Integrated Library Systems and Dis-Integrative Pressures

Alan Manifold[#]

Abstract: Automated library systems began small and have consistently, if not steadily, traced an arc from less to more integration. Economies of scale, a desire not to duplicate data unnecessarily, and the advantages of smooth and complete integration have been unanswerable arguments as applications have grown from single-purpose utilities to fully integrated systems, providing a wide array of core services to libraries. But throughout this history, there have also always been arguments in favour of less integration and more specialisation of function. Monolithic systems cannot provide the level of quality in many features that a small, focused implementation can in one, nor does a unified system offer as much opportunity for customisation and flexibility as one assembled from "best of class" applications. The tension between these two alternative views has been and will continue to be an important factor in the development of library systems. As the mission of libraries expands and brings with it a proliferation of automated systems, the need to integrate them will almost certainly prove stronger than the desire for superior functionality. But system developers must take into account the requirement for flexibility or they will alienate some of their most gifted potential users.

Keyword: Intergraded Library System, Library Automation, Library System

EVOLUTION OF INTEGRATED LIBRARY SYSTEMS

The Integrated Library System did not spring fully grown and armed out of some computer programmer's mind, as Athena sprang from Zeus's forehead. The ILS evolved over time from much more humble beginnings. Also, it did not evolve in a single form, but evolved separately in the offerings of a variety of vendors and institutions.

The first library systems, in line with the available hardware and the software capabilities of the time, were very simple, single-function applications designed to eliminate tedious manual work involved with particular batch processes. In the 1960s and early 1970s, there were a number of forays into "offline" computerisation of such operations as "…serial

processing and lists, circulation, acquisitions, and book fund accounting (and occasionally cataloging), depending on pressures being felt in the institution, the interests of library management, the interests of funding sources, available expertise, and the computer facilities or services available" (Montague, 1993, p. 78). Borgman (1997) reports, "Librarians and programmer-analysts applied standard systems analysis techniques to identify individual operations that could be mechanized, to determine relationships between tasks, and to isolate the points at which human judgment was required" (p. 219). These functions usually ran in the background, once per day or less, and were for staff, rather than public functions.

Over time, some libraries developed more than one of these computerised applications. Each one was designed separately, often by different people, and frequently running on different hardware. Few libraries had their own computers in the early days, so they borrowed facilities from parent organisations. Thus, they were completely dependent on what computer resources were available to them, which could vary from year to year (Bregzis, Gottlieb, & Moore, 2002, p. 51).

Many of the assumptions we commonly make about data these days were not so obvious in the early days, such as the desirability of entering and storing data only once, rather than separately in each disparate system (Lopata, 1995, p. 2). But librarians were smart enough to see fairly quickly the advantages of sharing data for multiple purposes. Some libraries that had the necessary computer infrastructure, and some visionary entrepreneurs began adding more and more features to existing systems to make them handle more essential processing. As the technology evolved, some of these operations could be done through terminals in real time, rather than only in overnight batches.

The result of this gradual process of tweaking, adding on, improving, and more tweaking was the emergence in about the mid-1970s of the first systems that could begin to be called Integrated Library Systems. An early integrated system might have had only circulation capability and even that was limited. But as these systems were implemented in various libraries, the demands of librarians for more functionality eventually bore fruit. The ILS came to be identified as a system that offered at least cataloguing, circulation, acquisitions,

and an online public-access catalogue (Dempsey, 2005). Other features/modules could also be present, such as serials control, Z39.50 searching, or some kind of booking capability.

Most of these systems started with one or more core modules, such as circulation or acquisitions, then built from there ("Northwestern joins RLG", 1346). As a consequence, one system might have had very well-developed circulation functionality, while another had a strong acquisitions offering. Libraries that had begun their own automation efforts were very reluctant to purchase an integrated system that did not handle the processes they had already automated at least as well as what they already had. But, in the interest of having an integrated system, most of these stand-alone systems were replaced by an integrated offering.

As the integrated systems available came to be identified with vendors, as opposed to the libraries where in many cases the systems were originally developed and implemented, the vendors came under increasing pressure to include full and comparable functionality. Libraries, often with the help of consultants, created Request for Proposal (RFP) documents that dictated the features they required in systems for purchase. The widespread sharing of RFP templates resulted in more and more commonality in the set of supported features. Each system had its strengths and its weaknesses, and each went about its work in a distinctive way, but there was a broad base of overlapping functionality in all systems. As of 2005, Dietz and Grant could confidently say, “It is generally agreed, even among vendors, that ILS products all basically do the same things and do them rather well. These basic functions account for 80 percent of product functionality” (p. 40).

PRESSURE FOR SEPARATION OF COMPONENTS

It was partly the history of evolution from focused, single-purpose system to integrated whole that raised and legitimised the desire for systems to support interoperation of various subsystems. A library felt that it had a perfectly reasonable desire to use, for example, modules from one vendor for acquisitions, from another for circulation, and yet another for cataloguing, and then create their own OPAC. An “open” systems would allow “…libraries to select components based on performance. By assembling a system from a number of modules supplied by different vendors, a library can obtain the best available quality in all

features, and perhaps even produce a system that includes all the desired functions" (Webb, 1987, pp. 257-258). Bregman and Burger (2002) sum up the prevailing argument at the University of Illinois in this way: "If, however, the library could shop for and link the best of all possible systems in each area – the best circulation system, the best acquisitions system, and so on – if one crashed, the others would continue to operate, which would not be true with integrated systems" (p. 77).

Not surprisingly, vendors completely rejected this idea, which was never a real possibility. Systems that are designed to be unified are not likely to be designed to be modular as well. One could not expect, for instance, to take a long-distance runner's torso, graft on a sprinter's legs, a weight-lifter's arms, and an engineer's head to create the smartest, fastest and strongest athlete ever. Even though each person has legs, arms, a torso, and a head, they are not designed to be detached from one another. In the same way, an integrated system's circulation and acquisitions modules will assume that cataloguing records are stored in a particular way, the serials component will expect fund records from the acquisitions function to be in a specific format, and the OPAC will make assumptions about how data from all of the other pieces is organised and stored.

For this type of mix-and-match functionality to work, each system has to share a common set of standards. The standards set down specifically what the expectations are for each of these subsystems within two broad essentials. First, an agreed delineation between the parts is necessary so that one system will be confident that another system includes all of the functions necessary for interoperation. Dietz and Grant (2002) state, "Such cooperation requires standardizations to ensure that products become increasingly plug-compatible in order to provide the level of integration that librarians are accustomed to in their ILS products" (p. 39). Second, a shared protocol for communication between systems is fundamental for the interchange of parameters, requests, and data.

While some standards grow from one particular product's way of doing things, others are created by consultation among a community of stakeholders, and still others are officially sanctioned by bodies established by governments (Sutor, 2006). The history of the library automation industry includes examples of all three models, despite Karen Calhoun's (2006)

accurate assessment of the "difficulty achieving consensus on standardized, simplified, more automated workflows" and an "unwillingness or inability to dispense with highly customized acquisitions and cataloging operations" (p. 13). In regard to the interoperability standards mentioned above, no system was designed along the necessary lines to develop into a de facto standard, there was no "community" of developers willing to work together on developing them, and no government agency stepped in to create and enforce such standards. Therefore, no such standards exist, and there has never really been any prospect that they would.

FURTHER EXTENSION OF THE ILS

As the definition of a core ILS stabilised, vendors sought to differentiate their products from each other. One way they did this was by supporting additional functionality, modules, and links. Two of the first of these were the ability to load journal citation databases locally and search them through the OPAC interface (Lynch, 1992, p. 412; Steffey, 1990) and the ability to search other library catalogues through the Z39.50 protocol ("NOTIS demonstrates", 1992). Other extensions included things like Interlibrary Loan (ILL), an interface with a robotic book retrieval product, media and room booking, federated searching capability, OpenURL resolver, selfcheck machine interface, and electronic resource management. Some of these added features became relatively commonplace, while others were limited to a small set of products.

ILL, booking, and electronic resource management capabilities all appeared both as features bundled with integrated systems and as stand-alone products. Typically, the advantage of having the capability provided by the vendor was the tighter and more feature-rich interface between the two. With full knowledge of the underlying structure of the ILS, any add-on module could facilitate very efficient workflows requiring interchange of data between the two systems. On the other hand, a stand-alone product, while hampered in its ability to interact with the ILS, could generally offer more customised and robust functionality than that provided by an all-purpose product (Jackson, 2000). Neither approach is ever likely to eliminate the other completely.

Of the basic modules of the ILS, the patron user interface or OPAC proved one of the most problematic. The explosive growth of the world-wide web, which happened after most of the widespread integrated systems were created, ushered in a new era in human-computer interfaces in general. Calhoun (2006) said this was a time when it was "…more important than ever to position the research library catalog successfully within a rapidly evolving information universe for scholarly research, teaching, and learning, and to adapt to sea changes in information seeking behavior" (p. 9). Vendors scrambled to produce a web version of their OPACs in order to meet the requirements of libraries with increasing reliance on the world-wide web. In most cases, the web version of the catalogue was only slightly more customisable than the Windows- or Mac-based OPAC had been. Libraries wanted more, but the structure of the software did not lend itself well to a very high level of customisation.

In fact, any type of OPAC was a bit of an afterthought. As described above, most of the integrated library systems evolved from one or more back-office support systems – circulation, acquisitions, serials control, etc. (Bowman, 2007, p. 318). Once the leap had been made from separate modules to a model where the data for all functions was stored in the same database, the opportunity arose to repurpose this data for display to library patrons. The provision of a public-oriented interface had great appeal to libraries, which saw it as a way to deliver the content of their card catalogues without limitation of location. But neither the structure nor the content of the data in use by the rest of the system was ideal for this new purpose.

DISCOVERY SYSTEMS AND THEIR FOLLOW-ON EFFECTS

The advantages of an integrated system had become so thoroughly accepted that few people were able to conceive of a solution to the problems associated with an integrated OPAC. Working with Endeca Technologies, the North Carolina State University (NCSU) Libraries pioneered a new concept for a patron-oriented interface to library data: the discovery platform (Breeding, 2007, p. 38). For this model, essential data is exported from the ILS database, ingested into the user interface's own database, and kept updated through an on-going export/ingest process. In that environment, the data can be augmented and enhanced. Also, data from multiple sources can be brought together,

allowing for a single interface for materials, both those that can and those that can't be easily represented in the ILS (Antelman, Lynema, & Pace, 2006, p. 131). Because it has its own database, focused entirely on the needs of the library patron, the discovery system can easily support the addition of tags, comments, reviews, ratings, and other user contributions, as well as links to external sources of similar data. Increasingly, these systems are able to provide results tailored specifically for a particular user. Finally, a separate discovery product, like all stand-alone products, has a better chance of providing the kind of customisability and flexibility desired by sophisticated libraries.

The quantum leap represented by the development of separate discovery platforms had considerable flow-on effects in regard to the integrated library system. It opened a door in people's thinking that permitted the exploration of creating stand-alone products which can be easily developed and customised by a single library, and can also interface with the ILS. The idea is to use the ILS for what it does well, which is to keep a large body of disparate data reliably synchronised, but to create programs to perform specialised tasks that fall outside the capabilities of the ILS. In other words, next generation systems, "…do not, in most cases, replace the library management back-end system, but instead graft new front-end, patron-facing components to the in-place ILS" (Cervone, 2007, p. 6). The same approach can be used to embed single-purpose widgets into a website to bring standard ILS functionality directly to users, in a well-defined and more easily controlled manner.

Growing demand for such capabilities has been answered by the development by vendors of Application Programming Interfaces (APIs) into the ILS. Most current APIs come in the form of a set of web services, which provide machine-to-machine interoperability using standard web protocols, "…a technology that's hot in the business world but finding much slower adoption in the library automation world" (Dietz & Grant, 2005, p. 39). One call might allow either querying or updating patron address information. Another might either retrieve a list of books checked out to a patron or allow them to check out another one. Having a full API set available can alleviate part of the desire to have the best of each module. If there is a particular way a library would like to display or enter data, a locally developed program may be able to supply the need. Even a fairly small set of calls can provide different libraries ways to accomplish a vast array of tasks without using the modules supplied with the ILS. The ILS

database is still central to all operations, and library staff can see and act on transactions that took place through the API, but the possibilities for innovative projects and applications is limited only by the imagination and resources available to the library. A standard set of API definitions supported by all systems would make this type of access even more cost-effective. Dempsey notes, "Various services will need to talk to the integrated library system. Think for example of wanting to check availability or place a hold or search the system from within another interface.... An agreed set of simple interfaces would be good." (Dempsey, 2006)

GROWING MISSION OF LIBRARIES

Does the successful decoupling of the patron-oriented discovery function and the ability to circumvent the ILS back-office modules signal the end of the ILS? That is one possible outcome of this trend, but it doesn't seem likely. A larger trend seems to be more relevant to the outcome: the expansion in the scope of libraries' missions. The ILS has traditionally focused on data at the level of a bibliographic entity, such as a book or a journal. But today's libraries handle a great many other things as well. More and more, libraries can have large collections of digital objects, which might be digital versions of photographs, maps, plans, etc. They also have to track a variety of electronic journal articles, often licensed through a subscription process. As eBooks proliferate, many libraries find that contracting with an eBook lending service like Overdrive is the most effective way to offer access to their patrons ("Ohio consortium", 2004, p. 28).

Probably biggest among the new information resources are the so-called "mega-aggregate indexes", such as Ex Libris's Primo Central (Rogers, 2010, p. 20). These services provide a single, massive index to hundreds of millions of electronic texts, including journal articles, both scholarly and less so, newspaper articles, and eBooks. Because all of the metadata from many sources is consolidated into a single index, performing a search on these products is incredibly fast, with nearly instantaneous retrieval of the associated records. From there, the full text (where available) is only a click away. While this sort of capability has been present for years through federated searching products, mega-aggregate indexes speed up the process so much that it makes it palatable for people who do not have the patience to wait for the completion of a federated search.

These various extensions to the basic service of libraries all serve to lessen the overall importance of the data in the ILS and thus the system itself: "…the library is acquiring additional systems: the ILS is managing a progressively smaller part of the overall library operation" (Dempsey, 2006). Where it used to seem quite reasonable to think of linking the best acquisitions module to the best circulation module (Pace, 2004, p. 36), that concept now seems a bit passé. Instead, a set of applications, including the ILS, will need to communicate with one another: "…it is now likely that there will be a series of (interoperable) applications covering OPAC's, ereserve, portals and document delivery services which may, or may not, be part of an integrated systems platform" (McLean and Lynch , 2003, p. 12). When libraries are faced with the prospect of integrating their ILS with a discovery platform, an eBook lending system, a mega-aggregate index, and a digital repository, the prospect of breaking the ILS into pieces that need to be integrated has no traction whatsoever. On the other hand, modules that have not been as tightly or as universally consolidated into the ILS may have less likelihood of being consolidated in the future. For example, ILL, link resolvers, and booking systems have a reasonable chance of remaining as stand-alone offerings.

Of course, all of this may change as the industry moves into the latest phase of its evolution. Systems such as Alma from Ex Libris and WorldShare Management Services from OCLC will be replacements for traditional ILS products, but surpass the old definition significantly. Of very recent vintage (some not yet operational), these products, increasingly labelled as "library services platforms", "fully embrace current-day architectures and technologies and are designed from their inception for delivery through multitenant software as a service" (Breeding, 2011, p. 33). On top of basic ILS functionality, they handle a variety of metadata formats, manage electronic resources, and may have built-in link resolvers and/or digital asset management capabilities. In short, they carry the integrative principle embodied in the ILS to a new level. In addition, with all the data held together by the vendor in the cloud, there is the possibility of extensive sharing of metadata, expertise, and work across the customer base. The Alma "Community Zone" is one manifestation of this concept. As platforms, these systems encourage sites to develop new ways to use their data. "Instead of a single interface, there are innumerable ways of interacting with the data. Instead of a single website, the data is free to be displayed anywhere on the web" (Sherratt, 2013, p. 2).

CONCLUSIONS

The integrated library system has always provided a significant set of advantages. The extensive and total integration of data and functioning that is possible in an integrated system cannot be matched by even the best integration of disparate systems. Beyond that, an integrated system provides for one-stop shopping, where the entire range of required features can be purchased in a single transaction. Then after implementation (by a single team), there is only one Support organisation to work with to keep the system operating properly. (Webb, 1987, p. 260)

If the lessons of history are applicable, we are likely to see two things continue as the next generation of systems grows and blossoms: an ever-growing scope of integration on the one hand, and a desire for superior module functionality on the other. Libraries have different needs, different values, and different stakeholders; their system selections reflect those differences. When it comes time to select a new system, a library might well be attracted to the sophisticated and elegant fund accounting of one system and the robust and intuitive course reserves capability of another; they will want both. At the same time, vendors will be folding more and more functionality into their integrated systems, so that operations that used to be handled primarily in stand-alone systems will be included as part of the base package of a new system. If a library particularly likes their old product, there might be a way to use it with the new library services platform, but there might not. Libraries are rarely in a position to be able to afford redundant systems.

Over time, the logic of incorporating what are now separate functions will become increasingly apparent. The ILS brought together functions that were previously separate, but it has now come to be regarded as an inseparable entity, although some aspects that were subsumed into the ILS were not as tightly integrated, and thus never came to be regarded as part of the base package. The next generation is likely to see the same pattern. Some things that are part of the scope of one or more of the library services platforms will eventually come to be regarded as core functionality, while others may not make the cut. Overall, expansion of the definition of an integrated library system (or library services platform) is

inevitable, but system developers will need to provide pathways, probably through APIs, into their systems to accommodate the desire for local enhancements and flexibility.

#About the Author

Alan Manifold

Manager, Digital and Library Applications

State Library of Victoria, Melbourne, Australia

Email: amanifold@slv.vic.gov.au

References

Antelman, K., Lynema, E., & Pace, A. K. (2006). Toward a twenty-first century library catalog. *Information Technology and Libraries 25*(3), 128-139. Retrieved from http://ejournals.bc.edu/ojs/index.php/ital/index

Borgman, C. L. (1997). From acting locally to thinking globally: A Brief History of Library Automation. *The Library Quarterly 67*(3), 215-249. Retrieved from http://www.press.uchicago.edu/ucp/journals/journal/lq.html

Bowman, J. H. (2007). OPACS: the early years, and user reactions. *Library History 23*(4), 317-329. doi:10.1179/174581607x254802

Breeding, M. (2011). A Cloudy Forecast for Libraries. *Computers in Libraries 31*(7), 32-34. Retrieved from www.infotoday.com/cilmag

Breeding, M. (2007). An industry redefined. *Library Journal 132*(6), 36-48. Retrieved from lj.libraryjournal.com

Bregman, A., & Burger, R. H. (2002). Library automation at the University of Illinois at Urbana-Champaign, 1965-2000. *IEEE Annals of the History of Computing 24*(2), 71-85.

Bregzis, R., Gottlieb, C., & Moore, C. (2002). The beginning of automation in the University of Toronto Library, 1963-1972. *IEEE Annals of the History of Computing 24*(2), 50-70.

Calhoun, K. (2006). The changing nature of the catalog and its integration with other discovery tools: prepared for the Library of Congress. Retrieved June 25, 2014, from http://www.loc.gov/catdir/calhoun-report-final.pdf

Cervone, F. (2007). ILS migration in the 21st century: things to think about this time around. *Computers in Libraries 27*(7), 6-8, 60-62. Retrieved from www.infotoday.com/cilmag

Dempsey, L. (2005). The integrated library system that isn't. Retrieved June 25, 2014, from http://orweblog.oclc.org/archives/000585.html

Dempsey, L. (2006). The (digital) library environment: ten years after. Retrieved June 25, 2014, from http://www.ariadne.ac.uk/print/issue46/dempsey

Dietz, R. & Grant, C. (2005). The dis-integrating world of library automation. *Library Journal 130*(11), 38-40. Retrieved from lj.libraryjournal.com

Jackson, M. E. (2000). Interlibrary loan and resource sharing products: an overview of current features and functionality. *Library Technology Reports 36*(6). Retrieved from www.alatechsource.org/ltr

Lopata, C. L. (1995). Integrated library systems - ERIC Digest. Retrieved from http://eric.ed.gov/

Lynch, C. A. (1992). The next generation of public access information retrieval systems for research libraries: lessons from ten years of the MELVYL system. *Information Technology and Libraries 11*(4), 405-419. Retrieved from http://ejournals.bc.edu/ojs/index.php/ital/index

McLean, N. & Lynch, C. (2003). Interoperability between information and learning environments – bridging the gaps: a joint white paper on behalf of the IMS Global Learning Consortium and the Coalition for Networked Information, version of June 28, 2003. Retrieved June 25, 2014, from http://www.imsglobal.org/DLims_white_paper_publicdraft_1.pdf

Montague, E. (1993). Automation and the library administrator. *Information Technology and Libraries 12*(1), 77-85. Retrieved from http://ejournals.bc.edu/ojs/index.php/ital/index

Northwestern joins RLG; brings NOTIS database. (1980). *Library Journal 105*(12), 1346. Retrieved from lj.libraryjournal.com

NOTIS demonstrates full Z39.50 interoperability and introduces two Z39.50-based products. (1992). *Information Standards Quarterly 4*(4), 1-3. Retrieved from www.niso.org/publications/isq

Ohio consortium adds ebooks. (2004). *Library Journal 129*(12), 28. Retrieved from lj.libraryjournal.com

Pace, A. K. (2004). Dismantling integrated library systems. *Library Journal 129*(2), 34-36. Retrieved from lj.libraryjournal.com

Rogers, M. (2010) Ex Libris beta testing Primo Central. *Library Journal 135*(3),20. Retrieved from lj.libraryjournal.com

Sherratt, T. (2013). From portal to platform – building new frameworks for user engagement: presentation to LIANZA Conference 2013, 20-23 October, Hamilton, New Zealand: Wai-Ora, Wai-Māori, Waikato. Retrieved June 25, 2014, from www.lianza.org.nz/sites/lianza.org.nz/files/p5_tim_sherratt_0.pdf

Steffey, R. J. (1990). The NOTIS multiple database access system: a look behind the scenes. *Online 14*(5), 46-49. Retrieved from http://www.infotoday.com/online

Sutor, B. (2006). Open standards vs. open source, Part 1: standards. Retrieved June 25, 2014, from http://www.sutor.com/c/essays/osvsos

Webb, T. D. (1987). Open systems sing a siren song. *American Libraries 18*(4), 257-260. Retrieved from www.americanlibrariesmagazine.org