Librarian Spoken Here?

Elizabeth Beaudin, Technical Administrator OACIS Project
Yale University Library
25th Internationalization and Unicode Conference
Washington, DC, March/April 2004

Abstract:
OACIS, funded by the Department of Education via a TICFIA grant, is an electronic union list of Middle Eastern journals developed in an open source environment. ("Union list" as used by libraries is a unified listing of materials held distinctly or in common by a group of libraries.) Unicode appeared to be the clear-cut solution in a system where: 1) the primary languages are English and Arabic; 2) a common data structure manages bibliographical and holdings information, in this case: MARC (MAchine-Readable Cataloging) as defined by the Library of Congress; and 3) the web display remains browser and font independent. Unicode represents the solution, although not a straightforward one.

First, the Romanized script used by library management systems to display non-Western languages tests the idea of a simple Unicode solution. Data extracted from these systems must be correctly converted, i.e. translated from "Librarian", before being loaded into OACIS. Along with several Western languages, OACIS includes 19 languages from the Middle East: the top 3 are Arabic, Turkish, and Persian.

Then, from a business perspective, the data translation and loading schemes need to accommodate varying data layouts and be administered at one site. The project partners currently include 7 US universities using 3 different library management systems and 1 German university. Six potential Middle Eastern partners are now identified; only some with automated systems.

Finally, data displays must accommodate both LTR and RTL orientation. The user interface to the web application allows for different languages, including Arabic. More importantly, the system must be able to handle the Romanized script in use and to display vernacular text when it becomes available.

Translation and integration are key elements of the project. This presentation will describe how Unicode can resolve these challenges and offer system commonality for use today and growth tomorrow.

Presenter:
Elizabeth A. S. Beaudin received her Ph.D. from Yale University, writing her thesis on medieval love narratives from Muslim Spain. She has taught language and literature courses at Fairfield University and Yale. At the same time, she is a systems architect with 20 years of information technology experience, during which time she has developed both academic and administrative systems in a university environment. Currently, along with OACIS, her projects include copy-editing an advanced Arabic reader.

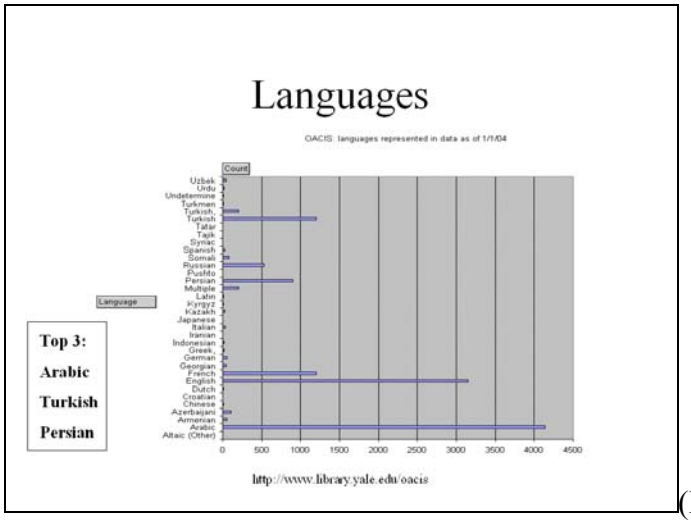
OACIS stands for Online Access to Consolidated Information on Serials. (http://www.library.yale.edu/oacis) This project, funded by the Department of Education via a TICFIA grant, is a publicly and freely accessible electronic union list of Middle Eastern journals developed in an open source environment. The term "union list" as used by libraries is a unified listing of materials held distinctly or in common by a group of libraries. Yale University Library, with over 11 million volumes, leads and coordinates a group of project partners from 7 other university libraries, 6 in the US and 1 in Halle, Germany. A number of additional libraries including two university libraries from the Middle East will join the project this year.

I came to this project with 20 years of IT experience, most of these at Yale. In addition, I had done my PhD at Yale in Medieval Spanish, so I had an insider's view of scholarship as well as an understanding of the IT evolution on campus. At the outset of the OACIS project, I thought that Unicode would be the straightforward solution for font display as well as data retrieval and storage issues in a system whose primary languages would be English and Arabic. But then I met the Library world from behind the stacks.

The project proposal stated that OACIS would include full bibliographical information and precise holdings (with owning libraries identified) for journals and serials related to the Middle East and published in numerous languages. Another essential element defined in the proposal was that OACIS would be accessible by all via the Internet. My first impression then was that I would need to develop a web enterprise solution including a database, a search engine, and interactive forms. This combination of features initially did not appear complex even when considering that the primary languages involved in the project would be English and Arabic. This linguistic element was a felicitous one for me, given my background in languages. Little did I realize at the outset that there was another language lurking in the background, i.e. in the data itself. I call this other language "Librarian".

Within the context of speaking "Librarian", I would like to present three points for discussion today: first, the pre-Unicode language set used by library management systems to display non-Western languages; second, from a business perspective, the coordination of dissimilar data layouts from partner institutions for accurate data translation and database loading; and third, methods for accommodating both LTR and RTL orientation in a browser independent web display.

To begin with, the OACIS prototype system, launched in November of 2003, currently holds data from six university libraries; today there are, in the bibliographical data, 25,000 title records of which 7900 are unique, and 45,000 holdings records. Since most partner libraries include differing countries in their regional definitions, the project partners decided to use the Library of Congress site for the Near East Section of the African and Middle Eastern Reading Room when determining countries and languages to include in the system. (See http://www.loc.gov/rr/amed/). Thus, the extract specifications for OACIS consider selection based on either 42 language or 46 country codes, such that OACIS now includes 34 languages from or about the Middle East: the top 3 non-Western are Arabic, Turkish, and Persian.



(PPT Slide 2)

The system, using an Open Source approach (PPT Slide 3), currently exists on a DELL Precision 350 with Intel Pentium 4 (2.26 GHz) with an architecture that includes

- Server operating system: Linux 2.4.18 via Red Hat 8.0
- Web server: Apache 2.0.4
- Database: MySQL 3.23
- Application languages: PHP 4.3.0, JavaScript 2.0, HTML

(For clarification I should note that Microsoft products could not address the nuances of the data involved and since we chose an Open Source approach from the outset, we wanted to avoid potential licensing issues with Microsoft and other proprietary vendors.)

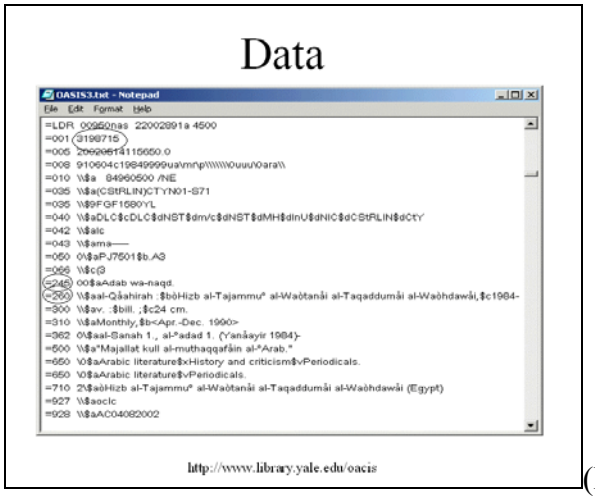
When designing the database structure, I choose to emulate that of MARC, as managed by the Library of Congress. MARC stands for **MA**chine-**R**eadable **C**ataloging and is the "format standard for the storage and exchange of bibliographic records and related information in machine-readable form".[1] The Library of Congress provides extensive documentation on MARC, which was of benefit in the design phase of OACIS.

MARC has a flat yet strong structure but is not suited to the relational style of a SQL database. Yet, since the OACIS system had to manage bibliographical and holdings information from library systems, I looked to existing MARC code tables to visualize and later organize a database structure. It was up to this point that learning to speak Librarian helped.

As an academic at Yale, I made daily use of Yale's online catalog and had often seen Arabic titles transliterated into English. So, I thought that these same titles could be easily imported into the OACIS structure I had developed. Yet, when displaying the titles of these serials in their own online catalogs, the university libraries depend on proprietary systems to manage the correct display of non-Western languages by way of Romanized scripts. So, when I received the first two data submissions, from Ohio State University and from Yale, I saw raw data in Romanized script that had to be translated before incorporating the data into OACIS, that is, *from Librarian into Unicode*.

[1] http://www.bl.uk/services/bibliographic/exchange.html

Here I would like to show a slide with incoming data as extracted from Yale's online catalog.



(PPT Slide 4)

Let me give you, at the same time, a quick tutorial in MARC. This is one record in the catalog for the journal entitled *Adab wa-naqd* (*Literature and Criticism*) published in Cairo. You will note that the record is made of several fields, as the librarians call each sub-record, with a prefix number to identify the function of the field. I'd like to point out two in particular: 245 and 260. The 245 field is the primary title entry. The 260 field indicates the place of publication. In this example, the title appears in the Romanized script used by libraries to represent the vernacular language in English characters. At first glance, then, the title presents no problem related to input into a new database. Data in, data out. But if we move to the 260 field, some of the challenges in translation appear.

You will note that the place of publication is listed as *al-Qåahirah*, which is Cairo in Arabic, but spelled, i.e. transliterated, in what I am calling Librarian. Thus, two characters are strung together (*åa*) to represent the Arabic letter (ا) *alif*. When I saw this I realized that I needed to rethink a few issues related to the database.

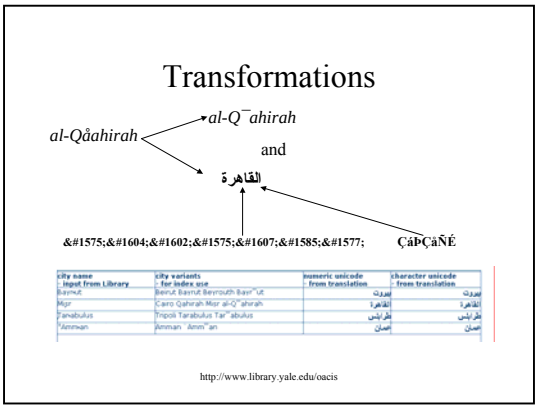
First, the fact that the library catalog did not include the place name *Cairo* in the record meant that I needed to expand the place of publication index in the database to include the Romanized spellings for all possible place names included in the data. Then I understood that by expanding the place names to include the equivalent Romanized script for Arabic names would not solve the entire problem since there could be an occasion when Cairo could be listed as *al-Qahirah* or as *Misr* (the version used in late 19th and early 20th century publications) or even as *Caire* (the French spelling). The Library of Congress Authorities helped supply a solution to this issue. From the Authority Headings Search site (http://authorities.loc.gov/), I was able to find alternate spellings for the major cities that would be in OACIS. With the help of one of Yale's very able work-study students, we were able to expand the place of publication index in the database to accommodate these spellings and provide robust searching possibilities.

[Side note: This issue spun into a feature of the system as well. Since our OACIS audience would be multi-national, I could incorporate the idea of alternate spellings into a multi-lingual web interface, which I will address later.]

Then came the challenge of translating from Librarian into Unicode. In the example, *al-Qåahirah* becomes ÇáÞÇåÑÉ, when one removes the prefix al-, as determined by library convention, and applies standard Unicode translation, thus *åa* to represent the letter (ا) *alif* works out to be *á*. Yet setting up the parsing routine was not as straightforward, since as you will notice *å* exists as one character coming in but represents another character coming out of the parsing schema. This overlapping of characters was not immediately apparent to me until I worked with larger datasets that included journals from Western countries, such as Spain and France, whose own languages included characters that Unicode used to represent others, for example, Tánger, where incoming *á* becomes outgoing *ä*.

I should make it clear here that it was not Unicode that presented the translation challenge; rather, the implementation by the partner universities of the Romanized transliteration schemes in their library management systems. Mind you, librarians did not invent these transliteration systems, but they were faithful implementers of them and over time cataloged using many versions of the Romanized scripts. For example, in one record Cairo was transliterated as in the example shown as *al-Qåahirah*, when in another the entry showed *al-Qahirah*. Both variations needed translation so that OACIS would produce what Librarians expected to see in the display of search results, that is, *al-Q¯ahirah*, using the macron to represent the letter (ا) *alif*, as well as produce the correct Arabic script as **القاهرة.**

It became clear that combinations of translation formulae would be needed, not only to produce deep indices to insure good searching but also to display search results that would respond to the two primary user audiences: the patron scholar and the librarian. Such that, an input value as seen here for *Mi˛sr* (old Cairo) becomes several different place names for searching and two different Unicode values for display: 1) using Unicode characters, here: ÇáÞÇåÑÉ and 2) using Unicode numeric values, here:
القاهرة



(PPT Slide 5)

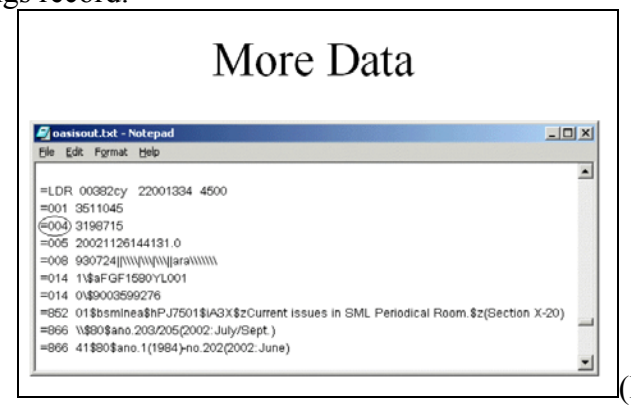
The same process used for place of publication was then applied to titles and subject headings, such that the basic search engine can search strings in these 3 items.

(PPT Slide 6)

[Side note: an Advanced Search engine will be added later this year which will allow for Boolean combinations on these 3 search items as well as the introduction of new search elements such as holding institution, country, and date ranges.]

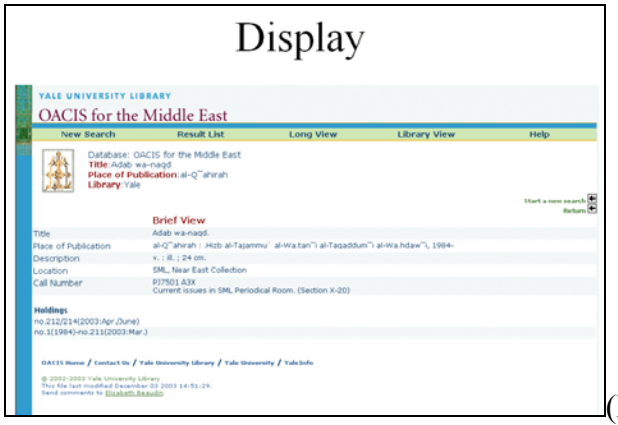
The second item for discussion emanates from a business perspective, that is, the data translation and loading schemes need to accommodate varying data layouts and be administered at one site. The project partners currently include 7 US universities using 3 different library management systems (Voyager, Innovative Systems, and NOTIS) and 1 German university (with an automated system based on PICA, a structure somewhat similar to MARC). Six potential Middle Eastern participating institutions are now identified; only some with automated systems. Two of these institutions will begin to participate this year.

If there were different entries for place of publication names, then inconsistency was also true in the implementation of MARC standards by the partner universities in the cataloging of bibliographical information and the related holdings data. Much like speaking a dialect of a language, here the partners spoke dialects of Librarian. Now for another MARC tutorial. The earlier slide showed one bibliographic record. Here is its corresponding holdings record.



(PPT Slide 7)

The 004 field, in this implementation from the Yale catalog, is used to link the holdings information to its bibliographic details. In this example, we can see where the current issues are found and which earlier issues are owned and available for scholarly use. After translation, that is, after the parsing routines and loading schemas are complete, this data appears as seen in this next slide.



(PPT Slide 8)

Yet, only 4 out of the 7 US partners adhere to this implementation. The others either imbed their holdings data in the bibliographical record by using local field numbers, normally reserved for local curators or selectors to note comments, or in a completely separate file, sometimes even in an Excel spreadsheet.

For project longevity and sustainability, it is necessary that all data manipulation be done at one site. Since Yale leads the project, the OACIS server physically lives at the library. With plans for mirror sites in Germany and in the Middle East, it is also important to have solid data maintenance and back-up procedures developed and managed at one location. For these reasons, the partner libraries participating today and the new institutions planning to take part in the future send their data to Yale.

The OACIS team at Yale has developed extract specifications (see our site at http://www.library.yale.edu/oacis/project/) for use by all participants. Even with this common starting point, the systems groups involved have reported the need to customize the specifications to meet specific library practices. The data currently arrives from the US libraries in MARC format. By using a utility called MarcBreaker, developed by Terry Reese of Oregon State University, this formatted data is broken down to exist as seen in the earlier slides.

It is at this point that the data translation and loading schemes must address specific practices and nuances in order to produce one unified set of data that takes advantage of Unicode standards. This is where any designer's original flow chart shows a little box that says "Process data for input" followed by another little box saying, "Load data". In reality though, and after several testing passes with data from partner libraries, my flow chart actually became a series of questions and bifurcating paths between these two simple steps. Such as, which library management system does the university use? Or, does the library management system already provide titles in the vernacular. If so, turn left. Do holdings exist as separate records? If so, turn right. And on and on the evolution goes. If the system design concern is how to incorporate scalability, the prevailing business question remains how to manage the need for merging data from different institutions, implicit in the creation of a union list.

One of our answers so far has been to ask all participants to supply data in MARC format. For our German partner, this will mean that the German systems group must first extract the pertinent dataset and then convert it into MARC format before sending it on. For new participants, this decision may imply quick or protracted entry into the OACIS system.

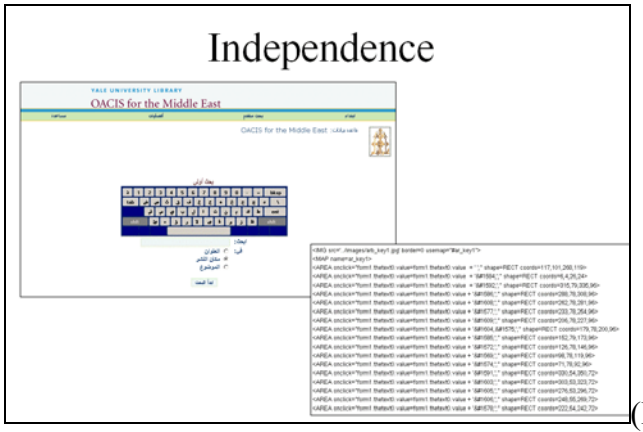
As of February 2004, we will begin adding new US participants based on their ability to provide data in MARC format that can be processed by existing parsing routines. Those datasets requiring further customization to the translation and input routines will be added to the system at later dates. One reason we came to this decision was that as the original partners were able to send their datasets we expanded the parsing formulae as new data problems arose. For example, one partner's library has more extensive holdings in Turkish than others. The earlier rules for managing Arabic transliterations did not correctly handle the Romanized script for Turkish titles. Once the Turkish titles were translated accurately, it appeared that the conversion routines had stabilized sufficiently to serve a larger audience.

Politics and distance complicate the business picture further. This year we will have 2 interns from Middle Eastern participating libraries. Due to prevailing geo-political difficulties, the introduction of interns – and their institutions' data – was delayed to Year 2 of the OACIS grant. As I mentioned at the beginning, this project is funded by a TICFIA grant, which stands for **T**echnological **I**nnovation and **C**ooperation for **F**oreign **I**nformation **A**ccess. Since the premise of a TICFIA grant is to make information from foreign sources available to US academics, it is essential for the life of our grant to introduce data from Middle Eastern institutions. To compensate for continuing political difficulties, the plan to develop interactive maintenance forms has received higher priority and has been modified. These web forms, to be tested and implemented early this year, will now allow data-entry by Middle Eastern participants. Data entered in this fashion will bypass 2 stumbling blocks: 1) data will not need translation from Librarian into Unicode, but instead allow for introduction of vernacular text directly into the MySQL database; and 2) Middle Eastern libraries can participate fully in the project from their home site whether they have automated systems or not and whether they adhere to US library cataloging practices or not.

Next, in a system where integration and flexibility are key and English and Arabic the dominant languages, data displays must accommodate both LTR and RTL orientation. As a by-product of needing deep search indices, the user interface to the web application allows for different language displays, including Arabic. In the opening search display, the menu bar entry OPTIONS currently allows the OACIS patron to choose a preferred language interface.

[Side note: Other options are planned for implementation this year, such as, a bookbag of selected search results – similar to the shopping cart approach on a commercial site; along with the ability to set preferences for search type, common search variables, etc.]

If Arabic is selected from the available languages, the search form changes to include an on-screen keyboard. This feature allows for browser and client platform independence. The user does not need to know whether Regional Settings are set to accommodate Arabic nor how to change keyboard settings from one language to another. The patron enters the desired search string that is then easily converted into Unicode numeric values to conduct the search by using simple Javascript behind a mapped image.



(PPT Slide 9)

Earlier I mentioned that the translation routines produce Unicode equivalent strings in both character and numeric forms. In the case of searching, as seen in this example, the numeric form of Unicode has proven more reliable. Yet, any text to be displayed as part of the web interface or the database indices is stored in Unicode character form since I found that it was more efficient to set up language tables and convert to Unicode using shareware found via the information provided on http://www.unicode.org . Nevertheless, for reasons of browser independence, the numeric Unicode method is the preferred technique.

Further, the OACIS system must also be able to handle the Romanized script in use today and to display vernacular text when it becomes available. For this reason, when a user selects Arabic as the interface language, RTL and LTR orientations are mixed when displaying search results in anticipation of a fully vernacular display. This is because titles are only available now in Western or transliterated form, as seen in this example:

(PPT Slide 10)

To produce this mixed output format, I have found that branching to different code based on POST and GET variables when processing PHP scripts is a more reliable method for painting the web screen than depending on HTML tags like <SPAN> whose application can vary from one browser to another.

[One last Side note: Once the vernacular is available, this display would be modified to show only the vernacular. The mixture of English and Arabic would continue only if the underlying record contains both languages.]

Finally, linguistic mixes, nuances, and varying implementations of standards – all these project factors make up what I call Librarian. The key elements of the OACIS project to date are translation and integration. As with any new system that must depend on outside data for its *materia prima*, translation – from one system structure or from one language to another – guides the system design process. Integration of the many variables into a unified language set produces a flexible product that can address the needs of a varied audience. Unicode is at the heart of the processes that build the OACIS system. Already, from OACIS demonstrations for our library colleagues, the curators of other regional collections have expressed interest in using OACIS as a template for other projects; union lists among these, since the lessons learned in the development of OACIS can be applied to other language groups. One colleague, for example, would like to apply the keyboard mapping technique to African languages such as Bassa Vah, Bete, Kpelle, Loma, and Mende.

As participating libraries move away from Romanized scripts to include vernacular text in their catalogs, the problems solved by integrating Unicode today will facilitate the display of full vernacular search results in OACIS in the future. And as libraries convert their cataloging systems to be Unicode-compliant, translation issues faced by OACIS today will fall away. New problems may still arise; for example, which Unicode standards will be used and how will these be implemented in the library management systems of the future? Yet, Unicode will continue to unify. To wit, on at least two occasions this year, I have heard disparate university groups (TICFIA grantees and IVY Voyager Users Group) express the desire to develop a Unicode font: this in response to the lack of a complete one, since universities need to accommodate so many languages, even Librarian.