of library system vendors that found that that most companies do not plan to support all the scripts in the world, but are concentrating on the ones that their customers need. [Tull 2002]

Aliprand suggests that the development of Unicode is part of the growing digital library environment, along with the provision of services to growing international audiences and the

expectation of 24 by 7 accessibility. The implementation of Unicode into computer operating systems and other software is flowing through into library systems and library standards, such as MARC and Z39.50. For example Version 3 of the Z39.50 standard provides for character set negotiation between the origin and target systems. [Aliprand 2000]

Library system vendors are beginning to incorporate Unicode into their products. There are already a number of systems on the market which can store data internally in Unicode, but the literature indicates that the exchanging of library data between systems in Unicode is minimal as yet. This will become a growing issue, as changes will be required to the large bibliographic databases from which libraries source their records in order to support both the storage and exchange of records in Unicode. Other associated issues include display issues, fonts, sorting, and storage requirements. Data storage requirements vary according to the encoding form, for example if UTF-8 is used, the space requirements relate directly to the character content of the data.

The MARC21 specification specifies the use of Unicode for the exchange of records. Currently the encoding of MARC records is limited to UTF-8, as it is recognised that there will be a period of transition to the 16-bit Unicode environment. Several positions in the MARC record leader take on new meanings e.g. the record length contained in Leader positions 0-4 is a count of the number of octets in the record, not characters. Diacritical marks must be encoded following the base letter they modify which is the opposite of the MARC-8 rule for encoding order. [MARC 21 Specification 2003]

The conversion of existing legacy databases into Unicode will be a challenge for libraries. However this will need to happen in order that the exchange of bibliographic data can continue in the Unicode world. The CHASE (Character set standardisation) project investigated the extent of the work required to convert the databases of national European bibliographic services to Unicode, as well as practical issues relating to the provision of bibliographic services in Europe. [CHASE 1997]

Aliprand points out that in pre-computer catalogue days compromises were often made between a totally accurate transcription and a form which could be easily and reliably found by searchers. This applies with equal validity in the automated environment, i.e. facilitating retrieval does not necessarily mean being precise in every typographical detail. [Aliprand 2000]

Diacritics (or accent marks) are used to remedy the shortcomings of the ordinary Latin alphabet for recording languages which have additional sounds. [Wells 2000] Transliteration is the spelling or the representation of characters and words in one alphabet by using another alphabet. In the English speaking world, romanisation is commonly used in library catalogues. However to achieve this, specialised knowledge is required by both library cataloguers and users. Other libraries have taken a simpler approach and ignored diacritical distinctions. [Erickson 1997]

Chandrakar examines the challenges of using bibliographic databases based on Indian scripts for a range of functions including storing, retrieving, sorting, filing and resource sharing. He concludes that Unicode is the only solution for creating such multi-script bibliographic databases. [Chandrakar 2002]

Zhang and Zeng describe the development and use of specialised character sets for Chinese characters, using multi-bytes for each character. As a result, one Chinese character may have different codes in different character sets. Libraries have been forced to build their own multi-script applications because commercial software has not supported the variety of scripts needed. However, the growth of the internet, along with web client software and web search engines that handle multi-lingual data, have pushed the demands for a single encoding for Chinese characters. The authors note that "as well as supporting the interchange, processing and display of written texts of the many and diverse languages of the modern world, Unicode also supports classical and historical texts of many written languages, as well as technical symbols in common use". Zhang and Zeng focus on what they perceive to be obstacles in the use of Unicode in library applications, especially in relation to Chinese. [Zhang and Zeng 1999]

The support of multiple languages at Ohio State University libraries since the 1980s is described in a paper to be published in Library Hi-Tech. [Unicode ... 2003] A plea for the continued expansion of the MARC21 character sets to match the Unicode character set is made. A range of issues relating to the configuration of workstations, including operating systems, input methods, web browsers and fonts, are described and suggestions for making practical decisions are incorporated. Guidance with system implementation such as the importance of separating input, storage and display issues; the complexity of sorting data containing text in multiple languages; and the pros and cons of including Unicode data in the library OPAC rather than linking to a separate system is provided. There is a useful section on guidelines for troubleshooting a range of possible problems.

Libraries should be specifying Unicode conformity when looking at new systems. Almost all computer manufacturers, operating system developers and browser producers have adopted Unicode. It is required by modern standards such as XML and Java. Although ASCII support will probably continue for some years, Unicode will displace it, as Unicode can do everything ASCII can do and more. [Why Unicode, 2000]

The multi-language environment at CAVAL

CAVAL's cataloguing staff can catalogue in over 50 languages. As Australia's pre-eminent provider of multi-language cataloguing, CAVAL has a pool of professional, experienced language specialists and is always happy to seek out skills in additional languages. In partnership with a Melbourne book supplier, the Foreign Language Bookshop (see http://www.flb.com.au/default.asp/), shelf-ready foreign language material, including ESL (English as a second language), travel guides, audio/video, and preselected packs of popular titles for libraries can be selected, catalogued and processed. Assistance with the selection of materials and/or translation of publishers' brochures is also provided.

Languages in which staff are currently working include: all the European languages (including Ancient Greek and Latin), Arabic, Bengali, Chinese (pin yin and Wade-Giles), Gujarati, Hebrew, Hindi, Indonesian, Japanese, Khmer, Korean, Malay, Pali, Punjabi, Russian, Sanskrit, Sinhalese, Tagalog, Tamil, Thai, Urdu, Vietnamese, Yiddish and Yoruba.

Implementing the Aleph system at CAVAL

CAVAL is using the Aleph system for two separate applications. The first is to support the CARM Centre collection and the second is for the cataloguing services that CAVAL provides for external customers. The CARM (CAVAL Archival and Research Materials) Centre is a high-density storage facility for low-use research materials, and in addition provides space for short-term storage of collections. It has a capacity of 1 million volumes, and provides a humidity and temperature controlled environment.

For a number of reasons, the system implementation process was quite protracted. Members of the CAVAL Implementation Team initially grappled with a range of issues, ranging from understanding the MARC21, UTF and Unicode standards; to developing internal workflows and processes along with procedures for the transfer and conversion of records to and from a variety of external systems.

The system was installed and initial training took place in January 2002. Although live usage of Aleph for the CARM Centre commenced in May, minimal progress was made with the use of diacritics until Ex Libris supplied an updated version of the character conversion routines in June 2002.

Further training was delivered by Ex Libris in October, assisting members of the team with the clarification of a range of issues, and subsequent development of workflows and procedures. Some cataloguing staff began using the Aleph system from October. In December 2002, training in the use of the Aleph system was provided for all CAVAL cataloguing staff and a database of 38,000 CAVAL records exported from Kinetica was loaded into Aleph.

Although both CARM and Cataloguing services were already in production usage, a number of other essential processes were implemented and finetuned during the early part of 2003. These included the collection of statistical data, the export of records to Kinetica and the import of records from CARM member systems.

The Aleph Implementation Team approached the process with enthusiasm but soon discovered that although the theory of Unicode is simple, in practice the implementation is complex. Many hours of consultation, research, discussion and testing were needed before we felt we had sufficient understanding to use diacritics in Aleph with confidence. After initial training and systems analysis, four major areas relating to the use of diacritics in Aleph were identified: storage issues for data containing diacritics, conversion issues i.e. the use of diacritics in data exported to and imported from external systems, data input issues and the display of data containing diacritics.

In order to ensure that all CAVAL cataloguing staff could use Aleph, it proved necessary to upgrade the PC environment and to ensure that sufficient technical support was available during the implementation process. It was essential that the Aleph Client software be configured correctly on each PC, that appropriate fonts were installed and that all operating systems, web browsers and printers being used for Aleph were Unicode-aware.

1. Storage issues

The Aleph system stores data in Unicode format. However at this point in time the systems with which CAVAL exchanges records are not able to accept data in Unicode format, so data encoding conversion routines must be used whenever exporting and importing records. As the facility to import Kinetica records into Aleph and also to export Aleph records to Kinetica is essential, initial testing was undertaken with Kinetica. Files were exchanged between the two systems with records created natively in each system. The tests incorporated checking the underlying data values of the records as well as analysis of the data as displayed in both systems.

The troubleshooting process proved to be a complex one. Diagnosis of a problem could potentially involve the operating system, the web browser, fonts used for screen display, fonts used for printing, character encoding and/or the input method editor.

2. Conversion issues

It took the team some time to understand all the issues relating to the conversion of diacritics in data exported to, and imported from, external systems. Once these were understood a structured testing process was developed. Initially the focus was on transferring data to and from the National Library of Australia's Kinetica system. Staff at the National Library of Australia were extremely helpful during the testing phases and worked closely with CAVAL staff to achieve the best outcomes.

Whilst initial exporting and importing of records between systems shows no variation in terms of the underlying data values, it was found that after several attempts at importing and exporting a record, occasionally several of the underlying data values changed after export from Kinetica or Aleph, and so subsequent attempts at reimporting the record into either system would contain incorrect values for these diacritics.

This was determined to be for several reasons:

- KineticaWeb, the source used for most testing, was confirmed to not support the full MARC8 character set, and to substitute values upon export for some unsupported diacritics/character combinations. This, of course, affects any further importing or exporting of the affected diacritics. An example of this is the 'Eth Lower' character which is exported from KineticaWeb with a hexadecimal value of '1F' (Subfield delimiter) instead of the correct value of 'BA'
- Kinetica display issues with some specific diacritics e.g. 'High comma centred'. For these diacritics the underlying data values is correct. This is a consequence of the MARC8 character set not containing all the characters available in Unicode (stored in UTF-8 format).

The understanding gained from developing the processes for exchanging records with Kinetica were applied to developing the exchange of records with other systems. By mid-2003, we were able to export files of records catalogued on Aleph at CAVAL in a format suitable for loading into the library systems of customers, and to import files of records produced by member libraries for items being transferred into the CARM Centre.

3. Data input issues

The initial challenge was to analyse and understand the issues. Once this point was reached it was then a matter of producing documentation and providing training for the cataloguing staff. As CAVAL's cataloguers are all experienced with Kinetica, this encompassed articulating the differences between Aleph and Kinetica client input conventions. It is hoped that in the future as Kinetica moves towards Unicode implementation this will become less of an issue.

The input order of diacritics when combining with other characters is different for Aleph and for Kinetica. The testing showed that the Aleph character conversion routines correctly export and import all diacritics CAVAL uses from Unicode to their MARC8 equivalents (where applicable) and back again, as long as the combining diacritics are entered in the correct sequence. If the diacritics are not entered as Aleph expects them, then there is a possibility that they may be incorrectly converted on export or import.

Testing was undertaken with Kinetica, which uses MARC8 encoding, using the Code Tables as described in Part 3 of http://www.loc.gov/marc/specifications/spechome.html .

4. Display issues

Once again, diagnosing the processes and understanding what was causing which problems was the initial challenge. Satisfactory display of diacritic characters was achieved using the fonts recommended by Ex Libris. However as some displays vary between Kinetica and Aleph, training and documentation was developed by CAVAL staff and provided for the cataloguers. Printing was also achieved after locating and then installing appropriate printer drivers.

Further development

Records created in CAVAL's Aleph system currently contain only romanised bibliographic records. Future developments will include the incorporation of native vernacular scripts. We expect to work with other libraries as they implement Unicode compliant systems, the goal being to import and export Unicode records. However, we will build on the lessons we have learnt during this initial implementation and will plan carefully for a phased approach.

Conclusion

Hindsight is always a wonderful thing and is especially the case when implementing a leading edge system. Although Aleph is in use in many countries around the world, and is successfully used by libraries in multi-language environments, CAVAL's requirement to catalogue in a large number of languages provided an additional challenge. Although the CAVAL implementation team began the process of working with a Unicode system with some naivety, by the time live operations began considerable expertise had been developed.

Once an understanding of the challenges was reached, a phased implementation process was agreed to and reviewed regularly. This included configuring the system, developing workflows, writing procedures and delivering staff training. The first critical decision was that data in Aleph would be stored internally in Unicode, but exported and imported using UTF-8. The second was that it was essential that CAVAL could import bibliographic records downloaded from Kinetica into Aleph, as well as export records from Aleph for loading into Kinetica. This ensures that CAVAL's data is reflected in the Australian National Bibliographic database. Assisting the CAVAL Cataloguing staff to adapt to the Aleph system and observing their excitement in using it has been a satisfying outcome for the Implementation Team.

A high degree of system interoperability is now enabled by the integration of Unicode into many components of the computer environment. The stage is set for further development at the application software and user interface levels in order to enable true multi-lingual access for the global community. Libraries and library systems must ensure that they are part of these developments.

References

Aliprand, J.M. 2000. The Unicode standard: its scope, design principles and prospects for international cataloguing. *Library Resources and Technical Services*, vol. 44, no. 3, pp. 160-167.

Chandrakar, R. 2002. Multi-script bibliographic database: an Indian perspective. *Online Information Review*, vol.26, no, 4, pp. 246-251.

CHASE: Character Set Standardisation. 1997. http://www.ddb.de/gabriel/projects/pages/cobra/chase.html Accessed 15 September 2003.

Cunningham, A. 1999. Multi-*lingual Unicode web page development.* Community Networking Conference: Engaging Regionalism, Ballarat Australia. http://members.ozemail.com.au/~andjc/papers/cn99.html Accessed 30 Sep 2003.

Erickson, J.C. 1997. Options for presentation of multi-lingual text: Use of the Unicode standard. *Library Hi Tech*, vol. 15, no. 3-4, pp. 172-188.

Felici, J. 2002. Unicode: the quiet revolution. *The Seybold Report*, vol.2, no.10, pp. 11-15.

Holmes, N. 2003. The problem with Unicode. *Computer*, vol.36, no.6, pp. 116-118.

Kass, James. *Does your browser support multi-language?* http://home.att.et/~jameskass/ Accessed 1 Sep 2003.

MARC 21 Specification. 2003. http://lcweb.loc.gov/marc/specifications/speccharucs.html Accessed 2 Sep 2003.

Tull, L. 2002. Library systems and Unicode: A review of the current state of development. *Information Technology and Libraries*, vol. 21, no. 4, pp. 181- 185.

Wells, J.C. 2000. Orthographic diacritics and multilingual computing. *Language problems and language planning*, vol. 24, no. 3, pp. 249-272.

Wood, A. 2003. Unicode and Multilingual Support in HTML, Forms, Web Browsers and Other Applications. *Alan Wood's Unicode Resources*, http://www.alanwood.net/unicode/ Accessed 1 Sep 2003.

Zhang, F.J. and Zeng, M. 1999. Multiscript Information Processing on Crossroads: Demands for Shifting from Diverse Character Code Sets to the Unicode Standard in Library Applications. *IFLA Journal*, vol.25, no.3, pp. 162-167.

The Unicode Standard: A Technical introduction. 2003. http://www.unicode.org/standard/principles.html. Accessed 1 Sep 2003.

Unicode: Pulling it all together to support multiple languages. 2003. IN PRESS. *Library Hi-Tech.*

What is Unicode? 2003. http://www.unicode.org/standard/WhatIsUnicode.html 2003.

Accessed 1 Sep 2003.

Why Unicode? 2000. *Library Systems,* Library Technology Reports, American Library Association, vol. 20, no. 10, pp. 79-80.

There is extensive documentation of fonts, code pages and character sets on the Microsoft web site http://www.microsoft.com/typography.

Acknowledgements

I would like to thank my CAVAL colleagues Lamis Sukkar, Eva Varga and Matt Wood for their professionalism, enthusiasm, dedication and persistence as we worked through the implementation process together.

Endnote

[i] Information about Kinetica and the Australian National Bibliographic Database can be found at <http://www.nla.gov.au/kinetica//aboutkinetica.html>