Investigating Multilingual, Multi-script Support in Lucene/Solr Library Applications

6/3/2010

Yale University Library
Jeffrey Barnett, Daniel Lovins, Audrey Novak, Charles Riley, Keiko Suzuki

This study was made possible by an Arcadia Trust grant to the Yale University Library in support of preserving and making accessible cultural knowledge.

INVESTIGATING MULTILINGUAL, MULTI-SCRIPT SUPPORT IN LUCENE/SOLR LIBRARY APPLICATIONS

TABLE OF CONTENTS

Executive Summary 3
Problem Statement 8
Current Environment 8
    Non-Roman Scripts in MARC Records 8
    Classic Orbis – The Voyager ILS and Non-Roman Script Functionality 9
    Yufind and Non-Roman Script Functionality 11
Importance of Non-Roman Script Support 13
Available Technologies 18
Gaps in Functionality 26
Library Environmental Scan 29
Commercial Solutions 35
    The Products 36
    Commercial Summary 39
Recommendations and Estimate of Resource Requirements 39
Conclusion 43
Acknowledgements 45
References 45
Appendix A – Internationalization Guide 48
Appendix B – Testing Scripts 54
Appendix C – East Asian Language Search & Display Requirements for Library Databases, Interim Report 59

Executive Summary

Problem Statement:

Yale has developed over many years a highly-structured, high-quality multilingual catalog of bibliographic data. Almost 50% of the collection represents non-English materials in over 650 languages, and includes many different non-Roman scripts. Faculty, students, researchers, and staff would like to make full use of this original script content for resource discovery. While the underlying textual data are in place, effective indexing, retrieval and display functionality for the non-Roman script content is not available within our bibliographic discovery applications, Orbis and Yufind. Opportunities now exist in the Unicode, Lucene/Solr computing environment to bridge the functionality gap and achieve internationalization of the Yale Library catalog. While most parts of this study focus on the Yale environment, in the absence of other such studies it is hoped that the findings will be of interest to a much larger community.

Non-Roman Scripts in Catalog Records

Voyager-based Orbis, our current catalog of record, was converted to the Unicode character encoding standard in 2005, thus supporting display of non-Roman scripts in our local discovery interface, but still not providing the kind of language-sensitive indexing, retrieval and display that our users would like.

In 2008, Yale implemented a secondary catalog portal called "Yufind", which currently has similar shortcomings to Voyager with respect to non-Roman content. However, because Yufind is built with state-of-the-art Lucene/Solr software, language-specific text analyzers are now available that, if integrated into Yufind, would improve discovery of original script content.

User Requirements

Recent survey results suggest that faculty and students consider non-Roman script access to library materials important for their research. Meanwhile, due to changes in how metadata are created and shared around the world, users expect, and librarians are preparing themselves to adopt, a more networked globalized future in which resources are increasingly represented in native scripts exclusively. Yale should find a way to respond to this research need and prepare for this increasingly multilingual future.

In the face of more ubiquitous original scripts, however, it would be a mistake to abandon Romanization completely. While the practice of Romanization may have begun life as a substitute for original scripts, it is clear that today our users find it valuable even when the original scripts are also present. A 2010 survey of East Asia library users, administered as part of our current study, reported more than 80% of respondents disagreeing that original scripts should be used exclusively.

Current Challenges and Opportunities

Unicode implementation in conjunction with Lucene/Solr provides the necessary elements for a successful multilingual discovery system. However, current implementations attempt to combine only a small number of languages or analyze only a small number of records. What has not yet been resolved is the best structure for a service that will handle multiple languages and scripts simultaneously, while providing core shared services such as relevancy, sorting, and identification of similar items.

Other challenges include directionality (e.g., Hebrew and Arabic read from right to left), ligatures (e.g., as in Arabic and Cyrillic), character folding (e.g., conflation of traditional Chinese with their simplified counterparts), and localization (e.g., language of interface).

Notable Examples of Lucene/Solr Deployed in Libraries

Noteworthy efforts toward the internationalization of web discovery using Unicode with Lucene/Solr in libraries include Stanford's SearchWorks, the University of Virginia's Blacklight, Michigan's Mirlyn, UNESCO's World Digital Library, and the Smithsonian's Cross-Collection Search platform. Contributions have largely been in the form of improved user interfaces (e.g., original script and Romanized fields displayed in parallel) but some sites have also made inroads into multilingual (including non-Roman) text analysis.

Commercial Support Options

Google and Microsoft both provide language analysis and translation services that can improve the functionality of a Lucene/Solr implementation, though the ability to integrate them effectively in local systems is limited by license terms and available APIs. Lucid Imagination and Sematext provide high-level support and supplementary modules for Lucene/Solr. Sematext also provides a Morphological Analyzer for machine learning (and identification) of languages not previously indexed. Two other vendors to consider are Teragram and Basis Technology, which provide expertise in Natural Language Processors (NLP) and data mining.

Recommendations

The Yale Library expends over half a million dollars annually on the creation and maintenance of non-Roman script cataloging metadata. Chinese, Japanese, Korean, Hebrew (and Yiddish), Arabic (and Persian) make up the largest percentage of native scripts in Orbis, and we now also have records with Ethiopic, Thai, Greek, and Cyrillic script. Combined, the non-Roman script holdings represent the second largest percentage of non-English content in the library following closely behind German. With the globalization of cooperative cataloging resources such as OCLC WorldCat and vendor-supplied metadata, the number of scripts represented in Orbis will continue to grow.

The need for functionality that supports non-Roman scripts in our discovery applications is clear. Our survey of the CJK community of librarians, faculty and students indicates strong agreement (94%) that non-Roman scripts are desirable in discovery, results set and record displays. We are also very aware that the Yale bibliographic database serves a

global community of multilingual users. On a daily basis more searches are executed against Orbis from outside the Yale network than from within. Furthermore, as just a quick survey of peer institutions (Brown, Dartmouth, Harvard, Stanford and the University of Michigan) demonstrates, support for non-Roman scripts is now widely available in the discovery interfaces of major U.S. academic libraries with area studies collections (albeit not to the extent recommended in our report). Finally, the cataloging community appears to be moving away from Romanization, and if successful, this change will greatly increase the reliance on original script for discovery.

Given the wealth of non-Roman data in Yale's collection, the importance our faculty and students place on non-Roman data access points, the inability of our classic Orbis application to support a robust multi-script discovery environment, and the availability of building blocks within Lucene/Solr for implementing an internationalized discovery interface, we recommend that the Library take the following steps: (1) proceed to integrate the available Lucene/Solr text analyzers into Yufind; (2) re-index non-Roman script records with a custom SolrMarc function that supports display of original script and Romanized fields in parallel; (3) contract with Sematext to integrate their Language Detection, Multilingual Indexer, and DidYouMean (DYM) ReSearcher modules into Yufind (or other Lucene/Solr applications that expose Orbis data).

This solution would address the most pressing original script functional gaps identified by area collections librarians, faculty and researchers. All JACKPHY languages (Japanese, Arabic, Chinese, Persian, Hebrew and Yiddish) would be searchable. Accuracy of result sets would be substantially improved to the level of near state-of-the-art precision. Result set and record displays would include interlinear non-Roman script with Romanized text.

Additionally, active involvement in the development of cataloging and technology best practices to support non-Roman script content in multi-lingual library databases is recommended. More in-depth, local analysis focused on language specific needs and user requirement differences between faculty, graduate students and undergraduates is also proposed.

Conclusion

The Yale Library has one of the world's great collections of non-Roman script language materials. The Library catalog, whether expressed through Orbis, Yufind, or otherwise, is the single most important mechanism for finding and obtaining these materials. Centuries of professional cataloging leave us a rich legacy of high-quality metadata in multiple languages and scripts, but our discovery tools have not kept pace. In order to make proper use of these metadata we need to implement more sophisticated multilingual indexing and text analysis. We know that faculty, students, and librarians want us to provide accurate results for 'known item', phrase, and keyword searches. In the case of Chinese, Japanese, and Korean languages that contain many character variants, we need to provide consistent and precise results sets that take these variants into account. Moreover, original script should be displayed in parallel with Romanized script both in results lists and within individual records. Aids to searching like auto-suggestion and spelling corrections for non-Roman scripts are desirable but secondary to the

more basic discovery and display functions highlighted in this report. The state of development in Unicode, Lucene/Solr, and Solrmarc is such that we can now implement these recommended functions at relatively low cost.

Investigating Multilingual, Multi-script Support in Lucene/Solr Library Applications

"…for mine own part, it was Greek to me."

--Casca, Act I, scene ii, *The Tragedy of Julius Caesar*.

«Αυτά μου φαίνονται κινέζικα.»

--("These appear Chinese to me"), Greek idiom.

「呢啲喺雞腸呀」。

--("This looks like chicken intestines [i.e., English]"), Cantonese saying.

Problem Statement

Over the past several decades, Yale has invested heavily in a sophisticated online bibliographic database which includes holdings in over 650 languages that together represent approximately 50% of the library's collection. Across this entire database, more than 600,000 titles are cataloged with the original scripts that have been supported to varying degrees in library online systems since the mid-1980s, i.e., Arabic/Persian, Chinese, Japanese, Hebrew/Yiddish and Korean. Combined, the non-Roman script holdings represent the second largest percentage of non-English content in the library following closely behind German. Thousands of additional records with these non-Roman scripts are added every year. Faculty, students, researchers and staff would like to make full use of this original script content for discovery and while the underlying data are in place, effective indexing, retrieval and display functionality for the non-Roman script content is not available within our bibliographic discovery applications, Orbis and Yufind.

In this report we look at our current environment and examine the importance of improved support for non-Roman scripts given user requirements and cataloging trends. We conduct a gap analysis of available technologies and an environmental scan of sites and commercial applications that provide enhanced support for non-Roman scripts. Finally we identify non-Roman script enhancements that can be added to Yufind specifically and Lucene/Solr more generally, and we estimate the level-of-effort and resource requirements needed to close the gap and deliver indexing, retrieval, and display functionality for our non-Roman script content.

This six-month study was made possible by an Arcadia Trust grant to Yale University Library in support of preserving and making accessible cultural knowledge.

Current Environment

Non-Roman Scripts in MARC Records

From the 1870s until the advent of Machine Readable Cataloging (MARC) and the OCLC cataloging network in the 1960s, bibliographic records existed in the form of catalog

cards written or printed in various languages and scripts. In the 1970s the Yale Library began entering records directly online in the national cataloging utilities, i.e., the two major union catalogs maintained by the Research Libraries Group (RLG) and the Online Computer Library Center (OCLC), within which libraries could exchange records, and from which Yale could derive both electronic records and printed cards for its own local purposes. Due to the state of computer technology at the time, the utilities could only support characters from the ASCII[1] Extended Latin character set. This ASCII restriction forced libraries to develop highly structured "systematic Romanization" techniques to provide consistency (and in some cases reversibility) in representing non-Western titles. As RLG initiated Chinese, Japanese and Korean (CJK) script cataloging in 1983,[2] Yale began adding these and other JACKPHY (Japanese, Arabic, Chinese, Korean, Persian, Hebrew, and Yiddish) scripts via RLG's Research Libraries Information Network (RLIN), which made it possible to again include these scripts online in union catalog records and locally on printed cards derived from those union records. This solution was not ideal because all non-Roman scripts had to be entered in special tags (MARC 880 fields) and the local online system was unable to display these specially-tagged fields. Initially the contents of the 880 fields were encoded in distinct script-specific code tables, necessitating special identification tags (066) to specify which tables to use. Since our local online catalog was still ASCII-bound, non-Roman fields had to be stripped out as records were imported from the bibliographic utilities. Over the following two decades Yale continued to expand its pool of JACKPHY-enhanced records as well as retrospectively convert non-Roman script data from its earlier catalog cards in both OCLC WorldCat and the RLG Union Catalog.

Classic Orbis – The Voyager ILS and Non-Roman Script Functionality

As the Unicode Standard[3] emerged in the 1990s and was then incorporated into library software at the start of the next decade it became possible to add scripts beyond the JACKPHY repertoire as well as to load records with non-Roman script content directly into Yale's local MARC database, Orbis. With the upgrade of Orbis to Voyager with Unicode in 2005, Yale converted its local database to the Unicode standard. This upgrade improved the display (though not always discovery) of records, as original script fields were now paired with their Romanized counterparts as illustrated below.

[1] American Standard Code for Information Interchange (ASCII) http://en.wikipedia.org/wiki/ASCII

[2] See http://www.oclc.org/research/partnership/history.htm

[3] http://unicode.org/

Fig. 1. **Parallel fields in ORBIS (Hebrew and Romanized text).**

Additional work was needed for our Chinese, Japanese and Korean (CJK) records since there was inconsistency in how spaces had been used for word segmentation in RLG Union Catalog versus WorldCat. (In many languages and scripts, spaces are used to separate text strings into words for easier readability, but in Chinese and Japanese and in some cases, Korean, spaces are not used this way. Still, machine indexing requires some way of delimiting words and the RLG strategy had been to add spaces artificially.) This issue was later resolved as the RLG Union Catalog was normalized and absorbed into WorldCat and then in February 2009 when the Yale Library removed all spaces between Chinese, Japanese, and Korean words in its local MARC database.[4]

The MARC Standards Office approved Unicode in 1998 as an alternative to MARC encodings.[5] To facilitate database conversion and record exchange between converted and unconverted systems, however, the Library of Congress discouraged, and most system administrators chose to exclude, Unicode characters for which there were no MARC equivalents. As of 2007, however, the entire Unicode Universal Character Set (UCS) is considered valid in MARC records, with the exception of deprecated or Private Use Area code points.

Between August 2007 and January 2008 a Yale Library "Task Force to Consider Using Non-Latin Greek, Cyrillic, and Thai Scripts in Cataloging" met to study options for adding scripts beyond JACKPHY in the local catalog. These three scripts were chosen because they were seen as the least complicated to implement and least likely to include characters without backward-compatible MARC equivalents. Also, records including these scripts were already seeping into Orbis via WorldCat, and a decision was needed on whether to accept them or try to

[4] While native Korean does use spaces between phrases, not words, the Library of Congress and some other libraries continue to use spaces between words artificially in order to improve indexing and retrieval in systems with known limitations. (cf. Jeong, et al., 2009).

[5] See: MARC 21 Specifications for Record Structure, Character Sets, and Exchange Media. CHARACTER SETS AND ENCODING OPTIONS: Part 3 Unicode Encoding Environment December 2007. http://www.loc.gov/marc/specifications/speccharucs.html

strip out specific fields during copy cataloging. As this work demonstrates, non-Roman scripts, including those beyond the JACKPHY set, are a growing presence in our catalog, and implementation of any new discovery tool needs to take this increase into account.

With the Voyager Unicode implementation and the removal of CJK spaces, the underlying non-Roman data are now in place in Orbis and continue to grow and also expand beyond JACKPHY. These scripts are included in record displays in Orbis, but related functionality was and is still lacking. An Orbis search using a non-Roman script will not retrieve all matching records, will not correctly sort those records that it does find, and will not display the result set using the non-Roman script. More specifically, Orbis (and the underlying Voyager software) does not support "culturally-appropriate" sorting (e.g., the way different countries organize information in dictionaries and telephone books), inclusion of non-Roman scripts in result sets, and the ability to search a given word or phrase in simplified (e.g., 毛泽东) versus traditional (e.g., 毛澤東) Chinese characters, or old versus new Japanese characters (e.g., 毛澤東 vs. 毛沢東), without generating different sets of results.[6] Another problem is that the chosen character variants in MARC records, decided upon decades ago, may be different from those currently produced by generally available Input Method Editors (e.g., 江戶 versus 江戸) and may cause records to fail validation at load time or fail to collocate at query and indexing time. Additional non-Roman script searching problems in Orbis include the incorrect parsing of non-Roman quotation marks, the inability to skip the correct number of non-filing characters (e.g., a leading Hebrew-script article 'heh' (ה) or the Arabic 'al-' (ال) should be ignored in indexing, sorting, and retrieval), and the inability to send records by email without losing non-Roman content. Searching non-Roman script in Orbis is so unsuccessful that Area Collections staff consistently advise readers to search the Romanized text instead. Essentially non-Roman script content constitutes a type of hidden, hard-to-discover collection within Yale University Library.

Yufind and Non-Roman Script Functionality

While classic Orbis continues to be Yale's catalog of record, over the past few years the Library recently completed the production implementation of as an alternative discovery interface.

Currently Yufind maps all non-Roman MARC fields into a single undifferentiated field in its underlying Lucene/Solr index. This solution supports the display of non-Roman script data as blocks of text at the bottom of associated records (see below).

[6] Although this functionality was under development in 2001 when Yale licensed the Voyager software only Phase I support for non-Roman scripts, i.e., the conversion of the database to Unicode, was made available by the vendor.

Fig. 2. **Hebrew script display as currently enabled in Yufind.**

The solution, however, provides a less user-friendly display than classic Orbis as the content is unlabelled (e.g., title is not differentiated from author), and without a side-by-side presentation of original scripts paired with Romanization, pronunciation and meaning are less clear.

As with Voyager-based classic Orbis, searching non-Roman scripts in Yufind is not recommended. As currently configured, a search in Yufind using a non-Roman script will, like Orbis, not retrieve all matching records, will not correctly sort those records that it does find, and does not display the result set using the non-Roman script. Moreover, Yufind's relevancy-ranking algorithm introduces new complexities (if also new opportunities) in the discovery of Yale's multilingual holdings. For example, the addition of differentiated non-Roman script fields to the index and the assignment of relevance "weights" will need to be considered carefully.[7]

[7]Faceting is also problematic since there is not yet an authority structure through which non-Roman script headings can be collocated and thereby represented consistently for faceted navigation. A long-standing best practice in information retrieval is to provide users with browsable lists of authorized terms, along with the ability to redirect searches from variant terms (including, in the case of Anglo-American catalogs, non-Roman script variants) to the browsable list. Otherwise, one ends up with difficult-to-browse and incomplete lists.

Importance of Non-Roman Script Support

USER REQUIREMENTS

While Romanization can be viewed as a stop-gap measure (i.e., between the time MARC was introduced and when databases were able to move beyond ASCII), it turns out that it continues to add value for our patrons. Non-native readers often find it easier than trying to search or browse in native scripts. Even fluent readers find it useful. Users often come to the catalog with information derived from other sources (bibliographies, newspapers, scholarly articles) which do not always include the original scripts, and especially for CJK, these users may not know the native spellings until found in the catalog, which hence serves as a reference tool. Additionally, Romanization helps collocate words. For example, the Hebrew word ניקוד and נקוד both Romanize (and thus normalize) in systematic Romanization as "nikud". It is interesting to note that in a 2008 survey of Yale's East Asia Library faculty and staff, nearly 83% of respondents prefer to search Orbis in East Asian scripts rather than systematic Romanization; while at the same time more than 73% opposed having records display in Orbis with East Asian language scripts exclusively. Users often come to the catalog with information derived from other sources (bibliographies, newspapers, scholarly articles) which do not include the scripts involved, and especially for CJK, will therefore not be able to know the native spellings until found in the catalog, which hence serves as a reference tool.

To better understand user requirements for searching, indexing, sorting and display of non-Roman scripts and Romanized content within a bibliographic discovery application two data collection tools were employed. The first was a focus group with area collections staff from the Yale Library. The second was a convenience survey of the East Asian library research community. Findings from the focus group informed the survey questions which were selected in part to drill deeper into some of the issues uncovered during the focus group. The studies executed as part of this project are just the start of the little-explored analysis of usability within faceted discovery applications for multilingual, multi-script content. Collective action among the community of research libraries with substantial area collections to develop common user-driven functional requirements would be a beneficial outcome of this preliminary work. More refined assessment, particularly focused on specific language communities, is recommended.

Focus Group

On the afternoon of January 29, 2010, the project team met with a focus group including fifteen staff members from the Yale Library. Invitations to the focus group were sent to twenty-one staff members the majority of whom regularly work with materials that are represented in Orbis by Romanized and original scripts and who support cataloging and/or public services. Additional invitations were sent to information technology and usability staff. Joining the discussion were two consultants from peer institutions--Dr. Martin Heijdra, Chinese & Western Bibliographer and Head of Public Services at the East Asia Library of Princeton University, and Sarah Elman, Head of Technical Services at the C. V. Starr East Asian Library, Columbia University. Dr. Heijdra and Ms. Elman are active contributors within the East Asian library community. Their contributions include work toward making native script more accessible within library discovery interfaces.

Results of Focus Group Discussion

The primary finding from the focus group discussion is that across the languages represented (Arabic/Persian, Cyrillic, East Asian, Greek, Hebrew/Yiddish and Southeast Asian), clear priorities exist for improved functionality for indexing, searching, sorting and display of Romanized and original scripts. Opinions on how best to implement the improved functionality differed by language, and neither the library nor commercial environments provide any common best practices.

The ability to effectively search non-Roman scripts was considered essential and given that functionality, participants expressed a preference for:

- continued inclusion in bibliographic records of both Romanized and original script although the value of the Romanized content varied considerably depending on the language. For example, for fluent users, Romanized content was described as essential for Chinese and Japanese and a hindrance for Southeast Asian languages. Participants recognized that the Romanized content is the only means of access for users who do not read the script.
- display of both Romanized and original script in result lists and individual records. Participants expressed, however, varying lay-out preferences. Some attendees preferred an interlinear display in which the Romanized and non-Roman text alternated. Others favored toggling between Romanized and non-Roman representations, and some thought a side-by-side presentation was best.
- a solution to the split result sets that arise from searches for variant forms such as simplified and traditional Chinese, old and new Japanese kanji, Korean hangul and hanja, and Yiddish/Hebrew digraphs. Participants agreed that the result sets should be combined, but did not reach a consensus regarding the way in which combined record sets should be sorted, displayed or presented within facets.
- interface and fixed field facet localization, i.e., the ability to toggle the language of display for the interface instructions, labels and fixed field value facets (such as language and format) was viewed by participants as desirable. When compared with the ability to search and display original script content, however, localization was not considered a priority for implementation.

Survey

This summary is based on an online survey that was made available for two weeks between April 5 and April 20, 2010. It consisted of 16 questions (with 37 data fields) asking respondents about their preferences for search and display of East Asian languages in library databases. The survey was publicized via email to the targeted groups: mainly East Asia librarians and library staff, area studies faculty, students, researchers, and others from East Asia-related programs as well as more general library staff who need to use Chinese, Japanese, and Korean (CJK) or other non-Latin scripts languages in their work. For a more complete analysis of the survey, see Appendix C.

User Profile:

A total of 366 surveys were returned. Sixty-one percent of respondents were librarians or library staff, while 35% were library end-users: 16% were faculty, 11% were graduate students, and small numbers of undergraduate students (5%) , and researchers (3%).

Close to the half of respondents reported English as their first or primary language (44%). Twenty-four percent answered that their first or primary language was Chinese. Twenty percent reported Japanese as their primary language and 5% said it was Korean. Yet, if we include responses from those who identified themselves as having reading knowledge or proficiency in a given language, Chinese language users were 46% of 366 respondents, Japanese language users were 59% and Korean language users were 15% respectively. There were more faculty and graduate students than librarians among Japanese language users (23% and 16%, respectively), than Chinese (17% and 8%) or Korean (11% and 15%).

Specifically deserving of mention is the unexpectedly high rate of comments provided by faculty (51% of 59), researchers (46% of 13) and graduate students (44% of 41) which might suggest a strong interest in better integration of original scripts in discovery applications among this population sample.

Survey outcomes:

Four broad conclusions can be inferred from the survey results:

1. Users want to search both Romanized and original script content in bibliographic descriptions.
2. Users prefer to include both Romanized and original script content in display. The strongest preferences are for the inclusion of both forms in search result list and individual record views. The language/script for screen interface and facets is secondary.
3. The solution to the issue of character and script variations (i.e., simplified and traditional Chinese, old and new Japanese kanji, and Korean hangul and hanja) is also a priority.
4. The current alternative for relevancy ranking is not ideal for most respondents, but results were inconclusive about the best alternative sort order.

Original Scripts vs. Romanization:

- More than 80% of respondents disagree (including 40% who “strongly disagree”) that it is acceptable to display original script exclusively. Graduate students and faculty tend to disagree more strongly than library staff.
- Regarding the language and script of search queries, the results are a bit more complicated. More than half agree in some way with the preference for using original scripts over Romanization, while over a third disagrees. More faculty, and especially more graduate students tend to agree compared to library staff who is evenly divided on this issue.

Display Preferences for Romanized and Original Scripts:

- Similar preferences are expressed for both "search result list" and "record view." Over 80% of respondents prefer both scripts shown in parallel or side-by-side and that it is important for users to be able to modify display according to their preferences.
- Display preferences for "screen interface" and "facets" are similar, too. While about 60% prefer both scripts, about a quarter of respondents prefer Romanization/English only. Compared to the above two display preferences for "search result list" and "record view", these two peripheral displays are rated as less important.

Character & Script Variations:

These questions related to character and script variations addressed the problems presented by simplified and traditional Chinese, old and new Japanese kanji, and Korean hangul and hanja. Results were analyzed by the language proficiency of the users (for example, responses from the Chinese language users were analyzed for the question about simplified vs, traditional Chinese).

- All three language groups expressed similar preference for collocation of character and script variants. The majority prefer retrieving results in both 'variant scripts' (when there are two) regardless of script used in the initial query. While about 60% prefer giving higher relevance to an exact script match, about 30% are fine without this special treatment. Less than 10% support the idea that only an exact script match, which is the convention in current Orbis and Yufind, should be returned.

Sort Order:

- Only 18% think "relevancy ranking alone is fine" and a total of 70% prefer "relevancy" along with an alternative alphabet-type arrangement. More in depth research is required to identify preferred alternatives.
- "Unicode code point," the most common sorting alternative to relevancy ranking currently in use in library systems is considered adequate by only 3% of respondents. The traditional dictionary order for Chinese characters, "by radical/stroke order", is also not considered desirable, with only 4% approving. Although user comments suggest that respondents may not have understood the terms used or the question itself.
- Among the alphabetical sort orders, arranging each language's alphabetical equivalent order is more popular than by Latin-script alphabetical order.

Additional Findings and Recommendations:

- This survey resulted in a rich data sample that, with more in-depth analysis, may generate valuable information about preferences across user categories and languages, especially given the high volume of free-text comments, more than a third of respondents (37.4% of 366) added at least one comment for the questions about searching and display.
- This survey focused on the CJK community. Investigations among communities of other language users especially Arabic/Persian and Hebrew/Yiddish are desirable. More in depth surveys and focus group interviews with different user groups and different language groups should be also considered.
- New types of language functions and additional information, such as "spell-checking or suggesting alternative words", "tagging in CJK characters", "Links to book reviews

(both English and the language of the item)" and "auto-complete for CJK", do not seem to be highly desirable for this audience at this time. We need to investigate further how these features are perceived by JACKPHY library users.

CATALOGING BEST PRACTICE

Given the growing importance of global networks in research and education, catalogers continue to seek ways to enhance search and browse capabilities for original scripts and languages. A clear indication of this practice is the forthcoming replacement of the *Anglo-American Cataloging Rules* (AACR2) with the new *Resource Description and Access* (RDA, expected publication date, June, 2010). The absence of the term "Anglo-American" from RDA's title indicates even at first glance that the English language (along with Latin script) is no-longer as central as it was in AACR2. RDA is deeply informed by the International Federation of Library Associations' (IFLA) Functional Requirements of Bibliographic Records (FRBR) model and its Statement of International Cataloging Principles (ICP). Resource discovery is likely to take place increasingly in original languages and scripts. It is telling that the IFLA Cataloging Section's 2010 conference papers include the topic: "Multilingual Bibliographic Access: Promoting Universal Access".

Recent articles in the professional literature point in the same direction. For example, in her 2009 *Cataloging & Classification Quarterly* (*CCQ)* article "No More Romanizing: The Attempt to be Less Anglocentric in RDA", Michelle Seikel notes that *RDA* is explicit in its preference for original languages and scripts over Romanization. Daniel Lovins made a similar observation in his 2008 *Judaica Librarianship* article, "The Changing Landscape of Hebraica Cataloging", namely that the new code retains "original language and script when transcribing data elements, with substituted or added Romanized fields permitted, but only as an 'option'. This is a departure from current practice where Romanized fields, with or without paired original script counterparts, are mandatory." In her 2010 *CCQ* article, "A Study of Romanization Practice for Japanese Language Titles in OCLC WorldCat Records," Yoko Kudo points out, "As increasing numbers of libraries offer vernacular script-based access to non-Roman language resources, the role of Romanization for information retrieval may be less critical than it used to be." Bella Hass Weinberg, a leading authority on the subject, captured this shifting paradigm in her recent book chapter: "Cataloging in Non-Roman Scripts: From Radical to Mainstream Practice" (2008).

A draft report from the ALCTS Non-English Access Working Group on Romanization (Nov. 24. 2009), discusses the current state of non-Roman script access in U.S. or Anglo-American library catalogs. Model A (with 880s) and Model B (without 880s) are compared, with most working group members viewing Model B as the likely (if not always desired) future, but recommending deferral until systems and staffing issues are better understood and the newly-multi-script National Authority File is better integrated into local discovery applications. Largely due to the contributions of non-U.S. national libraries, WorldCat already contains a mix of Model A and Model B records.

The implication of the cataloging trend away from Romanization and toward Model B is that a growing number of records will end up in Yale's catalog without any systematic

Romanization. This trend will make support of non-Roman scripts even more urgent.[8] Yale needs to find a way to prepare for increased multilingual, multi-script content in our bibliographic database by improving precision and recall of this content in all of our discovery applications.

Available Technologies

As indicated in the brief discussion of our current Yufind environment, it is this service environment and more importantly its underlying search and indexing engine, Lucene/Solr,[9] that now makes it possible to implement improved handling of non-Roman script content.

Lucene first emerged as an open source search technology over ten years ago, and is considered one of the "stable" Apache projects. Because of the liberal Apache license conditions as well as the speed and language-neutral nature of the interface it has been widely adopted by both commercial and non-commercial users ranging from IBM to Netflix and NASA to UNESCO. More recently ILS vendors have begun incorporating Lucene search capability in their "next generation" products and services. For example, Serials Solutions' Summon hosted service is based on Lucene and Solr. ExLibris Primo uses the Lucene libraries, and the Fedora/DuraSpace repositories are indexed in Solr. Contributions from all these users – commercial and non-commercial – have made their way back into the Lucene community.

Although the underlying Lucene/Solr software is already Unicode-compliant and includes many internationalization features, it has not yet been fully optimized for the complex multilingual, multi-script environment of a research library catalog. In this section a brief history of language support technology is described followed by an overview of multilingual and multi-script features and examples. The section concludes with a gap analysis of Lucene, Solr, SolrMarc and VuFind.

LANGUAGE SUPPORT HISTORY

Early computers and computer records using a seven-bit code evolved from teletype applications and were capable of encoding only 128 distinct characters. After accounting for control characters and numeric values and operators, there were only enough code points for 94 printable characters (not all alphabetic). In order to represent more than one language, each was assigned a separate code table for mapping the available code points. Multiple languages could not be encoded by a single table, and typically computers and computer files operated against individual "native" tables for each of the country/languages of origin. When languages shared a common alphabet (e.g. ISO-Latin) such as most Western European languages, they could also share a common code table with a few spare "extra" characters, but when the alphabets had no common base, e.g., Chinese, Hebrew, Arabic, the practice of "Romanizing" a

[8] Note that while the identification and parsing issues for non-Roman scripts remain the same, they are compounded for model B where they may occur throughout the record and not just in one specific tag, i.e., the 880s.

[9] That is, the enterprise search and indexing engine that powers Yufind as well as large database-driven sites like CNET, Netflix, LinkedIn, whitehouse.gov, the World Digital Library, the Smithsonian Institution Research Information System (SIRIS), and the Internet Archive.

language would attempt to phonetically transliterate the non-Roman language in the Roman alphabet. In addition to introducing confusion for similar sounding native words, the transliteration removes the possibility to sort or otherwise order (search) the original script.

UNICODE

In the late 1980s when the Unicode project first began to address the problem of representing multiple written languages in a common electronic form it faced more than the task of converting ink on paper into bits on a wire, or pixels on a screen. Firstly, written language itself is an encoding of spoken language, and has evolved in different cultures to use different means to represent individual words, phrases, and even silences or inflections. Most languages also incorporate rules for ordering and sorting text. Frequently, alternate visual representations are allowed for the "same" character, word, or punctuation, which must be reconciled when making comparisons. Finally, since different rules apply to different parts of the Unicode code space,[10] means must be provided to determine which rules to apply where. Rules, algorithms, tools, and raw data are continually being developed to address all these aspects of language encoding, all the time expanding as new languages and conventions are addressed.

Non-Roman languages have all the idiosyncrasies above, in addition to variances in what makes up the most basic unit of the code, the "character". Roman ("ISO-Latin1") scripts use a twenty-six (plus or minus) letter alphabet to "sound-out" syllables made up of consonants and vowels. Other languages use up to thousands of unique characters to represent complete sounds, which may or may not also be complete words; still others have distinct characters for consonants only, and represent vowel sounds only as transitions within words. Rules exist for these languages too, but they make even more important the need be explicit about the language being used. Unicode provides a standardized base on which higher level analysis and display can be based.

MULTILINGUAL/MULTI-SCRIPT IMPLEMENTATIONS FEATURES AND EXAMPLES

With the introduction of Unicode it is possible to store and display hundreds of languages[11] in their native script. Library usage has incorporated Unicode in both MARC21 and UNIMARC. This enhancement easily supports simple display of non-Roman scripts, and at one level this is sufficient in its own right for a native script to be searched for exact matches of single strings. The ability to recognize near matches, or multiple word queries and to support truncation, wildcards and user-friendly displays, however, requires layers of higher level language support. These range from interface localization, through word boundary recognition, to search optimization. Characteristics of languages and scripts and further explanation of each of these higher level language support features follows. (For a more in-depth discussion of these topics, see Appendix A, Levels of Internationalization.)

[10] According to Unicode Technical Report 17 "The range of nonnegative integers used to map abstract characters". The code points within the space have properties and functions (such as spacing, fractional spacing, or non-spacing) defined in the Unicode Character Property Model (UCPM) appropriate to the script they encode.

[11] http://unicode.org/repos/cldr-tmp/trunk/diff/supplemental/languages_and_scripts.html

TYPES OF LANGUAGES

Languages can be classified according to many different principles; for purposes of this document the most pertinent are degree of inflection and by their morphological typology. Inflections in a language include the conjugations of verbs and the declensions of nouns; more or less regular patterns that build off of a root word. The high degree of inflection in languages like Arabic presents certain challenges for search functionality that can be met with tools like stemmers. Languages with little or no inflection, such as Chinese, bring challenges of their own, as stemming is less effective or impossible, and tailoring the search would rely more on context, dictionary lookups, and word boundary detection. Indo-European languages tend to be moderately inflected languages. Uralic and Altaic languages tend to be agglutinative, where long sets of affixes may be added to a root to form a word-phrase, and in these cases the root may be harder to identify.

TYPES OF SCRIPTS

The kinds of scripts that exist can be classified according to the way each character represents a phoneme (sound, for group of sounds), or semantic unit (word, or idea). Alphabets roughly follow a principle of one sound or small set of sounds per letter. An abjad, such as Arabic, is a script that allows for the representation of only or primarily the consonants; vowel markings being optional or absent. Abugidas and syllabaries, such as the Brahmic scripts, are based on representing syllables; logographic and logosyllabic systems, such as Chinese, are based on representing units of meaning, words or ideas.

Scripts typically adhere to one directionality – either left-to-right or right-to-left. One notable exception to this rule includes boustrophedon directionality, where the script's direction is alternated in each line reading down a page, first left-to-right then right-to-left, as has sometimes been used in archaic Greek lapidary and Luwian hieroglyphics. Another is vertical text layout for many of the logographic and syllabic scripts of East Asia in which the vertical lines themselves may be arranged from right-to-left, as in Japanese, or vice versa, as in Mongolian. One of the more common exceptions to see in everyday practice is switching of directionality within the same line, when there is a mixing of text or script that runs right-to-left with another text from a script that runs left-to-right. Complex scripts are those that exhibit combining behavior between characters, whether to form ligatures (e.g., in Arabic), composite characters, or to appear in a rearranged order relative to the order of pronunciation.

LOCALIZATION FEATURES AND EXAMPLE

Localizing software or a resource refers to the process of adapting it to a specific locale, taking into account the country of the user and preferences for language and script. It involves translation of key parts of the user interface, but can extend to considerations such as time zone, currency, date format and units of weight and measure. Basic "localization" of web interfaces, that is presenting the option to view interface options, labels and instructions in a choice of languages, is a problem almost completely solved by Unicode itself and the ability to create translations of simple menu items for navigation and display. A simple example, as illustrated below, is the provision of English as a second option for the National Library of Florence. The native interface—

Fig. 3. **Default (Italian) interface for the National Library of Florence.**

--includes a single link to an alternate page with translated lables and menus, and a link back to the first:

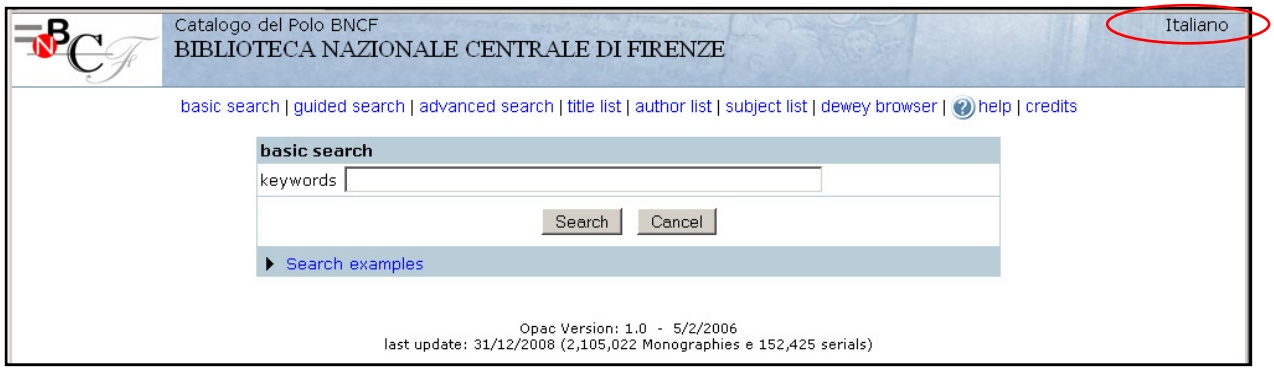

Fig. 4. **National Library of Florence interface, as localized into English.**

INTERNATIONALIZATION FEATURES AND EXAMPLE

After localization, the next challenge is the internationalization of content. Internationalization—often thought of not merely as a 'feature', but part of the 'architecture' of a resource—is a way of removing barriers to localization over a range of locales, involving conformance to international standards. Examples of features enabled through internationalization would include building in support for vertical or bidirectional text and allowing the resource to be more easily localized for scripts that use those features. Internationalization happens at many levels, from the character or glyph, to the recognition of punctuation and sort order.

One example of internationalization is the World Digital Library (WDL)[12]. The WDL, begun in 2006, is a joint project of UNESCO, the U.S. Library of Congress, and five partner institutions – the Bibliotheca Alexandrina, the National Library of Brazil, the National Library and Archives of Egypt, the National Library of Russia, and the Russian State Library – to develop and contribute content to prototype. It was launched publicly in 2009 with content from more that forty international institutions, including Yale.[13] The WDL site is built of the

[12] http://www.wdl.org/en/

[13] http://www.wdl.org/en/about/partners.html

same building blocks – Lucene, Solr and Solrmarc – and demonstrates both the potential and limitations of current technology.

The WDL provides indexed access to all metadata content in six UN languages and Brazilian Portuguese, but this functionality is accomplished by manual parallel translation and indexing in each separate language, rather than a common index containing the original texts and their translations. In addition to translating object titles and creators, and performing right-left conversion, the site also has language appropriate facet headings and content; however this facet translation is also a manual process.

The WDL represents an earlier implementation of desirable multilingual, multi-script functionality in a library discovery application. A more desirable and less resource intensive infrastructure would include automated processing, such as that offered by the Google translator[14] and support for a wider range of languages. Newly implemented language analysis tools in the most recent Lucene release will make it possible to introduce some of that desirable functionality. Nonetheless, regardless of how they achieved it, the WDL interface demonstrates many of the evolving internationalization features for library discovery tools. See for example:

Fig. 5. **World Digital Library interface localization (Chinese).**

[14] Detect web page language and translate to English

Fig. 6. **World Digital Library interface localization (Arabic).**

INDEXING (SEARCHING AND RETRIEVAL)

As a tool for information organization and discovery, one of the most powerful computer-aided techniques is indexing. In its simplest form, an index may consist of nothing more than a list that specifies what objects are registered for later reference. At the most complex, indexes may record properties, values, context, and attributes in greater detail and scope than the referenced objects themselves. It is important to recognize that an index and the objects it refers to are separate and distinct, just as records in a catalog are distinct from the items they describe. While brute force serial search (such as the "find" command in a web browser or text editor) is possible within single documents, or small collections, anything larger requires building an index. The Lucene library provides a number of building blocks for creating, populating, and searching text-based indexes. Because of its current world wide adoption, the libraries also include features uniquely adapted to identifying and supporting language specific characteristics.[15]

LUCENE/SOLR INDEXING BASICS

Lucene is a Java-based,[16] Unicode-compliant library for integrating index and search into applications,[17] which is important because Java and Unicode greatly simplify text

[15] For example, the Lucene Nutch web search tool has a language identifier plug-in that currently can recognize 14 different languages: Danish, Dutch, English, Finnish, French, German, Greek, Hungarian, Italian, Polish, Portuguese, Russian, Spanish, and Swedish.

[16] http://en.wikipedia.org/wiki/Java_%28software_platform%29

[17] http://wiki.apache.org/lucene-java/IndexingOtherLanguages

processing for non-Roman scripts. Lucene is a library of functions for building and evaluating formulas about documents.

Solr[18] and Solrmarc[19], both also implemented using the Java platform, add the capability of distributed cross platform web services (which is the preferred method of linking new services to existing utilities and data), and international standard metadata encoding to the task of representing, discovering and disseminating multilingual resources. Solr is a repository and access mechanism for the results of the Lucene manipulations.

Lucene refers to the objects being indexed as *documents*, for example a Netflix inventory record could be a document. The documents consist of one or more *terms* (meaningful distinct information units, such as quantities, locations, part numbers, words and parts of words) that are *analyzed* and normalized using a series of *filters* (custom programs that modify text sequences) to create index *fields* (sub-parts of the document with unique characteristics), and *facets* (sets of fields with common characteristics in multiple documents) that become the subject of subsequent *queries* (descriptions of what field values are of interest) about the indexed *documents*.

A query is essentially a formula consisting of term values and computed Boolean relationships that are evaluated against the terms and relationships in the index. Queries may also pass through a series of *filters* that massage the query according to pre-determined algorithms, similarly to the way that analyzers manipulate documents. In addition to exact matches, Lucene also supports relevancy, i.e., the ability to "score" the terms in a field or in a query on a numeric scale relative to all other objects in the index or in the query.

A *response* is an ordered series of data structures that lists the document ids for records containing the matched tokens in the order of highest to lowest score. In addition, the values of various component fields may be optionally stored on import and returned with the document ID in response to a query.

The combined manipulations of the document and query terms increase the likelihood of discovery. Here is an example:

Document analysis:

Document string (Title)	"Dances with wolves"
Tokenizer (white space)	<Dances> <with><wolves>
Filter (lowercase)	<dances><with><wolves>
Filter (stopwords)	<dances> <wolves>
Filter (stemming)	<danc><wolv>

Query analysis

Query string	"Wolf dancing"
Tokenzer (white space)	<Wolf><dancing>
Filter (lowercase)	<wolf><dancing>
Filter(synonym)	<wolv><danc>

[18] http://lucene.apache.org/solr/

[19] http://code.google.com/p/solrmarc/

The original document string ("Dances with wolves") and query string ("Wolf dancing") would not "match" in the conventional sense, but the analyzed tokens ("wolv," "danc") will (in any order). Note that analyzers are arranged sequences, each building on or adding to the analysis of its predecessor. Different analyzers are typically used for different content and for different fields. For example, "lowercase" is meaningless for date fields, and stemming clearly depends on the language of the text.

LUCENE INDEXING OF NON-ROMAN SCRIPTS

Early indexes, including early Lucene indexes were built on the assumption that texts were encoded as ASCII corresponding to the code table for the language, and in general the characters of the code table came from the Latin alphabet. As the means of encoding languages from other scripts emerged, the ability to build and use parsers, analyzers, filters, and fields based on those scripts has been added to Lucene and to the open source "contributions"[20] associated with it.

Conceptually, Lucene tokens correspond to words in the original text, but since the documents themselves can come from a variety of sources ranging from spreadsheets to novels, and may be combined in a single index, it is best to simply think of them as common "values" to be matched between documents[21], or between documents and queries. Different languages and in particular different scripts have different rules for encoding those values.[22] The basic technique for indexing texts of a given language is to assemble the filters and analyzers that understand those rules and map them to the common document-field-token structure used by Lucene. A number of such components exist, and the Lucene toolkit, together with the International Components for Unicode (ICU), simplifies building others as needed. Building analyzers and stemmers for a specific language has been widely established practice, and Web resources exist in the form of authority files, dictionaries, etc that can improve normalization. What has not been resolved, and what is very relevant to a complex multilingual, multi-script database such as Orbis, is the best structure for a service that will handle multiple languages and scripts simultaneously, while providing core shared services such as relevancy scoring and similar item detection.

[20] http://lucene.apache.org/java/2_9_0/lucene-contrib/index.html

[21] Note that a given token value only exists once per index, thereafter it is matched with all fields and documents containing that value. Thus indexes do not grow linearly with the number of documents indexed, but with the number of unique tokens in those documents. Obviously as more scripts are recognized, more unique tokens are recognized. For efficiency of recall, Lucene allows field values to be stored as they are indexed as well. Stored fields do cause an index to grow linearly.

[22] Simple examples include left-right vs right-left or top-down. Others include compound_word_parsing as in German and Scandinavian

MULTILINGUAL, MULTI-SCRIPT ENVIRONMENTS

Three broad approaches to dealing with multiple languages and scripts simultaneously have been suggested;

- One index for all languages,
- One index for all languages, with an extra language field so searches can be constrained to a particular language
- Separate indices for each language

Projects have been successful following each of the models, and we expect to consider all approaches going forward as their pros and cons emerge. For example, some languages might be best handled with a dedicated index, while others might be better suited to a common index.

Gaps in Functionality

As important as Unicode is in addressing multilingual texts, additional language awareness is necessary to accomplish acceptable search behavior. Among these behaviors are word stemming, stop word and synonym recognition, dictionary building, n-gram indexing and relevance ranking. Lucene and Solr use the Java Unicode support libraries to address these additional tasks. They are successful in handling large numbers of Roman scripts and are addressing growing numbers of non-Roman scripts as well. While they have been widely successful in commercial applications for single or dual language contexts, they face challenges in multilingual contexts where large numbers of languages are present simultaneously in indexes or queries. Following is a "Where they stand now" analysis of this environment.

UNICODE

Unicode is the de facto leader and flagship of multilingual text standard for processing. While new scripts are being added at a rapidly growing rate, the basic features of the encoding system have been stabilized, and will adequately address requirements presented by MARC content in major research library databases. We may find that in working with existing Unicode data under our control (our MARC records), that optimizing more of the Unicode-related functionality is desirable to enhance representation and access. The tools available from the Common Locale Data Repository (CLDR) or the International Components for Unicode (ICU) may prove useful to that end, but we are confident that Unicode addresses our needs, and we do not expect to make or propose any enhancements to Unicode.

LUCENE

Lucene has always had provisions for adding generic text analyzers and language specific features, but it has recently (in version 2.9.1) enhanced the configurability of these components, as well as bundling out-of-the-box solutions for Chinese, Arabic, and Persian. Free, but not bundled analyzers and stemmers for Korean and Japanese are also available. Work is also in progress to construct a generic "ICU analyzer" that would enable selective plug-in support for other Unicode scripts. These analyzers, though available, are not yet in use in our Yufind environment.

No mechanism currently exists, however, to analyze the native language of a text or query prior to indexing or searching in a Lucene environment. This gap is one of the significant

issues that would need to be addressed in a Lucene/Solr library discovery service. The open source and commercial communities are addressing this problem with varying degrees of intensity and success, but no clear best solution has emerged.[23] In language environments involving only two or three languages, analyzers may be statically assigned to different fields according to language, not content, but replicating fields for every possible language analysis is impractical.

SOLR

Solr is very closely linked to Lucene. They are both controlled by the same Apache Project team and Solr has already incorporated the Lucene 2.9.1 release, and the development teams are moving to a shared infrastructure to make future releases even more closely linked. In the next release Solr will include improved collocation (sorting and matching) support, also from Lucene, but in a separate package for improved configurability. As with Lucene, no explicit mechanism exists in Solr to determine appropriate language support in real time. Spellcheck, synonym and stopword support, services made available through the Solr layer, are all linked to single files, also, with no way to distinguish by script.[24] An appropriately identified synonym file could be the basis for matching variant forms in Chinese (simplified and traditional), Japanese (new and old kanji), and Korean (hangul and hanja), but only if it could be selectively applied.

SOLRMARC

Solrmarc takes advantage of the structured nature of MARC metadata and library classification systems to construct and populate customized schema for Solr indexes that enhance accessibility by using known fields to build facets that narrow and rank results of broader search results. Where language specific features are incorporated into MARC (such as fixed field values, 041 and 880 tags and indicators), Solrmarc makes these accessible in the Solr index as well. Note that Solrmarc does not affect query processing. As an example, there is no way currently to apply language mapping for queries, making such encoding less effective. Solrmarc recently unbundled itself from specific Solr versions, so that as improvements are made in Solr, they can be incorporated unchanged into a service based on Solrmarc. However, Solrmarc functions to identify links between 880 tags and their Romanized counterparts are not well developed and these are required in order to build analyzers specific to these fields alone, and thereby enable original script searching by subject, title, author, etc.

VUFIND

VuFind builds the library specific schema and properties which knit together Lucene, Solr, and Solrmarc along with real-time information and functions from a library ILS system into the user interface of the search engine. For index and search, VuFind incorporates all the language specific features of Solrmarc, and the components below it. VuFind also provides

[23] See the Java-Lucene email list for discussion of possible options (since 2005).

[24] A single file cannot include words that should be treated differently in different languages.

localization of its interface labels and instructions in eight languages, including Chinese[25], but has no means of translating the user interface of the back-end ILS (e.g. holdings locations or reserve loan terms), or of translating the fixed field derived facets such as language, format and geographic regions. Based on the current state of linked 880 fields in Solrmarc, VuFind does not yet fully support interlinear display of non-Roman script with the associated Romanized content outside the staff view of the record.[26] Similarly, since it cannot capture the 880 fields distinctly, they are only indexed under the "allfields" field, lumped with the otherwise all Romanized MARC tags, and as such cannot be separately analyzed. Once additional language features are available in Lucene, Solr, and Solrmarc, they must still be incorporated into VuFind before they will be visible to library patrons.

YUFIND

Yufind is Yale's customization and extension of the VuFind framework to the scale, depth and breadth required for an international research institution. Indexing, faceting, and field selection are all controlled through customizing the Solr configuration files. Field selection and text transformations are controlled by Solrmarc properties files and custom functions. Selection and format of ILS information and actions is customized through VuFind templates, style sheets and scripts. The interface localization feature is enabled for English, German, Spanish, French, Dutch, Portuguese, Japanese and Chinese. Lucene/Solr and Vufind features that would provide some non-Roman script indexing and retrieval functionality such as the Lucene language analyzers, and an available 880 tag index built with Solrmarc's "getLinkedFieldCombined" indexing, have yet to be integrated locally into Yufind.

SUMMARY OF GAPS

Not surprisingly, multilingual script handling is a challenging problem at all functional layers of a discovery system, but common threads emerge and lower level solutions tend to simplify the problems above them. The gaps that need to be addressed include: automated language identification, more robust text analysis and indexing, and interface localization. In a nutshell, non-Roman scripts in MARC 880 tags require special processing that the current Yufind instance does not execute. Fortunately, much industry and academic effort is already focused on the subject, and on improving related infrastructure[27] that, once integrated into our local environment, would fill these gaps. The tools exist to make the original script content usable, and the effort that is required is one of integration rather than original code development. Each of the improved infrastructure components now available must be integrated into the building blocks that together compose Yufind, i.e., into Lucene, Solr,

[25] The others are German, English, Spanish, French, Japanese, Dutch, and Portuguese. Localization is fully modular, which means that additional languages can be added, but only as translations for those languages are created and submitted (a possible role for YUL).

[26] The SolrMarc documentation (http://code.google.com/p/solrmarc/source/browse/trunk/docs/SolrMarc.doc, p. 16) describes an available pre-defined custom indexing routine called "getLinkedFieldCombined",.which is not yet implemented locally.

[27] TrebelClef Best Practices Portal http://trebleclef.eu/bestpractices.php ; Unicode Press release: http://www.unicode.org/press/pr-cldr1.8.html

SolrMarc, VuFind and our local Yufind interface templates. A convenient and favorable aspect to the well-structured nature of the required change is that it can be addressed in manageable incremental steps for both current and additional languages.

Library Environmental Scan

In this section we identify some of the leading multilingual, multi-script implementations of Lucene/Solr among research libraries and other cultural heritage institutions. Although no one implementation provides full multilingual, multi-script functionality each of these demonstrates significant advancement. One of the key challenges in such implementations is to identify the language of query terms, apply the appropriate tokenizers, stemmers, stop-word filters[28], etc., and then match the results against similarly-analyzed language-specific indexes.

Library applications use a variety of technologies to support the full variety of internationalization features (i.e., beyond just Lucene and Solr). For example, interlinear display of Romanized and non-Roman script as in SearchWorks, Mirlyn, and elsewhere, is supported by Java indexing routines in combination with scripting and templating code.

Fig. 7. **Interlinear display of Romanized and Japanese scripts.**

The ability to toggle language and script of facet values in the World Digital Library (WDL, see screenshot below) is accomplished largely through human-generated translation fields.[29]

[28] For example, the word 'the' is a stop-word in English but in French (as 'thé') means 'tea'. Obviously the term should be retained if the query language is French.

[29] An ideal (and more automated) way to toggle language and script, especially of controlled vocabulary terms, would use something like the WorldCat Identities and VIAF APIs. There are no Lucene/Solr instances that currently do this.

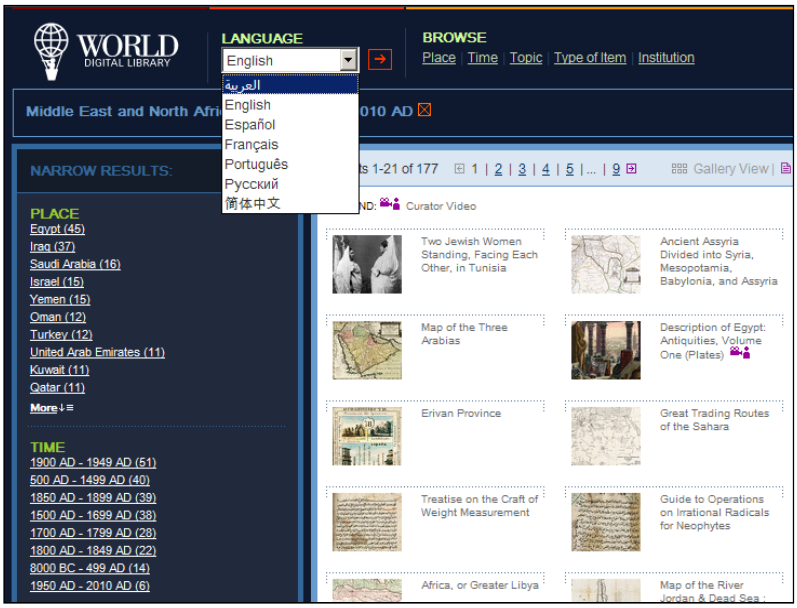

Fig. 8. **User control enabled in the WDL for toggling between localized interfaces.**

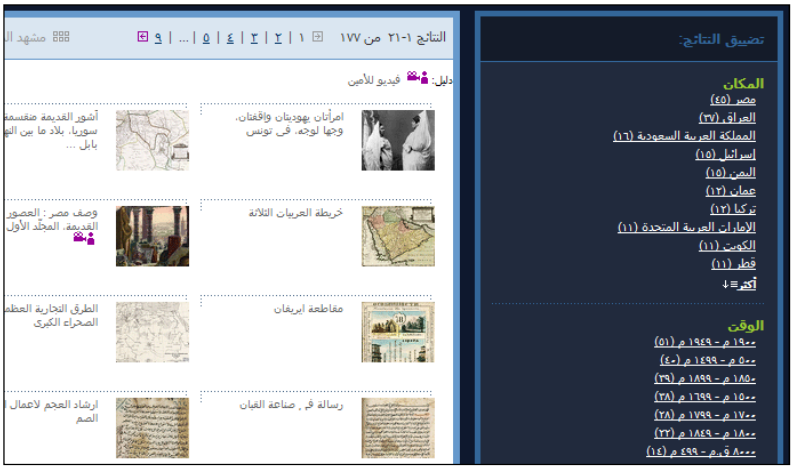

Fig. 9. **Right-aligned navigation panel for Arabic interface of the WDL.**

The WDL is arguably the most advanced non-Roman script Lucene/Solr implementation of its time, in the sense that it indexes and displays surrogates of various cultural objects (e.g., metadata and scanned images) in Arabic, Chinese, English, French, Russian, Spanish, and Portuguese. Localization includes not only translation of interface language but also every metadata element, facet, and facet value. The entire orientation of page elements (e.g., the navigation panel in the screenshots above) shifts when displaying right-to-left scripts. One of the developers pointed out that WDL can 'guess' the user's desired interface language based on incoming browser and system settings and the Chardet Universal Encoding Detector.[30] This, in turn, helps WDL present itself to patrons immediately in their preferred language.

[30]Per conversation with Dan Chudnov, Library of Congress, in code4lib IRC, March 2010.

The main challenge with WDL appears to be scalability. Some of the best features, e.g., all metadata elements represented in all seven languages, require intensive human intervention. For example, here are two versions of a WDL record that has been cataloged in seven languages:

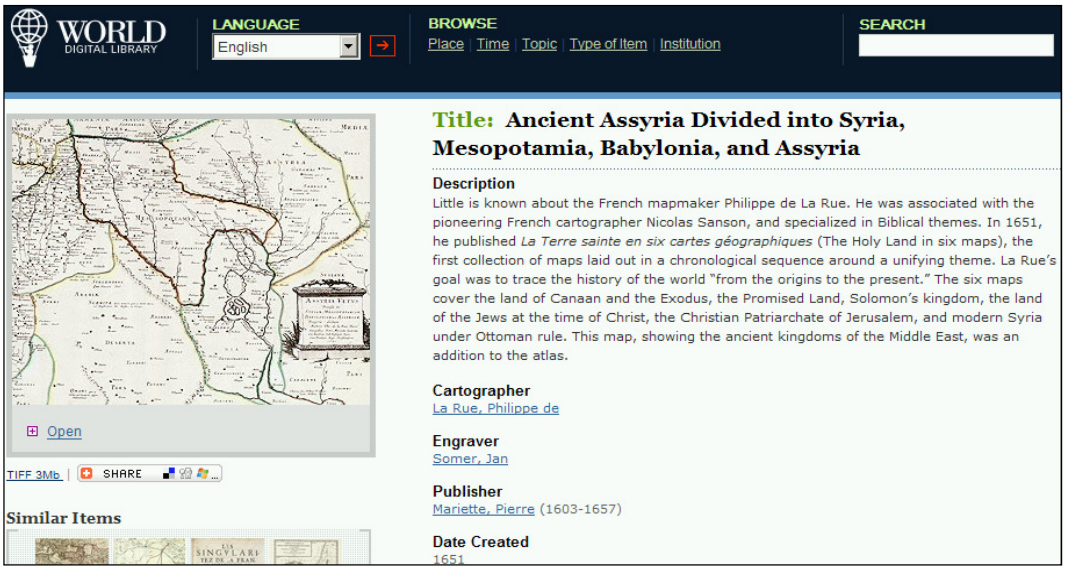

Fig. 10. **WDL record for Ancient Assyria, English interface.**

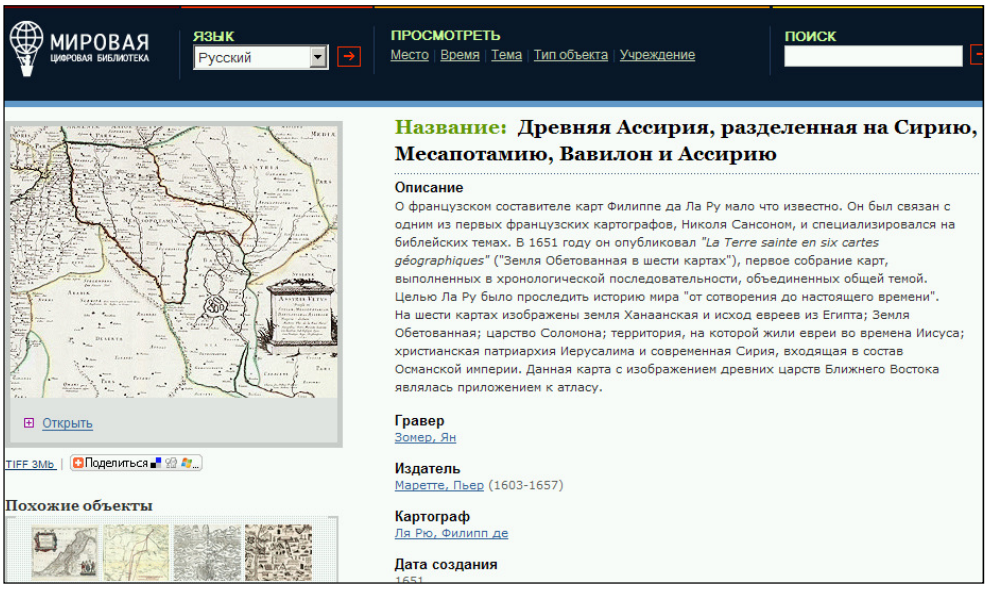

Fig. 11. **Same record for Ancient Assyria, with the interface toggled to Russian.**

In illustration of the scale problem, the WDL currently includes only 1,286 objects and seven languages, while the Yale Orbis catalog has over eight million objects in 650 languages.

Another example of an implementation with non-Roman script functionality is the Smithsonian Cross-Collection Search (http://collections.si.edu/search/): Note that the Romanized and non-Roman headings are interlinear in the record display.

Fig. 12. **Interlinear Romanized and Japanese data in a record in the Smithsonian Institution's SIRIS library catalog.**

Recent testing indicates, however, that there is no language-specific stemming or normalization across non-Roman-script character variants, which means that search results can be unpredictable and non-comprehensive (e.g., searching the same word in simplified vs. traditional Chinese characters retrieves different results). This is not a design flaw but simply a reflection of the fact that the developers focused initial efforts on integrating multiple databases and metadata schemes rather than on advanced non-Roman script functionality (cf. Wang, 2009).[31]

Among VuFind implementations, University of Michigan's Mirlyn (http://mirlyn.lib.umich.edu/) and the National Library of Australia (NLA) Catalog (http://catalogue.nla.gov.au/) stand out for their user-friendly displays of Romanization

[31] Another worthy site is the Internet Archive Universal Library (http://www.archive.org/details/universallibrary) but it proved difficult to obtain technical details on the Lucene/Solr search engine. Based on simple testing, it appears to index non-Roman scripts, but not apply any of the more language-specific text analysis.

alongside non-Roman text.[32] According to developer Tim Prettyman, Michigan's ALEPH™ ILS uses the concept of "fix (or expand) routines" which support "configurable" transformations at import time. When records are loaded into Aleph, 880 fields are converted into duplicates of their associated Roman-script tags (e.g., 100, 245), based on information in the subfield $6. They use another expand routine to copy the 008 language code to a subfield 9 within the converted 880 field to support field-specific, language-specific indexes. Michigan currently maintains separate indexes for Chinese, Japanese, and Korean. Because there were no field linking routines in SolrMarc at the time Mirlyn was being developed, Michigan removed SolrMarc's restriction against duplicate 245s and modified its Solr schema to accept multiple occurrences of otherwise non-repeatable fields.

As a Voyager library, NLA was not able to take advantage of ALEPH's export pre-processing to break up the blob of all 880 content into separately identified fields, which means they had to find other ways to link paired 880 fields, separate out different metadata elements (e.g., title vs. author vs. publisher), and provide interlinear displays. According to developer Mark Triggs, NLA did this by modifying the SolrMarc code to generate new index fields that look like this:

880_contents
880_dateSpan
880_era
880_geographic
880_isbn
880_new-title
880_old-title
880_physical
880_publishDate
880_series
880_subject
880_uniformTitle
880_us_author
880_us_author2
880_us_publisher
880_us_title
880_us_title2

Neither Michigan nor NLA has addressed the problem of normalizing facet values to a specific language, script, or authorized form (see, for example, http://tinyurl.com/yc8kkb5), which can lead to awkward and inefficient browsing, as can be seen in the presentation of facets below.

[32] These features are based on SolrMarc and PHP modifications

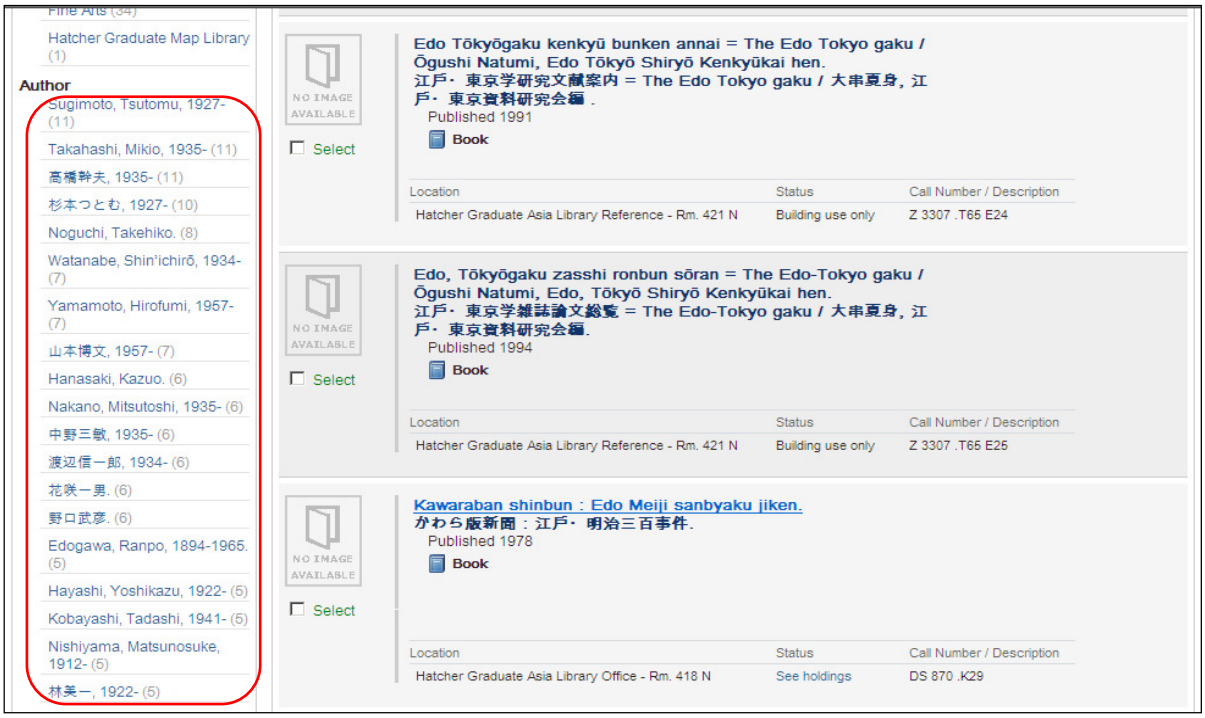

Fig. 13. **Facet values for the author field displayed in Romanized and non-Roman script, but without normalization, and not to satisfaction.**

Among Blacklight (also Lucene/Solr-based)[33] implementations, the University of Virginia Library's Virgo Beta (http://virgobeta.lib.virginia.edu/), and in particular, Stanford University Library's SearchWorks (http://searchworks.stanford.edu/), stand out for their integration of non-Roman scripts.

[33] http://www.lib.virginia.edu/digital/resndev/blacklight.html

Fig. 14. **SearchWorks display of non-Roman script in a search result set.**

In designing the architecture of SearchWorks, senior developer Naomi Dushay said she obtained "detailed problem definitions" from metadata experts working in Chinese, Japanese, Korean, and Hebrew. Recognizing that one cannot always infer script from the MARC 008 and 041 language codes, she is investigating script identification based on Unicode code-point ranges. She intends to use the n-gram tokenizer for Asian ideographs, with the understanding that the same analyzer chains will need to apply at both index time and query time. SearchWorks relies on default lexical sorting of Solr and Java, and has implemented a custom filter by Bob Haschart (the same one currently deployed in Yale's Lucene/Solr Finding Aids Database) that strips out diacritics and other non-sorting characters.

The Blacklight implementation at the University of Virginia, called "Virgo" uses the same custom-written CJK tokenizer that identifies characters by Unicode code point and splits them into single character tokens. According to Bob Haschart, "While this probably wouldn't make sense for searching an index where the bulk of the material is in CJK characters, in our implementation where the data is mostly Roman characters, this scheme seems to provide a good trade-off between searching power without needing to implement a CJK word boundary detector." This tokenizer is available for use in Yufind since it comes packaged with the SolrMarc indexer code that is shared by both VuFind and Blacklight.

Commercial Solutions

In line with the rapid international spread of internet commerce, demand for and supply of components for web services capable of providing interfaces in a range of international languages has grown as well. The software community has facilitated this by near universal adoption of the Unicode standard and its derivatives. Most internationalized software however focuses primarily on localization to one or two languages, with English a common second or third choice. Relatively few vendors explicitly identify their offerings as multilingual, i.e.,

capable of supporting several languages at once, and switching among them.[34] The following scan identifies those vendors only, and further narrows the problem to be addressed as indexing and search of multilingual structured text (i.e. not full text).

The Lucene/Solr program structure facilitates multilingual indexing because it separates the tokenizing step (recognizing word and sentence boundaries and stem properties) from the indexing and/or query steps (associating individual tokens with their parent document(s). Only the tokenizing step tends to be language dependent, and since tokenizing also removes sequence dependencies, right-left and left-right tokens can be intermingled without confusion. Document analysis in Lucene is also "pipelined" so the results of one analyzer can be made available to all its successors. In this way by putting a language (script) detection analyzer at the head of the stream it is possible to assure that only the appropriate analyzers for that script generate tokens.

Unlike relational database management system (RDBMS) databases such as Oracle, which supports Orbis, null tokens create no addition load or storage overhead in an index. Each of the Lucene compatible vendors make use of this capability in different ways and package the resulting product differently, but the result is the same.

The Products

Google[35] is included in this discussion because their core business is in fact search, and they are aggressively expanding web services in the international domain. They also provide a (high) benchmark in terms of performance and scope. At the same time, Google's technology is largely proprietary, and not easily integrated with other components of a library discovery interface (such as faceting or "more like this"). Conversely, Microsoft Bing[36], which has exceedingly rich tools for creating, searching and translation within individual multilingual documents, was not considered because it does not offer a convenient RESTful interface to its services, which makes it difficult to access from other web-based services like Yufind.

The remainder of the vendors, while offering varying degrees of openness in their license terms, all build on the Lucene/Solr analysis plug-in model and could be considered in mix-and-match combinations, depending on the needs of the library. All four have contributed portions of their work to the Apache/Lucene project, and three of the four were founded by project members.

Based on the recently upgraded language support in the ICU Java libraries, Lucid Imagination built and contributed back to Apache/Solr analyzers for CJK, Arabic, and Persian[37], and offers the whole as a "LucidWorks Certified Package", with technical support available in $1600/day bundles. Because open source projects tend to have multiple

[34] A conclusion shared by a recent MLIA Best Practice report from the Cross Language Evaluation Forum.

[35] http://translate.google.com/#

[36] http://www.microsofttranslator.com/Tools/

[37] A Thai analyzer sample is available separate from the bundle, as are third-party contributed analyzers for other languages.

interlocking dependencies, the availability of the package is advertised to reduce overall maintenance costs and improve performance.

In a similar model, Sematext offers stand-alone, one-time-priced software, with unlimited use and modification rights for language detection and multi-field indexing, also with $1600/day consulting. Both Lucid and Sematext have developed other stand alone analyzers, some of which are available from the project "contrib" site. The Sematext Morphological Analyzer will allow a site to train analyzers for any language for which a sufficient corpus exists.

The remaining two vendors have positioned themselves higher up the food chain as Natural Language Processors (NLP) and/or data mining technologies, and are priced accordingly with value added features such as entity extraction, which is useful in full text, but less so in structured (and tagged) text. Based directly on research for this project, Teragram is prepared to negotiate non-commercial licenses for the lower-level language detection and analysis of the languages its supports; a draft number is cited below. Basis Technology charges for its language support on a per-language basis, but might consider other terms for non-commercial use.

The commercial solutions for language detection are summarized in the table below.

Table 1. Commercial NLP index offerings, summary of functions and costs

Vendor	Product(s)	Function	Cost	Consulting fee	Open?
International Components for Unicode[38]	ICU4J	Java Libraries: Fully implements current standards Unicode collation, normalization, break iteration, script detect. Updated more frequently than Java Full CLDR Locale data	Free	n/a	Y
Lucid Imagination[39]	LucidWorks Certified Solr	Analysis for CJK + Arabic +Farsi	free	$200/hr ($1600 min.)	Y
Google translator[40]	Script detect/translate	REST interface (no analysis)	"Powered by"	n/a	N
Sematext[41]	MultiLingual Indexing Morphological Analyzer	Full Solr integration + import (no Hebrew)	$5995 ea.	$200/hr ($1600 min.)	OK to modify. No redistrib.
SAS – Teragram[42]	Teragram Language Identifier (TLI)	Partial Solr integration	~$xxK with "Powered by"	Inc. 1st yr. Then $5k/yr	N (single fee dev test/prod)
Basis Technology[43]	Rosette Linguistics Platform (RLP)	Full Solr integration + SDK	~$xxxK per language	Inc.	N (site license)

[38] http://site.icu-project.org/home The International Component for Unicode for Java (ICU4J) is a mature, portable set of Java libraries for Unicode support, software internationalization and globalization. Technically, not a commercial vendor, but major components have been contributed by IBM, Sun, Apple, and Google explicitly to encourage interoperable behavior.

[39] http://www.lucidimagination.com/

[40] Shown for reference only, not considered for Lucene based implementation.

[41] http://www.sematext.com/products

[42] http://www.sas.com/news/preleases/031708/acq.html

[43] http://www.basistech.com/about/

Commercial Summary

ICU4J is a relatively complete set of tools for Java language analysis (except for stemming), but has no bundled infrastructure to link to the web or to MARC. LucidWorks adds the Solr infrastructure and several key stemmers, as well a consulting advice and/or installation. Sematext offers more general tools, the ability to detect scripts based on real time analysis, and the ability to train analyzers to recognize and stem new languages. Teragram Language Identifier and the Rosette Language Platfrom from Basis Tech offer fuller-function Natural Language Processing (NLP) tool bundles than are needed for simple discovery, but also support more languages (e.g. Hebrew), and have indicated a willingness to investigate trimmed down versions for academic applications (but nothing yet modular and ready to deploy). They are normally packaged for "one-stop shoppers".

This is not to be taken as an exhaustive list of possibilities, but rather as a representative range of viable solutions.

Recommendations and Estimate of Resource Requirements

On June 16, 2001 the Yale Library signed a contract for a new library management system (LMS) that promised to "develop support for Chinese, Japanese, Korean, Hebrew, Arabic and other non-Roman character sets"[44] within 18 months. At that time, the Library considered support for non-Roman script so important that it formed a separate evaluation group dedicated to just this issue and included specific language and obligations about the required functions in the implementation timeline. Little did the evaluation committee know how prescient they were when they wrote, "We realize we will be going on faith in the development"[45], as now, a full decade later, we still do not provide more than the most basic discovery functionality for content described in original scripts.

Despite our current inability to make optimal use of original script content, we continue to create a robust multilingual, multi-script catalog of bibliographic descriptive metadata. The Yale Library expends over half a million dollars annually on the creation and maintenance of non-Roman script cataloging metadata. Chinese, Japanese, Korean, Hebrew (and Yiddish), Arabic (and Persian) make up the largest percentage of native scripts in Orbis, and we now also have records with Ethiopic, Thai, Greek, and Cyrillic script. With the globalization of cooperative cataloging resources such as OCLC WorldCat and vendor-supplied metadata, the number of scripts represented in Orbis will continue to grow.

The need for functionality that supports non-Roman scripts in our discovery applications is in no way diminishing. Our survey of the CJK community of librarians, faculty and students indicates strong agreement (94%) that non-Roman scripts are desirable in discovery, results set and record displays. We are also very aware that the Yale bibliographic database serves a global community of multilingual users. On a daily basis more searches are executed against Orbis from outside the Yale network than from within. Furthermore, as just a quick survey of peer institutions (Brown, Dartmouth, Harvard, Stanford and the University of

[44] Software License Contract, June 16, 2001.

[45] *Non-Roman Script Evaluation Workgroup Report,* Yale University Library, February 2001.

Michigan) demonstrates, support for non-Roman scripts is now widely available in the discovery interfaces provided by major US academic libraries with area collections materials (albeit not to the extent recommended in our report). Finally, the cataloging community appears to be moving away from Romanization, and if successful, this change will greatly increase the reliance on original script for discovery.[46]

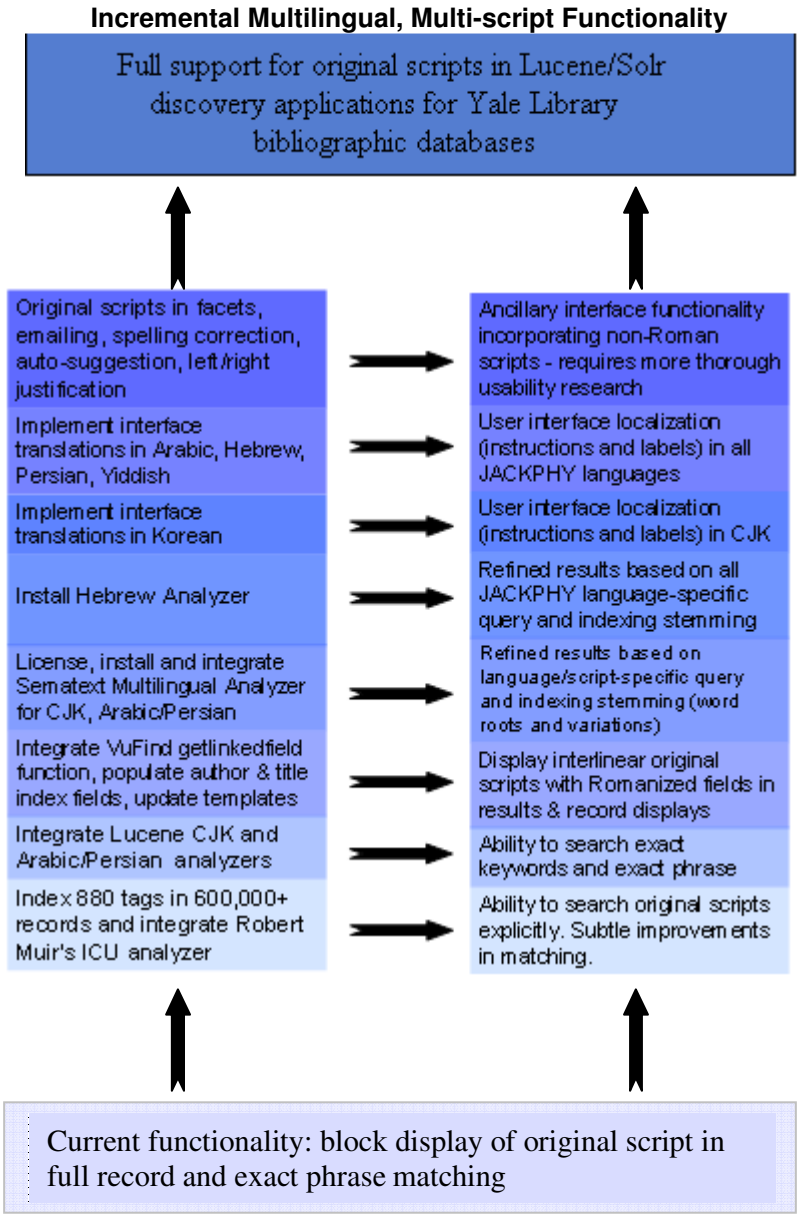

Fig. 15. **Outline of incremental steps toward refining multilingual support in the Yale Library catalog.**

[46] It should be noted that our survey results indicate strong continued interest in Romanization.

Fortunately, as was explained in more detail in the *Available Technologies* section of this report, the Unicode standard and wide adoption of Lucene and Solr in the commercial, academic and library environments puts multilingual/multi-script functionality very much within our reach. In fact, this infrastructure, which is made up of a suite of building blocks, opens the door to not one, but several, levels of non-Roman script functionality that can be implemented together or in stages in a series of shorter projects. (See *Incremental Multilingual, Multi-script Functionality* graphic on the previous page.) No matter what degree of effort is applied, real progress can be made toward at long last providing optimal access to the non-Roman content in the Yale Library bibliographic database.

Technical Recommendations

MINIMUM RECOMMENDATION:

At a minimum we recommend that the library integrate into the current Yufind environment (or any Lucene/Solr discovery application that is used with Orbis content) those functions that are already available in Lucene/Solr to support searching, indexing and display of original scripts. The current implementation of Yufind at Yale focused on optimizing the application for overall needs. Because this installation is now providing general discovery services in a production status, the necessary groundwork is complete and ready to support the requirements that would be introduced by original script functionality.

In broad strokes, the work required to deliver the minimum functions identified by users includes:

- Configuration of indexing schema for MARC tag 880 content (i.e., original scripts)
- Integration of the Lucene/Solr ICU, CJK, Arabic/Persian analyzers
- Integration of VuFind getlinkedfield function
- Extracting and reloading approximately 600,000 Orbis records (i.e., those with non-Roman scripts in MARC 880 tags)
- 880 content index builds
- Modification of result set and record displays

This work would result in:

- The ability to meet the high priority requirement to include interlinear original script and Romanized content in the result set and record displays within Yufind
- The ability to include a content facet in order to indicate for any result set the subset of records with original script content, and if requested, to limit results to that subset.
- Greatly improved query response. Searches for non-Roman script content would return results for keywords as well as known items.

Under this minimal scenario, stemming, i.e., the ability to return results based on word roots and variations, would not be available, nor would a solution for the issues related to simplified and traditional Chinese, old and new Japanese kanji, and Korean hangul and hanja.

This minimum solution would provide marginal improvement in search accuracy for Hebrew/Yiddish.

PREFERRED RECOMMENDATION:

To make full use of the Library's original script content, we recommend that the library integrate the existing Lucene/Solr analyzers as described above, and also that we contract with Sematext to license and also integrate their Language Detection, Multilingual Indexer, and DidYouMean (DYM) ReSearcher modules into Yufind (or any Lucene/Solr application applied against Orbis content). Sematext is the preferred vendor because their packages most closely match the scale and scope of our needs and because among the proprietary packages the terms of their license are the most flexible. In addition, the founder is an active Lucene committer and is likely to contribute work done in support of this project back to the Lucene community. A minor factor is also the convenience of working with a local, New York City vendor. The final choice of vendor, however, will need to occur after appropriate due diligence, but due to the open plug-in nature of the Lucene environment, if we are not satisfied with Sematext upon further investigation alternate vendors offer comparable functionality albeit with less flexibility and at greater expense.

The Sematext Analyzers would provide language detection, language-specific indexing, stemming and a solution for the problems presented by simplified and traditional Chinese, Japanese old and new kanji, and Korean hangul and hanja.

This solution would address the most pressing original script functional gaps identified by area collections librarians, faculty and researchers. All JACKPHY languages (Japanese, Arabic, Chinese, Persian, Hebrew and Yiddish) would be searchable. Accuracy of result sets would be substantially improved to the level of near state-of-the-art precision. Result set and record displays would include interlinear non-Roman script with Romanized text.

Additionally, given the diversity of languages spoken by the Yale community, more interface localizations (i.e., translations) should be added to Yufind as they become available in the VuFind core program. Making the existing VuFind translations available would be at zero cost. Yale Library staff with language expertise may also participate in the creation of additional translations,[47] particularly Hebrew and Arabic, for submission to the VuFind core.

ADDITIONAL RESEARCH RECOMMENDATIONS:

Results from the Area Collections staff focus group and CJK community survey about more advanced original script features suggest that the user interest in additional functional enhancements in support of non-Roman scripts is not fully understood. Before implementing features such as: alternate sort orders, interlinear facets, spelling suggestions, "more like this" suggestions, and right justification for right-to-left scripts, further research should be

[47] A translation is required for each of 230 terms used in the interface.

conducted. More in-depth analysis should also focus on language specific needs and user requirement differences between faculty, graduate students and undergraduates. In particular, a similar survey should be conducted with Arabic/Persian, Hebrew/Yiddish, Greek, Southeast Asian language and Cyrillic users. Studies of this nature will require the combined expertise of library usability and language specialists.

COLLABORATION RECOMMENDATIONS

CATALOGING:

Findings in the survey of the CJK community of librarians and researchers indicate a strong preference for including both Romanized and non-Roman original script content in bibliographic records. This finding challenges the current cataloging discussion that foresees a gradual decline in Romanization. Active engagement with the cataloging community to encourage a closer look at user preferences when working in the JACKPHY languages is highly recommended before cataloging policy is finalized.

TECHNICAL SOLUTIONS:

This six-month study identified a community of libraries that share a common interest in issues related to developing support for original script content within bibliographic discovery applications particularly those that are based on Lucene/Solr and SolrMarc. Collective action is highly recommended. Topics that would benefit from collaboration include: sharing results from user focus groups, surveys and usability studies; collectively developing and publishing functional requirements for how original scripts should behave and technical specifications for implementing those behaviors; collaborative code development and enhancement; fostering greater discussion and collaboration by actively creating opportunities for birds-of-a-feather/interest group gatherings at key meetings and conferences (e.g., Code4Lib).

Conclusion

The Yale Library has one of the world's great collections of non-Roman script language materials. The Library catalog, whether expressed through Orbis, Yufind or otherwise, is the single most important mechanism for finding and obtaining these materials. Faculty, students, researchers and staff want to be able to effectively search the Yale Library collection using original script. They want accurate result sets for known item, phrase and keyword searches, and, particularly for Chinese, Japanese and Korean that contain many character variants, they want consistent and precise result sets that include all variant forms. Additionally, original script should be displayed in parallel with Romanized content both in result lists and within individual records. The current state of development in Unicode, Lucene/Solr and Solrmarc has advanced to a level that makes providing the high-priority original script support features entirely feasible for a relatively modest cost. Such an implementation would bridge the functionality gap that currently exists in our local discovery application and achieve internationalization of the Yale Library Catalog.

Acknowledgements

This report was distributed in draft form to participants of the project focus group and to Yale Library cataloging and assessment experts. We wish to thank the following individuals for their comments:

Katie Bauer, Usability and Assessment Librarian, Yale University Library
Sarah Elman, Head of Technical Services at the C. V. Starr East Asian Library, Columbia University
Ellen Hammond, Curator, East Asian Library, Yale University Library
Martin Heijdra, Chinese & Western Bibliographer and Head of Public Services at the East Asia Library, Princeton University
Anthony Oddo, Greek Catalog Librarian, Yale University Library
Rich Richie, Curator, Southeast Asian Collection, Yale University Library
Joan Swanekamp, Chief Catalog Librarian, Yale University Library

References

[ALCTS] (2009). Non-English Access Working Group on Romanization: Draft Report (Nov. 24. 2009).

Chardet Universal Encoding Detector. http://chardet.feedparser.org/, visited 3/25/10.

Chudnov, Dan (2010). Private IRC discussion, 3/2/10.

Dushay, Naomi. (2010). Private email correspondence, 1/7/10.

Elman, Sarah (2005). "Voyager Unicode CJK Search and Display Problems." Internal Yale report, rev. November 28, 2005

Haschart, Robert. (2010). Thread "Implementers with non-roman script needs" in Blacklight Development discussion list, March 15, 2010 1:14 pm

Hatcher, Erik, Gospodnetić, Otis, and McCandless, Michael (2010) Lucene in Action, Second Edition, Manning Publications Co.

Heijdra, Martin. "IME variants not present in the EACC/MARC21 character sets". http://library.princeton.edu/projects/eacc/oldindex.htm Table 2: "Table for Duplicate Korean Readings: http://library.princeton.edu/projects/eacc/koreantable.htm

Internet Archive Universal Library http://www.archive.org/details/universallibrary

IFLA Cataloging Section. (2009). Statement of International Cataloging Principles. http://www.ifla.org/en/publications/statement-of-international-cataloguing-principles

IFLA Study Group. Functional requirements for bibliographic records. Munich: K.G. Saur, 1998. http://www.ifla.org/VII/s13/frbr/frbr.htm

Jeong, Wooseob, Kim, Joy, and Ku, Miree. (2009) "Spaces in Korean Bibliographic Records: To Be or Not to Be". *Cataloging & Classification Quarterly, 47*:708-721.

Kudo, Y. (2010). "A Study of Romanization Practice for Japanese Language Titles in OCLC WorldCat Records." *Cataloging & Classification Quarterly, 48*(4), 279-302. doi:10.1080/01639370903338352

MARC 21 Specifications for Record Structure, Character Sets, and Exchange Media. CHARACTER SETS AND ENCODING OPTIONS: Part 3 Unicode Encoding Environment December 2007. http://www.loc.gov/marc/specifications/speccharucs.html. Viewed 3/28/10.

Mirlyn (U. Michigan) http://mirlyn.lib.umich.edu/

Prettyman, Tim. Private email correspondence. March 16, 2010.

Saravanan, Udupa, and Kumaran (2010) "Crosslingual Information Retrieval System Enhanced with Transliteration Generation and Mining"

SearchWorks (Stanford) http://searchworks.stanford.edu/.

Seikel, Michelle (2009). "No More Romanizing: The Attempt to be Les Anglocentric in RDA". *Cataloging & Classification Quarterly,* 47:741-748.

Smiley, David (2009). Solr 1.4 Enterprise Search Server. Burmingham, U.K.: Packt Publishing

Smithsonian Collections Search Center http://collections.si.edu/search/

TrebelClef Best Practices Portal http://trebleclef.eu/bestpractices.php

Triggs, Mark. Private email correspondence, March 23, 2010.

Virgo (UVa) http://virgobeta.lib.virginia.edu/

Virtual International Authority File (VIAF) http://viaf.org/

Wang, Ching-hsien. (2009). A Collections Searching Center Using Lucene http://files.meetup.com/1446074/Lucene-Solr%20meetup%20CrossSearching.ppt

Weinberg, Bella Hass. (2008). "Cataloging in Non-Roman Scripts: From Radical to Mainstream Practice." in *Radical Cataloging: Essays at the Front*, ed. K. R. Roberto , 2008, pp. 28-39.

World Digital Library (UNESCO) http://www.wdl.org/

WorldCat Identities http://orlabs.oclc.org/Identities/

Appendix A – Internationalization Guide

Sections

1. User interface
    - 1.1.Localization vs. internationalization
    - 1.2.Locale data, translation, other general considerations
2. Character sets
    - 2.1.Language vs. script, character vs. glyph
    - 2.2. Types of scripts
    - 2.3. Types of languages
    - 2.4. Types of characters
    - 2.5. Glyph coverage, fonts, and rendering for proper display
3. Issues in search and retrieval
    - 3.1. Cross-script searching
    - 3.2. Sources of orthographic variation
    - 3.3. Limits of romanization
    - 3.4. Identifying sets of records by language and/or script

1. User interface

1.1. LOCALIZATION VS. INTERNATIONALIZATION

Localizing software or a resource generally refers to the process of adapting it to a specific locale, taking into account the country of the user and preferences for language and script. It usually involves translation of key parts of the user interface, but can extend to considerations such as time zone, currency, date format, and units of weight and measure.

Internationalization—often thought of not merely as a 'feature', but part of the 'architecture' of a resource—is a way of removing barriers to localization over a range of locales, involving conformance to international standards. Examples of features enabled through internationalization would include building in support for vertical or bidirectional text, allowing the resource to be more easily localized for scripts that use those features.

1.2 LOCALE DATA, TRANSLATION, AND OTHER GENERAL CONSIDERATIONS

There are standards in place that govern how a locale is described by the developer of a resource. In a webpage or an XML document, for example, the simple tag <en-us> identifies the locale by language (English) and country (US). Similarly, a resource can be tagged in more detailed ways to indicate, for example, the script used in the resource, as <zh-hant-tw> or <zh-hans-cn> for traditional and simplified Chinese in Taiwan and China, respectively. There are more levels of detail that can be indicated using locale tags, indicating dialects and broader or narrower geographic regions, but these examples suffice as an introduction.

Locale-specific data is collected through central registries, including the Language Subtag Registry and the Common Locale Data Repository, to make it easier to reliably

exchange and implement a basic level of support for upwards of 300 locales. Contributions to each registry are ongoing.

Within a given user interface, there are elements that may be localizable and others that will require no change between localized versions of the interface. Those that need to be localizable should not be hard-coded, but ideally kept in the code only as variables. This practice facilitates the process of supporting new locales, by allowing the developers to swap in a new set of translated strings for each variable as needed, rather than recoding each time another locale is to be supported.

2. Character sets

2.1. LANGUAGE VS. SCRIPT, CHARACTER VS. GLYPH

A script is a collection of characters used for the written representation of textual data, independent of language. A language may employ more than one script in its writing system, as for example with Japanese, which uses katakana, hiragana, kanji and romaji. Scripts may be, and often are, used by more than one language, and like language, often are influenced by each other and adapt over time. Scripts may be complex, incorporating use of diacritics or ligatures, or not; they may represent text through logographic, syllabic or alphabetic characters, or only as consonants with nothing to mark the position of vowels. They may or may not indicate variations in tone, vowel length, or contrast between characters that can have multiple meanings or pronunciations.

A character is a discrete unit of a script. For the most part, they are easily distinguishable from each other by design; the purpose of a character is to represent textual information. Due to borrowings between scripts, historical variation, differences in font design and formatting, differences between dialects and orthographies, and analytical interpretation, there is some fuzziness to the boundary of what constitutes a character in any specific case. Within a given encoding system or character set, decisions are made on where those boundaries lie, with varying degrees of consistency in adherence to stated principles.

A glyph is a visual representation of a character. The appearance of Latin letters can vary widely between fonts, so too can individual Tibetan, Arabic, Khmer, and Ethiopic characters. Glyphs are independent of characters. There are fonts that map glyphs that would belong to one character set onto machine-readable bytes of the characters of another set, in an effort to facilitate input for example, but this comes at the expense of the integrity of the stored data.

2.2. TYPES OF SCRIPTS

The kinds of scripts that exist can be classified according to the way each character represents a phoneme (sound, or group of sounds), or semantic unit (word, or idea). Alphabets roughly follow a principle of one sound or small set of sounds per letter. An abjad is a script that allows for the representation of only or primarily the consonants; vowel markings being optional or absent. Abugidas and syllabaries, such as the Brahmic scripts, are based on

representing syllables (such syllables may or may not be analyzable in smaller units); logographic and logosyllabic systems are based on representing units of meaning, words or ideas.

Scripts typically adhere to one directionality—either left-to-right or right-to-left. One notable exception to this rule include boustrophedon directionality, where the script's direction is alternated in each line reading down a page, first left-to-right then right-to-left, as has sometimes been used in archaic Greek lapidary and Luwian hieroglyphics. Another is vertical text layout for many of the logographic and syllabic scripts of East Asia, of which there also are different kinds, whether the vertical lines themselves are arranged from right-to-left, as in Japanese, or vice-versa, as in Mongolian. One of the more common exceptions to see in everyday practice is switching of directionality within the same line, when there is a mixing of text in a script that runs right-to-left with other text from a script that runs left-to-right.

Complex scripts are those that exhibit combining behavior between characters, whether to form ligatures or composite characters, or rearrange the order of written letters in comparison with the phonemic order. The technical standards and system requirements for these are often among the most demanding, but much progress has been made in recent years to improve support for them on most newer operating system platforms.

2.3. TYPES OF LANGUAGES

Languages can be classified according to many different principles; for purposes of this guide, the most pertinent are degree of inflection and by their morphological typology.

Inflections in a language include the conjugations of verbs and the declensions of nouns; more or less regular patterns that build off of a root word. The high degree of inflection in languages like Arabic presents certain challenges for search functionality that can be met with tools like stemmers. Languages with little or no inflection, such as Chinese, bring challenges of their own, as stemming is less effective or impossible, and tailoring the search would rely more on context, dictionary lookups, and word boundary detection. Indo-European languages tend to be moderately inflected languages. Uralic and Altaic languages tend to be agglutinative, where long sets of affixes may be added to a root to form a word-phrase, and in these cases the root may be harder to identify.

2.4. TYPES OF CHARACTERS

Characters as encoded for computational use are, in the most straightforward examples, alphanumeric: A-Z and 0-9. Likewise for any given script other than Latin, the basic elements may be encoded named as syllables, ideographs, and the like, along with punctuation marks. Many other kinds of characters exist that are used in everyday textual processing, and not all of them display. These include formatting characters, such as the Zero Width Space, the Zero Width Non-Joiner and the Combining Grapheme Joiner; control characters including Line Feed, End of Transmission, and Carriage Return, as well as characters used to select glyph variants, govern shaping behavior, or to mark the order given to bytes appearing in a stream of text.

2.5. GLYPH COVERAGE, FONTS, AND RENDERING FOR PROPER DISPLAY

The Unicode standard covers just over 100,000 characters in the latest version. In most font formats, there is a built-in limitation to the number of glyphs that any individual font can take up, effectively restricting each font to 65,536 characters or less. No single font is enough to provide comprehensive coverage of all the scripts encoded in the Unicode standard. Software that claims to be Unicode-compliant should be taking this factor into account, such that arbitrary restrictions on font management and character validation are removed, so long as the resulting data is also fully Unicode-compliant.

For valid data in a script to display correctly, a font supporting the character range for that script must be present on the user's machine, and the platform used should have an up-to-date version of the system's rendering engine that governs textual display. For newly encoded complex scripts, those with combining characters or contextual shaping, there is typically a delay between the time that the script encoding becomes available for use and the time that the rules for its shaping behavior are fully implemented on an operating system's rendering engine.

3. Issues in search and retrieval

3.1. INPUT

Multilingual users are becoming increasingly accustomed to inputting characters from the scripts used in their languages by adjusting the language settings in the operating system of their computers. By changing system settings, a 'regular' keyboard is all that is needed to serve for the input in the languages for which localized system support is available. For syllabaries, abjads, abugidas and logographic scripts, input can be through constructing a character one Latin key at a time, or sometimes through use of an Input Method Editor (IME). For alphabetic scripts and extended Latin, the setting change allows for virtual modifications of the existing hardware's keyboard layout.

Users and staff who are interested in searching or producing records in scripts with which they are not thoroughly familiar may find it useful to input via another method, such as online visual character pickers, or copying and pasting data from another source.

3.2. CROSS-SCRIPT SEARCHING

For languages and regions where multiple scripts are commonly in use, it is often necessary to repeat a search using a different script in order to retrieve the desired set of records. Fine-tuning the logic in a system to allow for search in multiple scripts may facilitate usability by reducing the number of times a user would have to repeat what is essentially the same search request.

3.3. SOURCES OF ORTHOGRAPHIC VARIATION

Often as a script's use crosses geographic, historical, linguistic, and even religious boundaries, variations in spelling are introduced that can complicate the picture for providing adequate search support. Rules governing cataloging practice may not always be sufficient in design or application to ensure an optimal level of access for the advanced user.

Some cases result from printing practices. In Latin script, often the absence of a diacritic is not a problem that affects access: there are rules for the cataloger to supply missing diacritics where it would normally appear, and even if the cataloger does not apply this rule, the search logic will still find the material, as the presence or absence of a combining diacritic in the data does not weigh heavily in retrieval.

In Arabic script, however, it is not uncommon to find materials printed in Egypt where the dots belonging above or below a given character were left out, with the intention of having them be supplied later by hand at the printer. In these cases as well, there are rules in place for the cataloging to compensate for the occasional absence of those markings, but not applying the rule will have greater consequences for retrieval than in the Latin script example: the markings in Arabic denote differences between two base characters that are encoded differently, not between a base character and a composite character that includes a diacritic.

Other sources of variation are a vestigial result of cataloging practice itself, due to restrictions on input and character validation in place before the migration from MARC-8 to UTF-8 (Unicode) encoded data, and a desire not to split an existing index or undergo a costly and time-consuming reconversion of past records. An example of this in Japanese is the use of one character under MARC-8 to represent either of two characters that were given separate encodings in Unicode, both carrying a value of 'hu' with a descending tone. A user is more likely to input the 'hu' character as it appears in the book he or she is looking for, although what is stored in the data in catalog records is only as one of the two similar-looking encoded characters.

3.4. ROMANIZATION: ITS UTILITY AND LIMITS

The longstanding practice of romanization has been one means of providing access points for the benefit of technical staff, students of languages using non-Latin scripts, and others. By now, it has shown some advantages even for native speakers of these languages, for example in Japanese, where there can be multiple pronunciations of kanji terms, and the user can benefit from a Latin-based sort order in the index to navigate Japanese records by their romanized access points.

Romanization also goes a long way toward fulfilling the needs of alphabetic scripts, facilitating input from a keyboard without requiring the user or staff to change system settings. In cases like Cyrillic and Greek, it can easily serve just as well as data in the original script would.

The same cannot be said for its utility with regard to abjads, abugidas, syllabaries, and to an extent Chinese and Korean, where there are more diverse arrays of methods by which these scripts have historically been rendered in Latin script. The users of languages using these scripts, when limited to Romanized-only access, are confronted with the challenge of figuring out just which methods and rules have been applied in cataloging the data they are looking for, and are not always aware of the existence of library-approved Romanization tables. Representations of vowel length, tonality, digraphs, and other linguistic features can vary widely, and compounding this effect may be the use within a printed source itself of a transliterated access point that varies from the Romanization method used in libraries. Such issues differ widely between scripts and languages; probably the best solution to deal with these issues is to provide technological assistance (suggestions, transparent conversions) at the time of user input.

There are many languages using non-Latin scripts for which Romanization tables have been developed and approved, others which await approval, and a fairly sizeable number for which no Romanization table has yet been proposed. Recent decisions in cataloging policy have made it possible to extend the practice of including original scripts into a record, in principle, for any valid Unicode script data; i.e., not only those that were encoded in MARC-8. Implementations making use of that decision have started to happen on a limited basis for a small number of scripts, whether through OCLC or at the level of individual libraries.

3.5. IDENTIFYING SETS OF RECORDS BY LANGUAGE AND/OR SCRIPT

There are several parts of a MARC record that can be used, by staff or users, to describe or determine the language of a work. Fixed fields, subject headings, the 041 field and the 546 field can all contain data about language. Fixed field values alone might be considered sufficient, were it not for the fact that only one value can be assigned to a given work, and the list of available values includes many ‘catchall’ categories for broad groups of languages. The 041 field allows the cataloger to specify the language that a work was translated from and into as well.

While MARC-8 character encoding was used, records that included data in scripts other than Latin were required to provide an 066 field that indicated which other scripts were used. That requirement has since been dropped. To the extent that original scripts are incorporated into a record, detecting the Unicode range to which those characters belong is the only means by which it would be possible to determine the script that a work was produced in.

Appendix B – Testing Scripts

The following questions are meant to identify types of functionality. They do necessarily assume that the functionality would be desirable in a specific system implementation. For example, while we want to know if a system can map Simplified Chinese and Traditional Chinese to the same indexing and searching targets, this is not the same as recommending a specific implementation should do this. Local usability testing among likely end-users is necessary to establish whether certain features should be implemented in a particular case.

ALL SCRIPTS

1. Are there any characters that do not display or display as empty boxes, question marks, something like this: "□" where there should be a character?

2. When retrieving a record
    2.1. Does the system display both non-roman and the paired roman fields?
    2.2. Are the paired fields co-located or displayed as separate blocks of text?

3. Without losing original script data in transmission, can you tag, cite, and e-mail yourself a record

4. Does system provide auto-complete for original script searches?

5. Does system provide spelling suggestions or alternative word choices for original script searches?

6. Does system allow user to specify language of interface (e.g., facets, instructions, labels)?

7. Does system allow user to specify language/script of facet values?

8. Facet display
    8.1. Do personal name facets display in Romanized form?
    8.2. In original scripts?
    8.3. Mixed?
    8.4. Do they appear in their native sort order (e.g., by alphabet letter, syllable, character radical)?
    8.5. Are they controlled by an authority file?
    8.6. Are comprehensive results returned for Romanized names?
    8.7. Are comprehensive results returned for original script names? If system returns "mixed" or "original scripts" only, are all relevant records retrieved or only a subset?

8.8.Example of mixed facet:

CJK (CHINESE, JAPANESE, KOREAN): SEARCH AND RETRIEVAL

1. Are result sets different when searching same terms in traditional vs. simplified vs. other variants of Chinese characters?

   Example: "Mao Zedong"
   毛泽东 Simplified
   毛澤東 Traditional

   Example: "Library"
   图书馆 Simplified
   圖書館 Traditional

   Example: "Economics"
   经济 Simplified
   經濟 Traditional

   Example: "History"
   历史 Simplified
   歷史 Traditional

   Example: "Art"
   艺术 Simplified
   藝術 Traditional

2. Are result sets different when searching same terms in old vs. new Japanese kanji?

Example: "University"

大学　New kanji
大學　Old kanji

Example: "Art"

芸術　New kanji
藝術　Old kanji

Example: "Buddhism"

仏教　New kanji
佛教　Old kanji

3. Are result sets different when searching same terms in hangul vs. hanja?

Example: "Korea/Korean"

한국　Hangul
韓國　Hanja

Example: "Economics"

경제　Hangul
經濟　Hanja

4. Is the same visual hanja character retrieved if inputted differently, resulting in different Unicode values?

Example:

An input of 이 yields 李, encoded at U+F9E1.
An input of 리 yields 李, encoded at U+674E.

5. Space and Punctuation
    5.1. Are result sets different with or without quotation marks?
    5.2. With or without spaces?
    5.3. Is this behavior consistent across scripts (hiragana, katakana, kanji, Hangul, Hanja, and traditional & simplified Chinese)?

6. IME variant characters: input "Edo bungaku" in standard Japanese IME "江戸文学" vs. MARC-8 equivalent "江戶文学". Are results the same?

HAPY SCRIPTS (HEBREW, ARABIC, PERSIAN, YIDDISH): SEARCH & RETRIEVAL

1. Are result sets different when searching with and without ‘ayn’ (‘) character?

2. Is the correct alif character, U+02BE, supported?

3. Are result sets different when searching when searching with and without ‘hamza’ (’) character?

4. Are result sets different when searching for terms with or without the prefixed article ‘al-’ (ال)?

5. Are result sets different when searching terms with or without alef (ا) or alef maksura (ى), Farsi yeh (U+06CC) ی, Arabic yeh (U+064A) ي ?

6. Are result sets different when searching terms with or without (ه), heh goal (ہ), Kurdish e (ە), teh marbuta (ة), and teh marbuta goal (ۃ)?

7. Are searches for records containing the Hebrew/Yiddish articles ‘a’, ‘di’, and ‘heh’ being indexed and retrieved properly?

8. Digraphs:

   וויִרטשאפט vs. ווירטשאפט

   On the left, the Yiddish word “Virtshaft” is entered with two separate vavs (i.e., key stroke ‘u’ in Microsoft’s Hebrew IME): U05D5 + U05D5; while on the right, a single double-vav digraph is used (U05F0). Does a query on each from of the word yield the same set of results?

HAPY: DISPLAY

1. Bi-directional
   - 1.1. Are bi-directional control characters recognized and functioning?
   - 1.2. For lines containing mixed HAPY and Latin script, are there issues with proper handling of the direction of the display?
   - 1.3. Are HAPY lines of text displaying with correct directionality (right to left) and justification (right side)?

GREEK, CYRILLIC, EXTENDED LATIN: SEARCH & RETRIEVAL

These scripts go beyond the JACKPHY scope, but are logical next steps for additional original scripts.

1. Is the Turkish dotless i (ı) searchable from Basic Latin i?

2. Are other Extended Latin characters searchable from corresponding characters in Basic Latin?

3. Are Cyrillic terms, where present in the record, searchable from Basic Latin equivalents, or vice versa?

4. Are searches for terms containing Greek characters α, β, etc. posing any noticeable problems? Are Greek terms searchable from Latin (e.g., α = a; b = β), or vice versa?

GREEK, CYRILLIC, EXTENDED LATIN: DISPLAY

1. Are records, such as those in Russian, Vietnamese, and Yoruba, containing stacked, doubled, or doublewidth diacritics, displaying as expected?

Appendix C

East Asian Language
Search & Display Requirements for Library Databases
Interim Report

Methodology:

This summary is based on an online survey that was made available for two weeks between April 5 and April 20, 2010. It consisted of 16 questions (with 37 data fields) asking respondents about their preferences for search and display of East Asian languages in library databases. The survey was publicized via email to the targeted groups: mainly East Asia librarians and library staff, area studies faculty, students, researchers, and others from East Asia-related programs as well as more general library staff who need to use Chinese, Japanese, and Korean (CJK) or other non-Latin scripts languages in their work.

User Profile:

A total of 366 surveys were submitted. Sixty-one percent of respondents are librarians or library staff, while 35% are library end-users: 16% are faculty, 11% are graduate students, and small numbers of undergraduate students (5%) , and researchers (3%) are represented.

Among the librarians and library staff, many perform cross-functional duties, resulting in about twice as many job categories reported than the total number of respondents (428 responses from 215 respondents). While 68% situated themselves within cataloging, over 30% reported working in collection development or reference. About 25% reported working in acqusitions, 20% in management, 11% in systems support, and 4% in inter-library loan.

Regarding types of libraries represented, most of the library-based respondents reported working for academic or research libraries (84%) while about 10% said they work in public libraries.

Close to the half of respondents reported English as their first or primary language (44%). Twenty-four percent answered that their first or primary language was Chinese. Twenty percent reported Japanese as their primary language, and 5% responded with Korean. Yet, if we include other languages for reading knowledge or proficiency, Chinese language users are 46% of 366 respondents, Japanese language users are 59% and Korean language users are 15%, respectively. There are more faculty and graduate students among Japanese language users (23% and 16% respectively), than Chinese (17% and 8%) or Korean (11% and 15%).

In terms of geography, the majority of respondents were in North America (82%), mostly the United States (78%). Ten percent are in Asia, 6% in Europe, and 2% in other regions. Within the United States, Connecticut is the most highly represented state, but still comprised only 22% of the total. Next was California (11%), followed by New York (9%). A total of 18 countries and 34 states are represented in the survey results.

Specifically deserving of mention is the unexpectedly high rate of comments provided by responding faculty (51% of 59), researchers (46% of 13) and graduate students (44% of 41), which might suggest a strong interest in better integration of original scripts in discovery applications among this population sample.

Survey outcomes:

Four conclusions can be inferred from the survey results.

1. Users prefer having both Romanized and original script content in bibliographic descriptions.

2. Users prefer to include both Romanized and original script content in display. The strongest preferences are for the inclusion of both forms in search result list and individual record views. The language/script for screen interface and facets come secondary.

3. The solution to the issue of character and script variations (i.e., simplified and traditional Chinese, old and new Japanese kanji, and Korean hangul and hanja) is also a priority.

4. The current alternative for relevancy ranking is not preferable to most respondents, but results were inconclusive about what would be better.

Original Scripts vs Romanization:

- More than 80% of respondents disagreed (including 40% who strongly disagreed) that it was acceptable to display original script exclusively.

- Graduate students and faculty tend to disagree more strongly (92% and 85% respectively) than library staff (76%).

- Regarding the language and script of search queries, the results are a bit more complicated. More than half agree in some way (25% 'strongly agree' and 33% just 'agree') with the preference for using original scripts over Romanization, while over a third disagree (12% 'strongly disagree' and 12% just 'disagree').

- Yet, more than 80% of graduate students agree (31% 'strongly agree' and 53% just 'agree') as well as over 60% of faculty (26% 'strongly agree' and 38% just 'agree'). Compared to these groups, library staff were more evenly divided on the issue (52% agreeing in some way vs. 40% disagreeing).

- The majority of comments support both original scripts and Romanization for both searching and display. Some of major reasons to support original scripts are: romanization system varies, and sometime difficult predict the romanization variations, difficult or unpredictable character readings especially proper names, etc.

- Some of major reasons to support Romanization include: Romanization could be used for bibliographic ciation; many citations related to CJK topics still use Romanization and it's hard to find them with only original script searches; it's hard for non-native CJK speakers who still need to work with CJK language materials; and it's useful for character/script variations (i.e. traditional "毛澤東" vs simplified "毛泽东" Chinese terms could search by one Romanized term "Mao Zedong"), etc.

Display Preference for Romanized and Original Scripts:

- Similar preferences were expressed for both "search result list" and "record view". Over 80% of respondents prefer both scripts shown in parallel or side-by-side (81% and 84% respectively), and express the importance for users to be able to modify display according to their preferences (88% and 87% respectively).

- Display preferences for "screen interface" and "facets" were similar. While about 60% prefer both scripts (62% and 56% respectively), about a quarter of respondents prefer Romanization/English only. Compared to the above two display preferences for "search result list" and "record view", these two peripheral displays rated as less important (with 3 being the highest score, "record view" rated 1.81; "search result list", 1.72; "screen interface", 1.51 and "facets", 1.51).

- Among 40 comments, which pretty much support the above statistical analysis, one of the interesting ideas by a few respondents is that the either information diplayed by "tool tip", not parallel, side-by-side, or toggled; i.e. if you have original scripts displayed, you could see the information in Romanization on mouseover.

Character & Script Variations:

These questions related to character and script variations addressed the problems presented by simplified and traditional Chinese, old and new Japanese kanji, and Korean hangul and hanja. Results were analyzed by the language proficiency of the users (for example, responses from the Chinese language users were analyzed for the question about simplified vs. traditional Chinese).

- All three language groups expressed a similar preference for the collocation of character and script variants. The majority (more than 90%) prefer retrieving results in both 'variant' scripts (when there are two) regardless of script used in the initial query. While about 60% prefer giving higher relevance to an exact script match, about 30% were fine without this special treatment. Less than 10% support the idea that only an exact script match should be returned.
- How important did the respondents think this variation issue was to be set by their own preference? This result came out slightly different by each language. More Korean

language users tend to think hangul and hanja's variation treatment should be set by users' own preference (1.67 points, out of 3 as the highest), while Japanese language users less so for Japanese old vs. new kanji variations (1.43). Chinese language users' preference on simplified vs. traditional variations were in the middle (1.57).

- One of the most interesting comments is " … Allowing many language choices for screen interfaces might be convenient, but they are counterproductive for encouraging better communication between researchers and librarians in the long run. I don't think it's important to allow users to choose their language preferences if both Romanized and original-language terms are included in the search result list and record view."

Sort Order:

- Only 18% think "relevancy ranking alone is fine" and a total of 70% prefer "relevancy" along with an alternative alphabet-type arrangement. More in depth research is required to identify preferred alternatives.

- "Unicode code point," the most common sorting alternative to relevancy ranking currently in use in library systems, is considered adequate by only 3% of respondents. The traditional dictionary order for Chinese characters, "by radical/stroke order", is also not considered desirable, with only 4% approving.

- Among the alphabetical sort orders, arranging each language's alphabetical equivalent order was more popular (43%) than by Latin-script alphabetical order (28%).

- Japanese language users reported having less of a preference for "each language's alphabetical equivalent order" (40%), and then 34% preferring "Latin-alphabetical order by Romanization". By comparison, 55% of the Korean language respondents prefer "each language's alphabetical equivalent" and 21% preferred "Latin-alphabetical order by Romanization."

- The respondents' comments vary from those of some who don't understand the terms used or the question iteself, or the ramifications of it, to those with a variety of opinions supporting some of the alternatives presented in the survey. This result needs further analysis by individual comments, or through an interactive focus group. In addition, several respondents commented that they would like to sort by publication dates.

Preference of Headings' Scripts:

- 63% of the respondents reported an experience using original script headings.

- Graduate students have used them the most (78%) while 28% of the faculty have never used them.

- While 42% use either kind of heading, about the same number of respondents have their own preference (44%), which were equally divided between original script headings (22%) and English/Romanization headings (22%).

- More than half of faculty and graduate students use "either" and about over 20% prefer "original script links" while library staff were divided almost equally among the three options, namely: "either", "English/Romanization links" and "original script links".

- More library staff prefers 'English/Romanization links" than the other two groups.

Priority of Other Functions & Additional Information:

- The top three priorities reported by survey respondents according to our 3-point ranking system were: "subject links in English" (2.27), "tables of contents" (2.14) and "cite records with CJK" (1.93).

- Other popular functions and additional information include: presence of "subject links or keywords in the language of the item" (1.88), and the ability to "email records in CJK" (1.80).

Additional Findings and Recommendations:

- This survey resulted in a rich data sample that, with more in-depth analysis, may generate valuable information about preference variations across user categories and languages, especially given the high volume of free-text comments, as more than a third of the respondents (37.4% of 366) added at least one comment for the questions about searching and display.

- This survey focused on the CJK community. Investigations among communities of other language users especially Arabic/Persian and Hebrew/Yiddish are desirable. More in-depth surveys and focus group interviews with different user groups and different language groups should be also considered.

- More than 60% of respondents use original script headings, which is not really authority controlled yet and does not function the same as English/Romanized headings. We recommend more research on how original script heading functionality can be improved.

- New types of language functions and additional information, such as "spell-checking or suggesting alternative words", "tagging in CJK characters", "links to book reviews (both English and the language of the item)" and "auto-complete for CJK", do not seem to be highly desirable for this audience at this time. We need to investigate the relative importance of these features further for JACKPHY users.

Additional data analysis and result charts, and the respondents' free-text comments, are available at the Yale Library SharePoint site:
https://collaborate.library.yale.edu/yufind/Shared%20Documents/Arcadia/EALSurveyAnlyzCharts.doc
https://collaborate.library.yale.edu/yufind/Shared%20Documents/Arcadia/EALSurveyComments.doc