International Journal of Research in Social Sciences
Vol. 13 Issue 03, March 2023,
ISSN: 2249-2496 Impact Factor: 7.081 UGC Approved
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Integrated Library Management Systems (ILMS): Merits and Demerits of Open Source and Commercial Software

Jawed Akhtar
Senior Library and Information Assistant (Cataloguer)
Dr. R. M. L. National Law University, Lucknow
E-mail: *jawedrmlnlu@gmail.com*

Abstract:

A computer-based system for managing internal and external resources, including tangible assets, financial resources, documents, and people, is known as an Integrated Library Management System (ILMS). It performs library automation and collection development tasks that are separated into numerous modules to simplify common library activities such as acquisition, cataloging, and circulation. It runs on a shared computer platform and is based on a central database, combining all library operations into a single, enterprise-wide system. The purpose of this study is to examine the advantages and disadvantages of popular open-source and commercial library management systems. Some of the LMS systems that were compared were Koha, Evergreen, NewGenLib, Libsys, Voyager, and SOUL. The paper recommends the best-suited solution for use in a university setting based on the benefits and drawbacks.

Keywords:
Integrated Library Management System (ILMS), Open Source Software, Commercial Software, Library Automation, Library Management

1. Introduction:

In human undertakings, information has always been a critical component. Mankind has always attempted to create, collect, and disseminate information. As a result of the information needs, libraries or documentation and information units have been established to execute the responsibilities of identifying, locating, evaluating, and mediating information. LMSs (Library Management Systems) are computer-based systems that automate one or more of a library's functions. To highlight the fact that all functions are controlled via a single database with procedures that openly communicate data between functional components such as catalogue entries and circulation transactions, LMS has also been referred to as Integrated Library Management Systems (ILMS) (Ali Adikata, 2006). The word 'integrated' refers to a system in which all library functional modules are processed against a single master bibliographic database, such as acquisitions, circulation, cataloging, serials control, budgeting, and OPAC (Online Public Access Catalog). Libraries are currently confronted with problems brought by an increasingly diversified and fast-expanding information environment. Institutional aspirations for enhanced operational efficiency go hand in hand with rising user expectations for faster and simpler access to important information (Joy, 2014).

Academic, research and national libraries can utilize the integrated library system to get the efficient, user-friendly tools and workflow support they need to satisfy the growing demands of library users. As a result of technological advancements and organizational growth, librarians' roles have evolved to include more than just preservation and circulation. They now also provide Information Services. The automation of library routines allows for the delivery of superior services and a better user experience.

2. Evolution of ILMS:

A snapshot of the evolution of LMS is discussed in five stages below, from the introduction of mainframe computers in the mid-1950s to the present-day concept of Web 2.0. The five stages are as (Reddy, 2013):-

2.1. First Generation Systems (the 1950s – 1960s):

- Starting with circulation, stand-alone un-integrated programmes were used;
- No standard metadata was used;
- The focus was on library housekeeping efficiency rather than user access;
- Vendor interest in LMS is non-existent; and
- The majority of the systems are mainframe computer-based and batch-processed.

2.2. Middle Generation Systems (1960s – 1970s):

- The Metadata Standard for Bibliographic Records (MARC) was released;
- The focus was on transferring bibliographic decentralized cataloging, and the distribution of catalogue cards;
- Manufacturers created systems that used catalogue data in other modules including circulation and acquisitions.
- The first-generation integrated LMS was born; and
- Character-based interfaces; are mostly minicomputer-based.

2.3. Pre-Internet Generation Systems (1970s – 1990s):

- Microcomputer-based systems with richer interfaces;
- Client-server LAN systems became the standard;
- Networking via LANs and WANs became available, and libraries began networking with closely connected libraries;
- GUIs allowed interactive programs to be created;
- Vendor-supplied systems with networking capabilities are now available;
- Home-grown systems are both unneeded and ineffective;
- The first-generation OPACs had a good time. The OPACs were designed with librarians in mind;
- The z39.50 Information Retrieval protocol made federated searching feasible; and
- Moving away from proprietary backend systems and toward RDBMS-based search solutions.

2.4.Internet Generation - Web 1.0 (1900s – 2000):

- With the invention of cheap INTERNET connectivity, New client-server systems that used the internet to store data and execute transactions became available;
- Rich GUI front ends using tools like Visual Basic and Visual C++;
- Platforms such as JAVA and .NET became popular for online application development.
- Open-source operating systems such as Linux have made an appearance;
- The backend was still mostly RDBMS-based with SQL-based search algorithms.

2.5. The Web 2.0 Era (2000 onwards):

- The web has evolved from an information delivery-only platform to a participatory platform, with ordinary people contributing via blogs, wikis, podcasts, and social networks;
- This has influenced library users' expectations of libraries and LMSs;
- Web services, improved interoperability, RSS/Atom feeds, and improved user experience in discovery apps, such as Amazon and eBay;
- Dissatisfaction with the LMS's and OPACs' homogenous character;
- Consolidations and mergers in the commercial market space are signs of industry upheavals.

Institutions increasingly have access to various types of business applications, and enhanced LMS integration with these systems is desired. It is clear that advances in hardware and software technology, as well as the adoption of new paradigms such as the relational model, object-oriented analysis and design, client-server topologies, and Web-specific languages, have influenced the evolution. The rise of the web and its dispersed environment under many platforms, formats, languages, and data models have had a significant technical impact, necessitating that the LMS enables interoperability (Ansari, 2017).

3. Perception of Open Source and Commercial Software:

The characteristics of open-source and commercial software development processes, as well as associated licensing approaches and intellectual property foundations, are examined further:

"Open Source" refers to a software licensing style in which the software's source code is commonly made available royalty-free to the software's users under circumstances that allow redistribution, modification, and addition, usually with some restrictions. An open system is the polar opposite of proprietary solutions in terms of design. The idea is that organizations like libraries can put together a collection of components and provide services that mix products from a variety of vendors. To better satisfy its internal or customers' needs, a library could, for example, utilize an integrated library system from one of the main vendors in conjunction with an open-source product built by another library or by itself. Software businesses are also donating paid programmer time and in-house projects (Ghosh & Panda, 2011).

"OSS is both a philosophy and a process. As a philosophy, it describes the intended use of software and methods for its distribution. Depending on your perspective, the concept of OSS is a relatively new idea being only four or five years old. On the other hand, the GNU Software Project – a project advocating the distribution of "free" software -- has been operational since the mid-'80s. Consequently, the ideas behind OSS have been around longer than you may think. It begins when a man named Richard Stallman worked for MIT in an environment where software was shared." (Lingam & Durake, 2019). Lingam & Duarke also explained that "Open source improves software stability and quality by allowing for independent peer review and rapid growth of source code. A program's license must ensure the ability to read, share, change, and use it freely to be certified as open-source".

Some of the popular open-source ILMS available are:

- Emilda
- Evergreen
- FireFly
- Koha
- PhPMylibrary
- OpenBiblio

"Commercial Software" refers to a business model in which a company's software is licensed to a client for a charge, either directly or through channels in the object, binary, or executable code. Customers frequently require support, training, upgrades, and other such services from the commercial company to effectively utilize the programme. Software source code is sometimes made available to certain users through special licensing or other arrangements, but it is rarely made available to the general public and may only be copied or modified by the terms of such agreements (Rao, 1999).

Some of the commercial ILMS software available are:

- ALICE
- LibSys
- LiBSUITE
- SOUL
- Virtua

Open-source and commercial software methods each have their strengths and weaknesses, and depending on the circumstances in which they are used, they can provide users with several benefits as well as trade-offs. The two methods aren't mutually incompatible, and businesses are increasingly finding ways to embrace both and coexist. This strategy allows for a stronger focus on the creation of higher-level components, where innovation may provide more value to clients. Software source code is occasionally made available to certain users via special licensing or other arrangements, but it is rarely made available to the general public and may only be copied or modified according to the terms of such agreements. Even in sectors where there have traditionally been few competitive goods, consumers today have a large selection of software options and suppliers to choose from. Some people favour open-source software because the source code can be freely copied, modified, and redistributed. Such features appeal to users who want to change the source code of the software, such as in academic contexts where experimentation is the primary goal or in situations where a high level of customization is required (Madhusudhan & Singh, 2016).

4. Analysis of Open Source and Commercial Software:

For a better understanding, open-source and commercial software are examined from the business, development, licensing, and technical viewpoints.

4.1. Business:

Although there are some fundamental differences between open source and commercial software suppliers' business strategies, both must discover ways to generate long-term revenue. Because their income model is reliant on the client licensing their programme, commercial software suppliers focus on the functionality, features, and innovativeness of their technology to suit consumer needs. Customers purchase new software versions as they have access to new capabilities, features, and benefits. This incentive encourages companies to invest significantly in research and development, resulting in increased productivity, cheaper business costs, and new learning aids. A system integrator who makes money by customizing solutions for customers using open-source software and charging them for the time and resources required to adapt the software to meet the demands of each unique user (Naik, 2016). Another strategy is to provide a free download of an open-source product and then convert them into paying customers for the full-featured version. It is well acknowledged that there is a set of open-source contributors who are driven by the altruistic belief that all software should be free and that code will be enhanced by volunteers who freely share their work for anyone's use and reference. From the consumer's perspective, the value a customer derives from a commercial software product is frequently linked to the license fee, programme capabilities, and product support. While a customer can hold a commercial software supplier personally responsible for their purchase, most open-source software has no "owner," making assigning blame difficult.

4.2. Development:

Another characteristic that has separated open source and commercial software in the past is the approach to software development. For the most part, commercial software development teams have always worked inside the bounds of a single company or unit. In both commercial and open-source software development methodologies, iteration of design, standards, coding, testing, release, and feedback is a common underlying development process. Skilled programmers can earn recognition in their own right for their contributions to software development by solving unusual and difficult issues, whether they use open-source or commercial software models (Saxena & Srivastava, 1998).

4.3. Licensing:

The major differentiator between open-source and commercial software techniques is software licensing. The conventional software licensing method, in which a client is provided rights to use the product in exchange for a fee, is frequently used by commercial software providers. In most cases, the consumer is only entitled to use, reproduce, or alter the programme in line with the license terms. Open-source software is distributed under a variety of licenses, all of which allow users to alter and redistribute it. As with commercial software, the license agreement is based on the program's copyright. The rights and permissions given are subject to terms and conditions. In general, these limitations set restrictions on how the programme may be altered or disseminated rather than requiring payment of a charge (Singh, K., 2012). The GNU General Public License (GPL) and the Berkeley Software Distribution (BSD) License are the two most used open-source licensing models. All derivative works of the programme, as well as future versions down the chain, are required to be licensed and distributed under the same conditions as the original software under the GPL. GPL-licensed source code stays GPL-licensed indefinitely. Developers that use the BSD License, on the other hand, can combine licensed software with their source code to build new products with few limitations. Except to pay the administrative costs of copying and shipping, the GPL bans charging money for the release of the source code. While the GPL allows for the commercial sale of open-source software, the license and access to the source code allow customers to freely redistribute or change the code without having to pay the original vendor. Fees for system setup, system management, support, maintenance, and other related services are also permitted under the GPL (Zaveri & Salve, 2018).

4.4. Intellectual Property:

Our intellectual assets - data, information, and knowledge – are one of the most significant assets we can utilize in the knowledge economy. Software owners lack the incentive and legal foundation to commercialize their innovations without Intellectual Property Rights, therefore the software sector cannot be a source of economic growth. Although open-source software is frequently accessible for free distribution, the open-source software paradigm requires intellectual property rights protection. Furthermore, open-source developers' source code is frequently authorized on the condition that proper attribution to the source code author is provided. As a result, regardless of the software models used, a strong intellectual property rights system is required (Chauhan, 2018).

4.5. Technical Reasons:

4.5.1. Cost:

The cost of open-source software vs. commercial software should be assessed against the product's lifecycle costs for a specific client. While some open source advocates argue that open source software is less expensive than commercial software, commercial software with comparable functionality may have a lower total cost of ownership, according to proponents of the commercial model. Open-source software may be less expensive than

commercial software in terms of up-front expenditures. When making purchasing selections, consumers must consider the cost of software over its whole lifecycle rather than just the one-time purchase price. Consumers should not assess the cost of software-only based on the initial purchase price, just as they should not compare the long-term expenses of buying inexpensive phones to more costly phones with low recurrent costs. They must also consider long-term support and maintenance requirements, as well as less visible factors such as product usability and productivity improvements. Purchasers should also factor in the expense of retraining customers who are already proficient in one product to become proficient in another. When one considers the number of time customers spend undergoing retraining as well as the initial loss of productivity as users become acquainted with the alternative product, such re-training costs can add up quickly. When deciding on an LMS, IT decision-makers should consider the complete spectrum of expenses, including lifetime and migration costs (Ray & Ramesh, 2017).

4.5.2. Security:

It has been claimed that open source software with publicly accessible source code is fundamentally more secure than commercial software with non-publicly accessible source code. Others argue that exposing source code makes it easier to uncover and exploit software flaws, while others argue that access to source code has no impact on software security. The viewpoints are varied. The three most important components in software security are the quality of the developers, the methods and tools used by the development team to reduce vulnerabilities, and the strength of the customer-software provider relationship. Regardless of the software development process used to produce the product or the rigor with which the programme was tested, a badly maintained product offers no security (Hanumappa, Dora, & Navik., 2014).

4.5.3. Flexibility:

The idea that open source software is more adaptable for consumers than proprietary software is based on the capacity of the client to examine the source code and make the necessary changes. This also enables technically adept people to uncover any system weaknesses and apply their software upgrades or fixes to resolve the problem. When you can change the source code of an open-source solution, you get what's known as "forking." All future upgrades or enhancements made to one version of the software will not apply to the other version if one developer modifies the programme source code and takes a different path than the original software. As a result, issues of compatibility and continuity will develop, which will need to be addressed. Customers that make their software modifications will discover that maintaining and supporting such changes is more complex, as support personnel must be knowledgeable about earlier adaptations as well as the skills required to make subsequent changes. Commercial software solutions, on the other hand, tend to have a better-defined and managed upgrade and migration route for their products. Customizations made using public application programming interfaces on such platforms usually function with current and future versions of the product with little to no adjustments (Upasani, 2016).

5. Evaluating ILMS:

The work of analyzing integrated library systems is required to select the most appropriate library management system to meet the library's automation needs. The following factors should be considered while deciding between open source and commercial software:

- Cost factors should be taken into account in their entirety. While cost is a significant consideration, it is rarely the primary element in a purchase decision.

- The overall personnel required for any software deployment should not be underestimated. Today's market offers suitable experienced and trustworthy labor for software platform support, which may be kept in-house or procured from an outsourcing provider. When evaluating and selecting a software solution, it is critical to consider the complete spectrum of personnel requirements.
- A shared duty between the client and the software supplier is required to ensure the secure and reliable usage of a product. The software vendor is responsible for developing the programme in line with security best practices extensively stress-testing the LMS and quickly developing updates and fixes when vulnerabilities are discovered. Appropriate and sufficient resources should be given by the client to guarantee proper software installation, deployment, and maintenance.
- If a source code security review is required, the necessary knowledge should be made available to thoroughly examine the source code of the components to be deployed. It should not be expected that just because the source code is published, it has been thoroughly scrutinized.
- Flexibility requirements for updating acquired software should be carefully balanced against whether the skills to use such flexibility are available, as well as whether the flexibility requirement is critical or merely incidental. The long-term support consequences of non-standard software changes should also be considered when making a purchasing choice (Pratheepan, 2015).

6. ILMS for Academic libraries:

The growth of an academic library is inextricably linked to the growth of the institution it serves. The previous notion of library service is restricted to a single library has undergone a massive transformation, expanding well beyond its four walls. Ranganathan's seemingly harmless fourth commandment, 'Save the time of the reader,' has taken on a new meaning, creating the concept of instantaneous library service, which has been put into effect through the use of ICT. The open-source ILMS was chosen by libraries firstly because it is cost-effective, and secondarily because it is useful, flexible, and free of vendor lock-in. Academic libraries have discovered that open-source ILMSs is not only more economical and customizable than proprietary ILMSs but also less easy to install and maintain. On average, the overall cost of an open-source ILMS, including both upfront and ongoing expenditures, was cheaper than that of a proprietary ILMS. The initial labour cost of installing an open-source ILMS, on the other hand, is frequently found to be greater than that of a proprietary ILMS. The term "free" in the context of free software refers to the ability to read the source code rather than having no cost. In reality, library software should assist both library operations and library services, as well as expand their reach regularly. When software is used to limit both, the library's development is hampered (Ukachi, V.N., & Onuoha, 2014).

Libraries have a history of adopting modern technology and keeping up with the times, according to their history. The current developments, on the other hand, are rapid, and libraries are struggling to stay up. To cope with the new scenario, they would require financial and administrative assistance from several authorities. Library workers must be educated in the rapidly developing areas of the profession to turn our conventional academic libraries into 21st-century libraries. Individual universities are stepping up to give comparable chances to library workers through refresher courses, seminars, and workshops, as well as in-house software.

References:

1. Ali Adikata, A. (2006). Students' library use: a study on faculty perception in a Malaysian University. *Library Review, 55*, 106-119.

2. Ansari, M. A. (2017). Library Automation in Indian Central Universities: Issues and Challenges. *Cataloging & Classification Quarterly, 55*(4), 247-265.

3. Chauhan, K. (2018). Evaluation in Use of KOHA Library Management Software in OPJGU, Sonipat. *Library Philosophy & Practice, 6*(2), 45-52.

4. Ghosh, T. K., & Panda, K. C. (2011). Automated Serials Control at the Indian Institutes of Technology: An Overview. *Program, 45*(2), 173-184.

5. Hanumappa, A., Dora, M., & Navik., V. (2014). Open Source Software Solutions in Indian Libraries. *Library Hi Tech, 32*(3), 409-422.

6. Joy, B. (2014). KOHA and LIBSYS: A Comparative Study. *Journal of Advances in Library and Information Science, 3*(4), 350-354.

7. Lingam, A. S., & Durake, D. A. (2019). Survey on Koha usage in India. *Indian Journal of Library Science and Information Technology, 4*(2), 92-94.

8. Madhusudhan, M., & Singh, V. (2016). Integrated library management systems: Comparative analysis of Koha, Libsys, NewGenLib, and Virtua. *The Electronic Library, 34*(2), 223-249.

9. Naik, U. (2016). Library Automation Software: A Comparative Study of Koha, Libsys, Newgenlib and Soul. *International Journal of Library Science and Research, 6*(6), 77-86.

10. Pratheepan, T. (2015). Integrated Library Management Systems (ILMS) - Open Source and Commercial Software: An Assessment of the Merits and Demerits. *Journal of the University Librarians' Association of Sri Lanka, 19*(1), 54-70.

11. Rao, R. (1999). Features of Library Automation Software: A Comparative Study. *SRELS Journal of Information Management, 36*(4), 211-228.

12. Ray, A. K., & Ramesh, D. (2017). Open Source Software (OSS) for Management of Library and Information. *International Journal of Library and Information Studies, 7*(2), 20-31.

13. Reddy, C. S. (2013). Comparative Study of Free/Open Source Integrated Library Management Systems (FOSILMS) With Reference to Koha, Newgenlib, and E-Granthalaya. *E-Library Science Research Journal, 12*(1), 1-10.

14. Saxena, S., & Srivastava, R. K. (1998). Evaluation of Library Software Packages Available in India. *DESIDOC Bulletin of Information Technology, 18*(5), 9-17.

15. Singh, K. (2012). *NewGenLib: Open Source Software's In Indian Libraries, 1*(6), 173-179.

16. Ukachi, N., V.N., N., & Onuoha, U. D. (2014). Library Automation and Use of Open Source Software to Maximize Library Effectiveness. *Information and Knowledge, 3*(4), 74-82.

17. Upasani, O. S. (2016). Advantages and Limitations of Open Source Software for Library Management System Functions: The Experience of Libraries in India. *The Serials Librarian, 71*(2), 121-130.

18. Zaveri, P., & Salve, D. (2018). Status of Library Automation Software used in Mumbai College Libraries. *Knowledge Librarian Special Issue, January*, 381-389.