CHAPTER IV
OPEN SOURCE SOFTWARE MOVEMENT: AN ANALYSIS

4.1. Introduction

The term ‘software’ was coined by John W. Tukey and he used it for the first time in 1958 in his article published in *The American Mathematical Monthly* (1958). However, the idea of software was devised by Alan Turing (1936) during his presentation on Turing Machine. Earlier, terms like ‘computer program’ and ‘code’ were in use. Ceruzzi (2003, p. 80) defines software as “a single entity, separate from the computer’s hardware, that works with the hardware to solve a problem.” The early software, the so-called ‘computer program’ and ‘code’, were installed or programmed in the computer during their configuration and it was a herculean task to change, delete, uninstall, and reinstall software on computer. In fact, initially software were part of computers and were not available separately. Computer manufacturers used to sell computer and software as a single entity. But in 1968, with the intervention of the US government, computer manufacturing companies were forced to sell software as a separate unit (Ceruzzi, 2003, p. 106; Kelty, 2008, p. 119) and thus, emerged the computer software industry. However, the real growth of software industry was observed only after the invention of microcomputers in mid 1970s, which brought the computer on the desk of the people. 1990s observed development of Web and Windows system software that tremendously increased the use of application software with Windows. However, the history of software is very extensive and complex, this chapter would concentrate on the development of open source movement with special reference to the software in use for library and information management.
4.2. Genesis of Open Source Movement

In the beginning, there was no existence of proprietary software or shareware/free software. Software were nothing but codes to perform a particular task on the computer and were available with computers in source code (Daffara & Gonzalez-Barahona, 2004; Chudnov, 2007). These codes were frequently used by educational institutions to study them. Customers were also able to modify them as per their needs as in the case of open source software, hence, there was no difference between the user and the programmer. At the launch of the 701, the first commercial computer by IBM in 1952, there was no difference between hardware and software, all computers had been sold with software of one kind or the other. These software were free software with the code. Richard Stallman (2002) also discussed about a software sharing community which was in existence in 1970s to share and help the users who were facing problems or bugs in using operating system ITS (Incompatible Timesharing System) developed by MIT.

The history of free and open source software is twofold. One is related to the development of UNIX operating system and the other related with Richard Stallman’s GNU project.

4.2.1. Development of UNIX

The idea of UNIX was industrialized in 1965 when Massachusetts Institution of Technology (MIT), General Electric Company (GE), and Bell Telephone Lab (BTL) came up with a project MULTICS (Multiplexed Information and Computing Service) with the objective to develop a multi user operating system (Evers, 2000, p. 11). But it did not produce any positive result and in
1969, BTL declared to withdraw its resources and manpower involved in the project as the project was moving towards failure.

However, two BTL researchers Dennis Ritchie and Ken Thompson did not lose their enthusiasm and developed some program and games which they tried on GE645 and PDP-7 computers by altering the codes. Though such a software did not work but they got huge experience from this. Thompson had already developed a file system for an operating system.

In 1969 Ken Thompson spent few weeks on writing *Kernel* (core of any operating system), *shall* (program to read commands) and an *assembler* (to convert codes into software). To work on these, he used DEC’s PDP-7 computer and the results were great for him. He named it UNICS (Uniplexed Information and Computer Service), later renamed as UNIX. Initially, he kept this program very small due to less capacity of PDP-7 computer. It was a time when PDP-10 was in market and Thompson and Ritchie made several attempts to convince BTL to acquire it but they did not succeed.

In 1970s Ken developed a programming language known as B which was later improved and completed by Dennis Ritchie and named C in 1972. C has proved itself very powerful and in common use, even after four decades of its development, because of its portable programming language and flexibility to move program from one kind of computers to another. In the same year BTL acquired the latest PDP-11 computer seeing the success of Thompson and Ritchie on UNIX. “With several BTL staff members from outside the research group using the typesetting facilities of the PDP-11, the need to document the operating system grew. The result was the first Unix Programmer’s Manual by Thompson and Ritchie, which was dated November 3, 1971” (Salus, 1994). Now BTL was using UNIX for their computers instead of having the operating
system combined with the PDP-11 (Weber, 2004, p. 26). The very next year they rewrote UNIX to add more functionality and released the 2\textsuperscript{nd} version of UNIX. With this release, number of installations in BTL reached to ten (Salus, 1994). Ritchie and Thomson had rewritten the kernel of UNIX in C language instead of an assembly language that made it hardware independent. This third edition or version of UNIX was installed in sixteen sites, in 1973, all within AT&T (Evers, 2000). It was the fourth year of development of UNIX but it was not much known elsewhere except AT&T.

A major landmark came in the development of UNIX when Ritchie and Thompson presented a paper on UNIX at the ACM Symposium on operating systems in October 1973. This symposium proved a big advertisement of UNIX to related audience. After that BTL had a number of requests for copies of UNIX. The publication of their paper on UNIX in conference proceeding in July 1974 caused an explosion in demand of UNIX unanimously.

AT&T was able to sell this product at minimal cost in ‘as it is’ form without any royalties to AT&T and without any support and bug fixes (Weber, 2004, p. 28; Salus, 1994). At this time it was open in source form and its source code was being studied in the universities.

When there is no system to help one out then people with similar problem get assembled to help each other and the same was in case of UNIX. Users of UNIX came together to share thoughts, information, codes, bug fixes etc. The group gained popularity by their name Usenix Association. Users around the world were working on UNIX like any true open source software where programmers of various universities and organizations were developing UNIX and new developments and improvements were merged with the new release by AT&T.
The developers of UNIX were not able to help or support users due to legal binding but the user groups were working constantly to solve the UNIX error faced by the users all around the world.

A big change came into working of UNIX in its 7th version which was released in 1979. The most influential feature of this version was its portability. This version was successfully run over computers manufactured by DEC, IBM and Interdata (Evers, 2000, p. 13). The earlier version of UNIX was able to run on different computers manufactured by DEC only. The continuous development and popularity of UNIX made its manufacturers serious towards UNIX as a commercial product and they constrained the distribution of source code in new licence. The new licence terms made many universities to drop UNIX from their course of study. But, until 1982, AT&T could not commercialize UNIX in its full capacity due to the consent decree which restricted AT&T to enter into software business. A court decree in 1982 removed the restriction form AT&T to come into software industry. After this, the development of UNIX became more aberrant. A number of derivatives were observed in the market which confused the users. The most prominent of those is BSD. Other derivatives were either based on AT&T’s UNIX or BSD. However derivatives from AT&T required licence from AT&T whereas this was not the situation in case of BSD. There were some suits filed on BSD too for breaching the licence restriction of UNIX but could not succeed as the earlier versions of UNIX were released with free source code.

4.2.2. BSD (Berkeley Software Distribution): First Open Source License

UNIX came into the University of California, Berkeley in 1974 after Thompson’s and Ritchie’s presentation on the Operating System. At the same time, UNIX was included into the courses of study. Computer researchers, studying and working in Berkeley, started to work on the
improvements of UNIX by adding new features and kernels to the existing UNIX. With this they had a full derivative of UNIX which they could distribute to the users. In 1977, a software distribution licence was developed at the University of California known as Berkeley Software Distribution (BSD). This software distribution licence contained almost all the features that an open source licence should have. BSD allowed the users to use, edit the source or its binary codes, and redistribution of the software with or without permission. However, there were some restrictions on it. The licence reads that “[all] advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the [organisation]” (University of California, 1999).

The development of UNIX and BSD licence was indicating the future revolution towards the free and open source software although the term Open Source was not in existence at that point of time. But people had begun to support the concept of free software. At the same time, when UNIX and BSD were getting momentum, Richard Stallman had started his crusade for free software by establishment of Free Software Foundation.

4.2.3. GNU and FREE SOFTWARE FOUNDATION

Richard Stallman, who had started working with MIT AI Lab (Massachusetts Institute of Technology Artificial Intelligence Lab) in 1971, was the central character of GNU (GNU Not Unix) and FSF (Free Software Foundation). His duty was to improve and enhance the quality of ITS (the Incompatible Timesharing System), an operating system designed and used by MIT AI Lab for its PDP- 10 computers.

In the initial phase, the software was free and MIT AI Lab used to share it with others on demand. Stallman writes:
"We did not call our software "free software," because that term did not yet exist, but that is what it was. Whenever people from another university or a company wanted to port and use a program, we gladly let them. If you saw someone using an unfamiliar and interesting program, you could always ask to see the source code, so that you could read it, change it, or cannibalise parts of it to make a new program". (Stallman, 1999, p. 31)

However, the situation radically changed in 1980s with unfortunate discontinuation of Digital PDP-10 series computers. As the ITS was created and designed using machine depended assembler language, it resulted in obsolescence of all program related to ITS.

The new modern computers of 1980s had their own proprietary software where the purchaser of the computers had to sign a non-disclosure agreement. When MIT AI lab purchased new modern computers, all its scientists required to sign a non-disclosure agreement even for getting an executable copy of the operating system. But Richard Stallman was of a different philosophy and supporter of free software. In words of Richard Stallman (1999) "[signing this agreement meant that] if you share with your neighbour you are a pirate. If you want any changes, beg us". Stallman was not ready to give up his freedom to help others. At this stage, he was left with three options; first, join the proprietary software world by signing non-disclosure agreement and promising not to help others. In words of Stallman (1999) "I could have made money this way, and perhaps amused myself writing code. But I knew that at the end of my career, I would look back on years of building walls to divide people, and feel I had spent my life making the world a worse place."

The second option he had was to leave the computer software industries and turn his way to some other business where his software developing skills would not be misused. But this could not have been helpful for the others and would have wasted his programming skills. The third option which Stallman opted was to develop software that would be free for users not only to use
but also to modify or redistribute. His choice was to develop an operating system as it is very crucial component of the computer and works like a soul of it. If there is an operating system installed in a computer, one can do many things with it. But if there is no operating system, the computer would be dead.

He chose to develop an operating system that is compatible with UNIX. Hence, he started a project for the same and named it ‘GNU’ which stands for ‘GNU is Not UNIX’. He tagged this future operating system with his pioneering term ‘free software’

He gave up his engagements with MIT in 1984 to focus entirely on GNU. The primary motto behind leaving his job was to make sure that in future MIT may not claim GNU as its own product as Stallman was its employee. In very short time he got a product called GNU Emacs. He distributed the program tape at a cost of $150 which was cost of mailing and of the tape. The software was free. During this process Stallman also felt a need of copyright licence to make sure that all derivatives of the software also remain free and in public domain. Stallman (1999)cites an example of the X Window System which was developed at MIT, and released as free software with a permissive license, and was soon adopted by a number of computer companies. These companies added X to their proprietary UNIX systems, and covered them by the same nondisclosure agreement. To avoid this possible future of software, Stallman decided to write a software distribution licence known as GPL (General Public License). This licence introduced the concept of ‘copyleft’. This license is widely used, even today, by open source software to give freedom to users to use, distribute and modify the licence.

The next step of Richard Stallman’s movement was to bring GNU project under an organisation. In 1985, he established a tax free charity organisation under the name ‘Free Software
Foundation’ (FSF). With this organisation he could handle several activities like accepting
donation, selling copies of free software or providing other services. Stallman has attempted to
formally define the term ‘Free Software’. This term refers to the following freedoms which are
granted to the user of the software (Free Software Foundation, 1985):

- Freedom to run the program in any place, for any purpose and forever.
- Freedom to study how it works and to modify it as and when required. This guarantees
  supply of the Source Code of the software.
- Freedom to redistribute copies of the software as it is or in modified form so we can help
  our friend and neighbour.
- Freedom to improve the program and release the improvements to the public. This also
  requires the source code.

Stallman made use of the GPL software distribution licence for any software released by FSF to
guarantee these freedoms. Through this licence, the author allows the user of the program to
exercise these freedoms. In addition to this, FSF also bestowed a freedom to the author to apply
any restriction s/he wishes, i.e., to credit the original author even in its redistribution in modified
form. However, such restrictions must not offset with the above freedoms.

Richard Stallman’s FSF was earning popularity very quickly. Stallman’s planning was to write a
complete full operating system but even before that the components of the OS was demanded by
the computer users. GNU Emacs was used by a number of computer users with UNIX or other
system software. It was supporting Stallman to improve GUN Emacs, but on the other hand it
was delaying his work of writing the full operating system. FSF was also getting donation in the form of money, labour and computers to support the noble cause of Richard Stallman.

By 1990’s Stallman successfully completed all components of GNU except its Kernel. Without a Kernel, which is heart of any operating system, it was difficult to bring out a complete operating system. Stallman wished to develop its kernel based on Mach. Mach was an operating system microkernel developed at University of Utah. Mach was a commercial Kernel and to make it a component it was required to have Mach as a free Kernel. Using a commercial Kernel for free software may affect its freedom as Mach was bound under copyright licence. Stallman, with help of his technical supporters, could write a Kernel for GNU which was known as GNU HURD. However GNU Kernel was full of errors and was incomplete and unsuitable to be used with GNU. Even today HURD is not completed. But, it was fortunate that there was a man who could do this Job and it was Linus Torvalds.

4.2.4. LINUX

Linus Torvalds, an ambitious but unknown student of Helsinki University, started to build a Kernel in 1991. His aim was to build a free UNIX kernel for Intel based computers. He initiated to build this keeping Andrew Tanenbaum’s Mimix system in mind, which was based on UNIX system and published as a book (Kavanagh, 2004). It was also used as a teaching source in universities. However Linus was unhappy with this Kernel. Linus had created his kernel successfully and as he kept it free, he received a number of feedbacks from other computer experts. In the next year, in 1992, Linux was combined with GNU. It resulted in a complete operating system known as GNU/Linux. Kernel by Linus has made the dream of Richard Stallman true. In his words “It is due to Linux that we can actually run a version of the GNU
system today” (Stallman, 1999). Since then, the combination of GNU and Linux with other free software expanded in popularity and become commonly known by single term ‘Linux’. GNU and Linus Torvalds continuously released improved version of GNU/Linux. In 1994 they released Version 1.0, which could compete in terms of reliability and stability with the commercial versions of UNIX. Using Linux, a number of versions were developed. Some of the very prominent models are Debian GNU/Linux developed in 1993, Red Hat GNU/Linux developed in 1994, and Ubuntu GNU/Linux developed in 2004.

4.2.5. Open Source Initiative

After merging with Linux, number of supporter of GNU increased considerably. After 1997, a new philosophy and some disagreement within FSF towards software were growing in FSF and with the passage of time it came out as a new terminology, i.e., Open Source. Eric S. Raymond, wrote a paper ‘Cathedral and bazaar’ and presented it in GNU/Linux Congress organized by Peter Salus, a member of FSF, under the sponsorship of FSF in 1997. The main concentration of the paper was on the two ways of production of free software. Raymond in this paper supported the open way of producing free software. Cathedral, in his view, was the structure of GNU Emacs that has the source code with it but the access of it was restricted to an exclusive group of software developers while Bazaar model of free software also comprises source codes that are available online to every individual to edit and improve. He credited Linus Torvalds for this model of free software.

Eric S. Raymond was an active member of GNU Emacs in 1980s but he distanced himself from this in 1992 after a dispute with Richard Stallman for making some unauthorised changes in
GNU Emacs. In words of Raymond “It frustrated me so much that I decided I didn’t want to work with [Richard Stallman] anymore” (Williams, 2002). Thus he was connected with the philosophy of free software but distanced from FSF and Richard Stallman, however, he continued to work on free software community that enabled him to get invited in GNU/Linux Congress and present his views.

Although most portion of his paper was similar and in practice as free software, but, Raymond provided a new business way through free software. This paper left a very huge influence that it was going to make its place in history. Netscape Communication Corporation, a commercial software company was in the market with its web browser Netscape Navigator, and was involved in a furious struggle, known as ‘browser war’\(^1\), to make its browser dominant browser on the desktop. With the release of Internet Explorer 4, Microsoft decided to give away Internet Explorer as inbuilt software with Microsoft Windows operating system (Wikipedia, 2011). This step of Microsoft brought a higher impact on market of Netscape, resulting in a rapid loss in the number of users of Netscape Navigator. The market condition made it easy for Netscape Communication to understand that it is time for some hard decision by changing the rules, and a presentation by Eric Raymond gave them a choice. Influenced by Raymond’s presentation, Netscape decided to release the source code of Netscape Navigator to public.

On January 22, 1998, Netscape announced to give the source code of Netscape Navigator free to the public under a project known as Mozzila Project. It boosted the morale of Eric Raymond and some of his colleagues. Netscape Communication, through an email, acknowledged Eric Raymond’s paper for their declaration: “On behalf of everyone at Netscape, I want to thank you

---

\(^1\) Wikipedia provides a comprehensive detail of the browser war between Microsoft and Netscape at their page http://en.wikipedia.org/wiki/Browser_wars.
for helping us get to this point in the first place. Your thinking and writings were fundamental inspirations to our decision” (Raymond, 2001).

Soon after this declaration, a strategy session, on 3rd February 1998 at Palo Alto, California, USA, was held by Eric Raymond with Todd Anderson, Chris Peterson (Foresight Institute), John Hall and Larry Augustin (Linux International), Sam Ockman (of the Silicon Valley Linux User’s Group). “The strategy session grew from a realization that the attention around the Netscape announcement had created an opportunity to educate and advocate for the superiority of an open development process” (Open Source Initiative, 2012). Eric Raymond who was already in conflict with Richard Stallman wanted to pave a new way to offer the world free software without any rivalry with commercial software. Chris Peterson of Foresight Institute coined and suggested a term ‘Open Source’ for it. It is further quoted by Open Source Initiative (2012) on its history page:

The conferees believed the pragmatic, business-case grounds that had motivated Netscape to release their code illustrated a valuable way to engage with potential software users and developers, and convince them to create and improve source code by participating in an engaged community. The conferees also believed that it would be useful to have a single label that identified this approach and distinguished it from the philosophically and politically focused label "free software." Brainstorming for this new label eventually converged on the term "open source", originally suggested by Christine Peterson. (Open Source Initiative, 2012)

Being motivated by this small conference, Eric Raymond spread the word about his philosophy and tried to gather more supporters. He got an approval from Linus Torvalds who was convinced by his philosophy. In addition to that, he got support from many other computer professionals working for Free Software Foundation. Even Richard Stallman seemed convinced by Raymond initially to adopt the term ‘open source’ but later on rejected it. (Tiemann, 2006; Stallman R., 2012; Williams, 2002).
Raymond and his friends were making all possible efforts to market the term open source and they were getting success too. But, soon they felt that to market the term ‘open source’ in a better way, they badly required a platform. Hence Raymond decided to form ‘Open Source Initiative’ with the help of Bruce Perens, a colleague of Richard Stallman and member of FSF, and Tim O’Reilly, another supporter of FSF. In February 1998, Raymond formally declared the launch of Open Source Initiative (OSI) to regulate the term ‘open source’ by providing a definition of this term for the people interested to make their program open for the public. His call to support the term ‘open source’ was as following:

“After the Netscape announcement broke in January I did a lot of thinking about the next phase -- the serious push to get "free software" accepted in the mainstream corporate world. And I realized we have a serious problem with "free software" itself.

Specifically, we have a problem with the term "free software", itself, not the concept. I've become convinced that the term has to go.” (Raymond E., 1998)

Eric Raymond was the first President of OSI and Bruce Perens the first Vice President. OSI, with its initiation, provided its definition of Open Source Software (OSS) and criteria to decide the open source software. According to OSI, "Open source promotes software reliability and quality by supporting independent peer review and rapid evolution of source code. To be certified as open source, the licence of a program must guarantee the right to read, redistribute, modify, and use it freely" (Open Source Initiative, 1998).

In addition to open source definition OSI also identified ten criteria to decide open source software. In other way, these criteria are broader form of definition of open source.

- **Free Redistribution:** The first criteria to decide open source is free redistribution. The licence of software shall not restrict anyone from redistributing or selling the software as
a whole or a constituent of aggregate software. The licence shall not compel to pay any royalty or fee against the software, its redistribution or sale.

- **Source Code:** Source code is a prerequisite for open source. To be identified as ‘open source’ software, a program must comprise its source and must permit the software in source code as well as in executable form. In case the software is not distributed with source code the developer must have publicized the source code from where it could be downloaded with the help of Internet, without paying any fees.

- **Derived Works:** Open source licence must permit alteration and derived work forms. In addition, the licence should also permit the redistribution of derived work under the same licence of original software.

- **Integrity of Author’s Source Code:** The licence of open source software may limit source code from being distributed in modified form if, the licence allows the distribution of ‘patch files’ with the source code for the purpose of modifying the program at build time. However, the licence must allow distribution of software built from modified source code.

- **No discrimination against person or group:** The open source software licence must not discriminate against any person or a group of persons.

- **No discrimination against field of endeavour:** No group should be restricted by licence to make use of software for any specific field of endeavour. Anyone from any field must be free to use the software as per his/her requirement.

- **Distribution of licence:** The licence of the software must apply to all of them to whom licence is redistributed without adding any additional licence.
- Licence must not be specific to a product: The software licence should not depend or be restrictive to a specific software distribution.

- The licence must not restrict other software: The licence must not restrict the software distributed under other variety of licence. For example, it should not insist that the other software used with this software should be an open source software.

- The licence should be technology neutral: Open source software licence should not be restricted to the use of a particular technology or user interface.

(Open Source Initiative, 1998)

Since this declaration of OSI, the open source movement boomed up and was heard everywhere. However, the internal conflict between two groups, i.e., Free Software Foundation led by Richard Stallman and OSI led by Eric Raymond, on their philosophies continue and is still in existence.\(^2\)

### 4.2.6. Success of Open Source Software Movement

By early 21\(^{st}\) Century, open source software movement has observed a number of success stories that include Ubuntu and other versions of GNU/Linux Operating System, MySQL and Postgres SQL for creating databases, Firefox web browser, Apache web server and programming languages such as Perl, PHP, Python and Ruby. According to Web Server Survey 2013, Apache\(^3\) is used by more than 50% hosts while its nearest competitor is Microsoft Web Server which is used by 17.22 sites (Netcraft, 2013). As per Browser Statistics and Trends more than 80% used

---

\(^2\) To read more about the struggle of two groups please refer Free as in Freedom by William S. Sam, Introduction to free software by Jesus M. González-Barahona et al., and Open Sources: Voices from the Open Source Revolution edited by Chris DiBona, Sam Ockman and Mark Stone.

\(^3\) [http://www.apache.org](http://www.apache.org)
Web Browsers are open source, i.e., Firefox and Chrome (W3Schools, 2013). In survey of Statecounter also Firefox and Chrome are beating commercial web browsers (Statcounter, 2013). The latest version of Apache Open Office is 3.4 which was released in May 2012 and in a period of one year it has more than 50 million downloads (Open Office, 2013). These are few examples out of the huge world of open and free software. The use of these software is so wide that every computer of the world must be using one of these by one means or the other. These software are used not only by those who are using these because they cannot afford costly commercial software, but also by the larger organisations which are using these software just because of the quality provided by OSS. Some of the very large examples are Google, Amazon, Facebook etc., which must be in use by every user of Internet.

Since then a number of Free and Open Source Software (FOSS) came into existence from every field such as Banking, Accounting, Medical, etc. One can easily find suitable open source software not only for operating system but also for server, web server, database, data processing and many more. Although we may not be conscious enough about FOSS, but our Internet Service Providers use one or other free and open source software to provide Internet services. Additionally, majority of commercial library management software vendors also use free and open source software to build their product.

4.3. Open Source Movement and Libraries

4.3.1. Computer application in libraries

The development of computer application in libraries has been experienced since late 1950s and early 1960s to automate administrative, user and technical service functions. However the idea of an integrated library system was generated long back by Venn ever Bush (1945) in his paper ‘As
you may think’, which later realized in the form of COMIT, a programming language to retrieve a particular string of information, by Dr. Victor Yngve and others at MIT Lab in 1957 (Yngve, 1958). Another presentation by Herley E. Tillitt (1970), entitled “An experiment in information searching with the 701 calculator” at an IBM conference at Endicott, New York in 1954, also evidenced the existence of information retrieval program. Tillitt, a working professional at US Naval Ordnance Test Station (NOTS), applied library computerization that motivated the world to use computerization in libraries. A number of similar smaller projects of computerisation were running, to give the account of all of them here is very difficult, albeit the introductory chapter provides a gist of historical development in library computerisation. However, it is not difficult to understand that library professionals were very much aware and interested in computerisation of library functions since the very beginning.

1960s witnessed the development of MARC (Machine Readable Catalogue) after the several efforts of Library of Congress. MARC is still very prominent cataloguing standard worldwide. After these initial successes a wide number of efforts took place to create various applications to automate library work such as information retrieval, circulation, indexing etc. However these developments and innovation were taking place at institutional levels only. Well funded institutions of American, Austrian and European continents were increasingly getting involved in automation technologies. Library of Congress, MIT, OCLC, University of California and other were leading organizations in such programming inventions. One thing must be noted here that all the computer programmes developed during 1950s, 1960s were batch processing, they were not as simple as today’s integrated systems and librarians and computer programmers of those days had to work very hard to manage those computer programs.
Increasing interest of librarians in computerisation paved the way to establishment of several companies especially dealing with library computerisation, to name a few, Geac, SIRSI, Data Research Associates (DRA), CARS, Ex- Libris, PALS, Gaylord, and Innovative Interface, Inc. Libraries used to enter into a long time contract after buying a library system from these companies as switching the vendor was like starting from the beginning. The costing of the computer program and their management was very expensive and out of reach of libraries and institutions with scant finance. Development of computer programs to automate the library functioning, on the one side, eased the work of libraries, but at the same time a number of libraries of developing and under developed countries were lacking these programs as it was unaffordable for them to acquire costly computers and library automation programs.

With the passage of time, acquisition of computers became affordable and also indispensable for any type of organization or institute and so also for a library. However, having computer applications or library automation software and managing them were still unaffordable. In initial phase computer programs for library functions were not available as free product; libraries interested in computerization needed to hire the programmers or enter into an agreement with library computerisation vendor to design a program as per the need of a library which was still very costly.

### 4.3.2 Beginning of Open Source library systems

There were some free tools in existence for libraries. Most of them were developed by OCLC and Worldcat which were distributed free to its member libraries but were out of reach to non-member libraries (Clarke, 2000, p. 31). A free cataloguing utility named CDS/ISIS (Computerised Documentation Service / Integrated Set of Information Systems) and later
renamed as WINISIS (Windows Integrated Set of Information Systems) was developed by UNESCO in 1985 and distributed to libraries worldwide for free through their website.

The free software movement or open source movement was already in the lime light in 1980s nonetheless, libraries were still away from the concept of free software. During this decade UNIX was already installed on computers while Linux was about to be born. During late 1980s concept of free software was taking momentum and commercial vendors were looking towards free software as an opportunity to accomplish their commercial goals. This resulted into establishment of Cygnus Solutions⁴, the world’s first open source software company by Michael Tiemann with the help of David Henkel-Wallace and John Gilmore. Tiemann found business opportunities in open source as he states:

“Open Source would unify the efforts of programmers around the world, and companies that provided commercial services (customizations, enhancements, bug fixes, support) based on that software could capitalize on the economies of scale and broad appeal of this new kind of software.” (Tiemann, 1999)

By late 1980s and early 1990s the whole world had been looking at FOSS as an approach to save their money to be invested on commercial software or FOSS as a business strategy itself. However libraries did not turn towards it before late 1990s; instead, there were some proprietary software which ware using open source software as a component of their ILS.

4.3.3 Advocacy of Daniel Chudnov to open source software in libraries

In most of the possibilities, Daniel Chudnov, the most well-known advocate of the open source software in libraries, drew the attention of library professionals towards open source software through his historical writing ‘Open Source library system: getting started’ in February 1999 on a

⁴Cygnus Solution is company that has developed ‘Red Ha:’ another open source version of Linux.
website (http://www.oss4lib.org) created by him to provide open source software to library professionals (Morgan, 2012; Clarke, 2000). In August, 1999 he republished it in ‘Library Journal’ with modified title ‘Open source software: the future of library system’, that had done enough to bring the term to the attention of library community (Chudnov, 1999a).

Daniel Chudnov (1999b), in his writing encouraged the library professionals to draw their attention towards open source software by elaborating the benefits of open source software and equalizing the philosophy of librarianship and OSS. He suggested that using open source software the libraries can cut the costs involved in development and management of proprietary software in library as well as the libraries could modify the codes as per the requirement of the library. He says “We are an educational institution, and we are here for people to learn about computers. That should include learning how the software on this computer works.... Libraries should actively discourage the concealment of generally useful knowledge, and that includes proprietary software." (Chudnov, 1999b). This writing by Chudnov became a milestone in the field of OSS in libraries. He was appreciated by a number of library professionals for his efforts in this writing and starting OSS4Lib in early 1999 (Clarke, 2000; Morgan, 2012). However, before Chudnov’s writing also there were a few examples of efforts made towards using OSS; some of them were cited by Chudnov in his writing. Some of them, for example are, use of Linux, Apache, MySQL and PHP (LAMP) for managing information about online resources and selected subject-based Internet resources at Yale's Cushing/Whitney Medical Library (Chudnov, 1999b). Further, yet another example is the enunciation of ‘Keystone Principles’ by ARL (Association of Research Libraries) leaders who acknowledged that “libraries were responsible for creating innovative information systems and advocated for the development of open source
solutions” (Jaffe & Careaga, 2007). According to Chudnov (1999b) the following factors and situations insist libraries to use OSS.

- OSS permits libraries to reduce the costing on library software that can be utilised in other developmental activities in library.

- A library wishing commercial support can also adopt an OSS as there are several organisations which provide technical support for OSS.

- Using OSS, the library does not prohibit itself with a single vendor. A library at any time may change the vendor for technical support.

- Being developed by an open community, there are several online support options available for OSS.

- Library professionals could also build their OSS project and make it open for the library community to improve it.

4.3.4 Dissenting voices to open source software in libraries

This writing of Chudnov led to several reactions towards use of open source software in libraries, some of them supported OSS while a few perceived OSS as “still only a distant possibility” (Breeding, 2002). Marshall Breeding (2002) produced his arguments why OSS in libraries could not be a reality soon:

- A movement of libraries, including Library of Congress, from self-developed ILS to proprietary software ensured that they do not have resources to develop and maintain library automation system. Then how one can expect development of an OSS.
Proprietary ILS developed by commercial vendors are result of research of more than five years and their software are having program codes of millions line. They have great teams of computer programmers and development of an OSS is a million dollar project. It is hard to see that even a large library would have all these facilities to develop a large scale ILS.

Technology is developing so fast that even commercial ILS developers with their great teams find it difficult to cope with the demands of libraries. Hence how a team of open source developers with their limited resources could meet the ever rising expectations of library professionals.

A number of OSS enthusiasts are present but very few library administrators are ready to take risk of strategic reliance on OSS. Libraries lack the ability to fund the programming staff for their commercially-supplied system, and much less towards the development of OSS.

Marshall saw no future of OSS although Koha was already born by the time he was writing this. He did not see any “paradigm shift approaching where commercial companies yield to open source and free software.” He observed further, “The opportunity for libraries to develop Open Source applications in the digital library arena is narrow” (Breeding, 2002).

4.3.5 Open source software as an essential aspect for the development of library services

However, Draft Report (Digital Library Federation, 2001) of Digital Library Federation (USA) did observe OSS as an essential aspect for the development of Library services. The report puts their findings as under:
- OSS is an economical alternative to libraries' reliance upon commercially supplied software.

- OSS is essential if libraries are to develop software and systems that meet their patrons' needs.

- OSS ensures that library systems and online services will be more functional for libraries and their patrons and as such be good for library patrons.

Roy Tennant (2000) saw the opportunities in OSS but he felt that “unless the OSS application is a well-developed and stand-alone application such as the Apache web server, use of OSS will mostly occur in large libraries (of all types) that are more likely to have staff who can install and maintain the software”. Similarly, Muir (2005) accepted that OSS is beneficial for libraries from the financial point of view and it gives opportunities to the libraries to modify them according to the need of the library but at the same time he raises doubts as to the availability of technical expertise and support in the long run to contain any bug or error. This turns out to be its single most drawback.

However, with the passage of time OSS proved the people who were skeptic regarding the acceptability of open source software in libraries wrong. In a very short period after Chudnov’s call to adopt OSS, a number of projects for open source library applications found their way. Additionally libraries started to shift from commercial to open source software. It was enough to change the minds of people who doubted the future of OSS. Roy Tennant (2003), after 3 years of raising doubts about the future of OSS in his publication, accepted that “open source is [going] into mainstream”. Similarly Marshall Breeding (2009) who saw open source ILS as a distant possibility agreed that “The open source ILS movement has progressed past the point where its
viability can seriously be questioned. The current momentum of open source ILS adoption makes it almost inevitable that it will represent an increasing portion of the library automation landscape”.

There are software which are developed purely on free and open source platform and the users need not spend even a single penny for these. Using free and open source software, one can easily provide high level technical facilities to the users and make the library automated. However, the libraries who find it convenient to obtain commercial support for their ILS may also obtain the same for OSS. Today a number of vendors exist who provide support for OS ILS. Taking commercial support, for those who have enough financial resources and wishing to hire a vendor, for open source ILS again will save the libraries from being blackmailed by the vendors of commercial ILS as source code of open source is always open and changing the vendor may not harm the activities and functionalities of the library in any respect. Availability of commercial vendors for OSS is an answer to the skepticism of Muir as when he said “who officially provides support: after all, there is no vendor to complain to, and this could make it harder to ensure improvements and fixes are made” (Muir, 2005).

4.3.6 OSS4Lib and Code4Lib

Development of OSS in LIS was started for tny LIS tools by distributed efforts of libraries spread over different corners of the world but today it has integrated as a crusade at International level and approach of the library professionals toward open source software can be considered as a movement in the field of Library and Information Services. Today for almost every function, an open source application exists. OSS4Lib (http://www.oss4lib.org/) and Code4Lib (http://www.code4lib.org/) are two major websites and communities that are dedicated for the
development and distribution of OSS for the use of libraries. OSS4Lib was developed by Daniel Chudnov in 1999 with the mission to “build better and free system for the use of libraries” (Chudnov, 2005). This site maintains the list of the OSS, programming languages, and protocols and standards for libraries and keep tract of the news of OSS in libraries. OSS4Lib has listed over 500 library specific application under more than 15 categories for various functions of libraries. These applications are (as on 23rd May 2013) listed under following categories:

- ADA (1)
- Bibliography (63)
- ILL-DD (19)
- ILS (67)
- Image Processing (15)
- Information Retrieval (47)
- Library Administration (12)
- Metadata (56)
- OPAC (92)

- Readers' Advisory (30)
- Reference (13)
- Repositories (39)
- Reserves (7)
- Resolvers (5)
- Text Processing (19)
- Training (4)
- User Access Management (12)
- Workstation Management (5)

(OSS4Lib, n.d.).

Code4Lib was initiated by a group of library programmers in 2003 with an objective “to providing a harassment-free community experience for everyone regardless of gender, sexual orientation, disability, physical appearance, body size, race, or religion” (Code4Lib, n.d.). Code4Lib also organises annual conferences on OSS since 2006 and continues it till now.

4.4. Open Source Software Projects for Libraries

After the above discussion on the expansion of concept on OSS in libraries, it is necessary to have an in-depth look at prominent open source projects in libraries. The upcoming discussion covers the prominent open source projects, the surrounding environment, and responsible factors for their development. However, it will not cover the technical analysis of the software itself as it
will be covered in Chapter V. Open Source Software projects in the following section primarily covers open source ILS, open source digital library software and open source content management software.

4.4.1. Open Source Library Management Systems

4.4.1.1. Koha (Open Source ILS)

Koha\(^5\), claimed to be the first open source library management system, traces its origin to the 1999 in New Zealand. Koha is a result of the frustration of the library professional with the commercial library management system. A group of public libraries of Horowhenua Library Trust (HLT) in New Zealand felt the need to change their commercial ILS, Catalist, as it was not complement with Y2K\(^6\) issues. The library was not able to do anything with the software as it was a closed source software and they were forced to replace their ILS with a new ILS. The other problem with Catalist was that it depended on modems to maintain their network which was increasing their telephone line cost (Anctil & Behcshti, 2004). Instead of going for another commercial ILS, HLT decided to develop their own ILS suitable to their needs. HLT Librarian Rosalie Blake did all that was required to convince the council to get exceptional circumstances grant for replacing the current ILS and she succeed in it. However, the approved fund was 50% of the actual cost and rest had to be managed by the library itself from their annual budget (Ransom, Cormack, & Blake, 2009).

Being much cost conscious, they sent RFP (Request For Proposal) to several vendors but the proposals were either costly or could not comply to the library needs. They had very limited

\(^5\) To monitor project koha, refer http://koha-community.org/
\(^6\) Y2K (Year 2000) was a problem related to the dates and due to this error computer read every year of 21\textsuperscript{st} century as year of 20\textsuperscript{th} century. For example is somebody types 2001, the machine describe it as 1901 while some computers were reading 2000 as 19100. Actually this was created as a solution to save the disc memory but at the end of the century it become a vulnerable issue.
options left, first, increase the budget which was near to impossible, second, to compromise with their needs which they did not want to do, and third that they write an ILS of their own. Having Katipo communication, an IT firm, in their partner list, increased their motivation to choose the last option. They gave this responsibility to Katipo Communication to program an ILS for HLT. Work of ILS started in September 1999 and was completed by January 2000, however, it was available to the world for download only in July 2000 as the first release (Koha Community, n.d.). Katipo Communication has suggested that the software will release under GNU General Public Licence (GPL) as an open source software so that the project would have longevity. HLT accepted this suggestion and its release was free for all.

Initially, it was named C4 (Cheap and Cheerful Copy of C) but then, after a lot of discussion, they adopted a Maori language term ‘Koha’ which meant ‘gift’ or ‘donation’ (Engard, 2010). “We chose Koha as the name, because it’s free and because it’s our gift to the world” (Ransom, Cormack, & Blake, 2009). The reaction of developers was that Koha would have a long and bright future as the world’s first open source integrated software. Though, the earlier started project on open source ILS was reported as Avanti ILS project begun in 1998 by Peter Schlumpf, it could not be as complete and successful as Koha.

Initial release of Koha was enough for HLT libraries though it did not have several features such as Z39.50 and MARC. After publishing its release on Koha’s mailing list, people around the world started showing their interest in it. Some school libraries in Coast Mountains School District of British Columbia in Canada started using Koha with the help of Steve Toneson, the network engineer of the district. The worldwide interest in Koha rose only after its adoption by Nelsonville Public Library (NPL) in Ohio, USA. NPL also funded to develop Z39.50 and MARC feature in Koha to make it more advanced and viable. Slowly and gradually a number of libraries
moved to Koha from closed source software. The constant move of libraries towards Koha also generated the opportunities for commercial vendors to establish themselves as Koha technical supporter. Some of the commercial vendors are Koha Water Solutions, Liblime, and Equinox. Presently Koha is running in its 3.10.6 version.

4.4.1.2. Avanti (Open Source ILS)

Avanti is the earliest started project on an open source ILS by Peter Schlumpf in 1998. Nonetheless the project had very small success until the end of 2004. This is what gave chance to Koha to be acclaimed the world’s first open source ILS. Avanti is dependent on a single man’s effort as it does not have any supporting community of developers. By the end of 2008 it could have only a cataloguing and patron accessible catalogue module. Its capacity is limited and it is configured to catalogue and search up to 16,000 titles and 32,000 items (Schlumpf, 2008). The programming language of the software is java and it is compatible to run on Linux and Windows. However it failed to fascinate library community and no library has reported using Avanti MicroLCS (Ancil & Beheshti, 2004).

4.4.1.3. Evergreen (Open Source ILS)

Evergreen, another promising open source ILS project, took its way in 2006 at Georgia, USA. Georgia Public Library Service (GPLS), is a unit established by University System of Georgia with the objective to “empowering libraries to improve the life of Georgians” (Georgia Public Library Service, 2013) through dissemination of information and encouraging reading, learning.

---

7 http://www.bywatersolutions.com
8 http://www.liblime.com
9 http://esilibRARY.com
10 To keep track of the Avanti project visit http://www.avantilibrariesystems.com/microlcs.html
11 Keep track of http://www.open-ils.org/ to know further developments.
and education with their continuing support and improvement of Georgia’s public libraries. They provide excellent information services to people and support the libraries through various programs. One of those is PINES (Public Information Network for Electronic Services). PINES provides automation support to the libraries all over the State. As on 15th January 2013, PINES has been providing its services to 281 public libraries in Georgia. The main feature of PINES is that a user registered at any library may borrow items from any public library in PINES Network (Georgia Public Library Service, 2013).

Initially, PINES library automation installations were built on the Unicorn ILS provided by Sirsi Corporation (now known as Sirsidynix) under a contract started in 1999 and remained in force until 2005. However, GPLS lost its interest in Unicorn only after initial few years and started looking for another alternative and they came up with the idea of developing their own ILS. In an open published letter in 2004, Lamar Veatch, State Librarian of GPLS, announced that they will not renew the contract of Sirsi Corporation beyond 2006 and will encourage an open source software development effort to create ILS for SPINE libraries (Veatch, 2004). The project to develop an ILS was started by a group of programmers of PINES in the year 2004 and they successfully switched to Evergreen by September 2006 (Breeding, 2006, p. 1). The latest version of Evergreen is 2.3.4 released in February 2013. Evergreen is configured to run on Linux only. Like Koha, success of Evergreen has crossed borders of Georgia and sprouted interest worldwide. As per the claims of Evergreen, it is used by more than 1000 libraries worldwide (Evergreen, 2013). Commercial support for Evergreen is available from Equinox Software, a company that developed Evergreen and provided commercial support.
4.4.1.4. NewGenLib (Open Source ILS)

NewGenLib\textsuperscript{12}, perhaps is the first effort to develop an ILS from a developing nation like India. NewGenLib (New Generation Library) first came into existence in 2005 with the joint efforts of Kesavan Institute of Information and Knowledge Management (KIIKM) and Verus Solutions Pvt. Ltd., both based in Hyderabad. The initial releases of NGL (NewGenLib) was commercial and was not available free of cost until 2008. On 9\textsuperscript{th} January 2008, both the partners decided to make NewGenLib as an open source software under GNU's General Public Licence. KIIKM is the primary source of finance for the development of NGL while Verus Solution took the responsibility to develop and promote it by engaging professional programmers. Verus Solutions provides technical support to install and make the software active at libraries willing to adopt NGL through online community and using desktop sharing systems. If the library wishes to have onsite support, NGL also has a provision for commercial onsite service. A number of libraries in India such as Bangalorc University Library, Indira Gandhi Institute of Technology, Birla Institute of Management Technology (BIMTECH) along with several other libraries in Asia and Middle East are using NGL. Verus Solutions is actively working on the development of NGL. The current version of NGL is 3.0.4 R2 that can be downloaded through their website.

4.4.1.5. Others (Open Source ILS)

A number of other open source ILS projects were in existence but many of them either could not be successful or could not be completed at all. Some of these projects are Openbiblio\textsuperscript{13}, OPALS\textsuperscript{14}, PhpMyBibli\textsuperscript{15}(PMB), Emilda\textsuperscript{16}, Invenio, Learning Access ILS (LA ILS)\textsuperscript{17}, iVia,

\textsuperscript{12} To remain abreast about this project keep track of http://www.verussolutions.biz.
\textsuperscript{13} More details of the project available on http://obiblio.sourceforge.net/
\textsuperscript{14}http://www.opals-na.org/ is the project sit of OPALS
\textsuperscript{15} To keep track of this project, follow http://www.pmbservices.fr
BiblioteQ, and ABCD. Openbiblio originated in 2002 in Spain with efforts of Dave Stevens and it is still under active development. Openbiblio has all modules required in a general ILS such as OPAC, cataloguing, circulation, patron, and reports. It is very much popular in small and rural libraries of Spain, Chile and surrounding areas. Openbiblio is recommended for those libraries which do not have possibilities to reach beyond a collection of 50000 volumes and those which do not require much advanced features. This software can be installed either on Linux or on Windows.

OPALS (Open Source Automation Library System) is another successful software born in USA in early 2000s and was used mainly in School libraries. It is considered to be a perfectly suitable open source ILS for school libraries. It covers all required modules from the point of view of a school library. However, what makes it strange is that though it is an open source ILS, one cannot download it and install it oneself. It is an open source but not free from cost. OPALS is available in SaaS (Software as a Service) model through cloud computing. One has to pay an annual subscription cost towards hosting of the software, although they do not charge for the software. Mediaflex\textsuperscript{18} is a vendor involved in installation, hosting and management of OPALS. Due to SaaS model and non availability of the installation files, the software is hardly installed out of USA. The software claims itself to be an open source software but non-availability of binary and source code to download violates the conditions to be called an open source software.

PhpMyBibli (PMB) is another open source ILS developed in France in 2002 by librarian, François Lemarchand. Currently it is managed by PMB Services. PMB is another cross platform software and can be easily installed on Linux or Windows. This software is also suitable for

\textsuperscript{16} Project site of Emilda is http://emilda.org/
\textsuperscript{17} Project LA ILS can be tracked from http://www.learningaccess.org/ils-system/
\textsuperscript{18} http://www.medialflex.net
small libraries. It does not have a module to manage serials; hence, it is less popular in academic libraries where serials acquisition is an important and broad process. Initially it was licensed under GNU General Public License but presently it is another similar licence CeCILL (CEA CNRS INRIA Logiciel Libre) born in France.

Emilda is another open source ILS having very few modules such as circulation, OPAC, and cataloguing. It can be installed on Windows and Linux. Due to its limited modules and complex installing system it is not adopted in many libraries. Learning Access ILS (LA ILS) was ILS developed in 2000 in the state of Washington, USA. Earlier it was named as Koha West and Openbook. This software is also in use in public and school libraries. This software is declared as an open source software but they yet have to release its source code formally. ABCD (Automatización de Bibliotecas y Centros de Documentació) is a recently launched open source ILS. Its launch was declared by BIREME (Biblioteca Regional de Medicina), a health organisation in Brazil in 2009. ABCD is adopted mainly in libraries of Latin America. To make the software easy, BIREME has started a wiki\(^\text{19}\) that provides required documentation on installation and management of ABCD. The most important feature of the software is that it is based on ISIS software model, developed in 1970s by UNESCO that provided the software without licensing cost; however, this model provides only executable files of the software. Presently ABCD is funded by BIREME and Govt. Brazil.

\subsection{4.4.2. Public Knowledge Project}

Public Knowledge Project (PKP) is a non-profit project established by Dr. John Willinsky in 1998 at the Department of Language and Literacy Education, Faculty of Education, University of

\footnote{http://bvsmodelo.bvsalud.org/php/level .php?lang=en&component=27&item=13}
Columbia, Canada with an objective to improve scholarly and public quality of research (Public Knowledge Project, n.d.). Initially PKP was working in area of open access literature by encouraging open access journals. Soon after initiation of his project, he realized that open access journals also required to be managed online with the help of some software and there is no such open source software available and PKP decided to work to develop an open source software to manage online journals. With the efforts of programmers of PKP, they came with an open source online journal management system known as OJS (Open Journal System) in 2001. It turned out to be very successful and famous. Currently it is running its 2.4.2. OJS facilitates all functions required to manage a journal such as submission, author and editor registration, distribution of manuscripts to reviewers, etc. According to PKP website, as of October 2012, OJS has been used for 14, 700 titles around the world (Public Knowledge Project, 2012). On the line of OJS there are some other software developed by project PKP. These are Open Conference System (OCS) developed in 2002 and used to manage a conference online. OHS (Open Harvester System) developed in 2003 allowed to create a searchable index of metadata from OAI compliant archives. Yet another software OMP (Open Monograph Press) is used “for managing the editorial workflow required to see monographs, edited volumes and, scholarly editions through internal and external review, editing, cataloguing, production, and publication. OMP will operate, as well, as a press website with catalogue, distribution, and sales capacities” (Public Knowledge Project, n.d.b). OMP was released on 17th September 2012 and is the newest software released by PKP. There are several conferences held by PKP to increase the awareness about open access literature and scholarly publishing throughout the world to boost the effect of scholarly publishing.
4.4.3. Open Source Digital Library Software

4.4.3.1. Greenstone (OSS Digital Library Software)

Greenstone is world’s first open source digital library software developed in New Zealand at University of Waikato. The roots of Greenstone development were traced to the establishment of New Zealand Digital Library Project in 1995 with an initial collection of 50,000 documents. This project was supported and funded by the New Zealand Lotteries Board and operating funding from the New Zealand Foundation for Research, Science and Technology (Witten & Bainbridge, 2007). With New Zealand Digital Library Software, a fully searchable CD ROM Database could be produced. After production of CD ROM database for an NGO, The Zealand Digital Library Project team also produced a CD ROM Database for UNESCO. By the end of 1997, the software was renamed as ‘Greenstone’ which is valued more highly than gold in Maori language. In addition to that, it was released as an open source software under GNU GPL licence. The initial open source release was distributed in 1998 at greenstone.org (Witten & Bainbridge, 2007). Greenstone is a full-fledged Digital library software with the capacity to create a digital library on Internet, Intranet and produce a CD ROM based database. The development of Greenstone is undertaken by University of Waikato, New Zealand and it is promoted by UNESCO. The current version of Greenstone is version 3.0.

4.4.3.2. Dspace (OSS Digital Library Software)

Dspace is another open source digital library project. Dspace is designed to capture, store, index, preserve, and redistribute the digital literature. The idea of Dspace was born in MIT libraries in 1997 when the need for a digital library software was noticed. This need raised conversations among MIT librarians and finally in a meeting with Hewlett Packard Labs in 2000, a contract
was signed for software development plan. The 1st version of Dspace was released after the signing of this contract in 2002. It was released as an open source digital library software under BSD licence. Dspace is a very simple and effective digital library software which even a non-technical person can manage effectively. It follows all major international standards such as OAI-PMH, Dublin Core, and CNRI etc. In less than one decade Dspace became very popular among libraries throughout the world. As on 15th June 2013, 1450 users have been registered on dspace.org (DSpace, 2013) while according to OPENDOAR it is mostly used by open source digital library software in the world (OpenDOAR, 2013). Current version of DSpace is version 3.1.

4.4.3.3. Eprints (Open Source Digital Library Software)

Eprints is another enthusiastic open source digital library project started by Steven Harnad. Eprint came into existence in 2000 as an outcome of Santa Fe meeting in 1999. Eprints is developed by University of Southampton, UK and released under GPL License. Similar to DSpace it also supports OAI-PMH and other international standards. Presently it is ranked among the most widely used open source software.

4.4.3.4. Others (Open Source Digital Library Software)

Besides the above widely used open source digital library software there are several other projects available for digital library creation. These are CDS- Invenio, DoKS, Fedora, MyCoRe, and SOPS (SciX Publishing Services). CDS-Invenio\(^2\) (CERN Document Server - Invenio) earlier known as CDSware is a project of CERN laboratory released under GNU General Public

\(^2\)http://invenio-software.org/
License in 2002 to run CERN Database Server. DoKS\textsuperscript{21} (Document and Knowledge Sharing application) is another open source digital library software project initiated by Katholieke Hogeschool Kempen (KHK) with an objective to develop an application to organise electronic thesis and curricula vitae of students of Flemish University Colleges, Belgium. The project took its way in 2002 and funded by Institute for the Promotion of Innovation by Science and Technology in Flanders, private industry partners and non-profit organizations (Baccarne, 2007). The current version of DoKS is 1.4.1. Another project on open source digital library software is Fedora\textsuperscript{22} developed at Cornell University with the financial help of DARPA and NSF. The first version of Fedora was released in May 2003. Fedora is repository type digital library software that is capable to store and disseminate several types of electronic documents to the user. Similarly MyCore\textsuperscript{23} was released in 2004 at Essen University, Germany and SOPS\textsuperscript{24} was also released in 2004 with the financial help of European Commission (EU). However, the open source digital library software which are used comprehensively all over the world are DSpace, Greenstone and Eprints.

4.5. Findings and Interpretations

4.5.1. Major Findings

From the detailed discussion of the open source movement in the earlier sections, the major findings are summarised below.

\textsuperscript{21}http://sourceforge.net/projects/doksproject/
\textsuperscript{22} http://www.fedora.info
\textsuperscript{23}http://www.mycore.de/content/below/index.xml
\textsuperscript{24}http://www.scix.net/sops.htm
• In 1952 when IBM launched the 701, first commercial computer there was no difference between hardware and software; all computers were sold with free software of one kind or other with their codes.

• In 1970s there was a software sharing community to share and help the users who were facing problems or bugs in using operating system ITS developed by MIT.

• The history of free and open source software has two aspects, namely, development of UNIX operating system and the development of GNU project of Richard Stallman.

• The presentation of a paper on UNIX at the ACM Symposium on operating systems in October 1973 by Ritchie and Thompson and the subsequent publication of their paper in conference proceeding in July 1974 created a landmark in the development of UNIX as it resulted in generating a number of requests for copies of UNIX.

• UNIX Users around the globe were working on developing and improving UNIX but release of a number of derivatives and versions confused and annoyed the users, leading concern towards an uncertain future of UNIX.

• The first Open Source License, a software distribution license, was developed at University of California known as Berkeley Software Distribution (BSD).

• The voice for free software distribution was raised by Richard Stallman, a scientist at MIT Labs., who went on to develop GNU an operating system that would be free for users not only to use but also to modify or redistribute. He wrote a software distribution licence known as GPL (General Public License) which is widely used even today and he also established Free Software Foundation.
• Linus Torvalds built a free UNIX kernel and in 1992 his kernel was combined with GNU and eventually became popular as LINUX.

• In 1997 Eric S. Raymond wrote the historical paper “Cathedral and bazaar” on production of free software. His paper created such an impact that Netscape decided to release the source code of Netscape Navigator to public.

• Chris Peterson of Foresight Institute coined the term ‘Open Source’ to refer to free software without any rivalry with commercial software.

• In 1998 the Open Source Initiative was formed to market the term ‘Open Source’

• By early 21st Century open source software movement had a number of success stories. Although computer application in libraries was witnessed from late 1950s, but computer programs for library functions were not available as free product and were very expensive.

• The concept of free software touched the libraries only in late 1990s.

• Daniel Chudnov, an advocate of the open source software in libraries, was responsible for bringing the attention of the library community to this concept.

• There were also some dissenting voices to open source software in libraries.

• Starting with tiny LIS tools like OSS4Lib and Code4Lib the library community is witness to promising and multi module Open Source Library Management Systems like KOHA, Avanti, Evergreen, NewGenLib etc.

• Public Knowledge Project (PKP) of John Willinsky is open source software to manage online journals.
Greenstone, DSpace, Eprints etc. are Open Source Digital Library Software to manage the digital literature in libraries.

4.5.2. Interpretations

Open source software has become an inevitable reality not just in library but any type of organisation and even in daily lives of people. The widely used open source software is GNU/Linux operating system which is used by almost every individual who uses Internet because all major applications used by people today use Linux as operating system of their servers such as Facebook, Google, Amazon, etc. Apache Web Server has been used by maximum number of websites in the world and has beaten all commercial web servers. Its nearest competitor is Microsoft Web Server. Among Web Browsers also, open source software is leading with Firefox and Chrome. Yet another famous open source software is Apache’s Open Office that claims to be very successful and prominent among the computer users throughout the world. These software are being utilised by computer professionals of almost every industry including library and information science.

However, open source software used specifically for libraries and information centres are Koha, Newgenlib, Evergreen, Dspace, Greenstone, Eprints etc. Additionally there are many other types of software that are used in the libraries. The open source software movement and libraries are closely related as philosophy of open source software to distribute free software reflects in the philosophy of libraries to disseminate information freely. That is why once Amarel (2008) rightly asserted that “libraries and the open movement mesh together logically.” Free Software Foundation follows the open movement as a philosophy and does not support proprietary software in any way while the Open Source advocates do not loath proprietary software if they
do not have a better open source software to replace the closed one. Even the most prominent open source advocates use proprietary software. Howard County Library, known for most optimum use of open source software, provides access to their OPAC through the computers using Microsoft and Apple’s operating system (DeGroff, 2008). But, open source software have secured their place in all types of libraries.

The estimation of usage of open source software and its impact on libraries is difficult. However, there are several literature available about success of open source installation in libraries from every corner of the world. In fact some studies show that open source ILS form a significant portion of automation industry (Breedings, 2013).

Open source software have proved efficient in every aspect such as price and support. Every open source software is supported through more than one community and through their developers as well. Most of the time the participants of the community are users of the software who are experiencing the software, hence, the bug finding and offering solutions are quicker than a commercial vendor. The new users get support from the community and soon get ready to support other new users. However, without good supporting communities it is difficult for open source software to develop. A few examples of such software are Avanti, PhpMyLibrary, Openbiblio, Emilda, etc.

Another benefit of open source software is that it is continuously under development; thus users can avail the latest features. Yet another very important benefit of using open source software is that it does not bind anyone to continuously use it. It gives freedom to migrate to any other software any time which is not an easy task in case of commercial software.
Continuous involvement of library professionals in open source software installation and its use helps them in becoming aware of the complexities of information technology and in comprehending the intricacies of software used in library automation. Some information professionals gain considerable expertise of software by using open source software that they themselves could develop or support the development of an open source software by way of providing programming help, bug finder or through helping other installations. Open source software is a philosophy and it joins the people around the world in a chain where each one is connected and supported by others. Lastly, it could be said that in open source world each one is a user, developer and programmer who supports this movement to the best of his/her capabilities.

OSS in association with open access\textsuperscript{25}, Web 2.0 and upcoming Web 3.0 shall be beneficial not only today but also in coming future. The most common forms of open access are open access journals and open access repositories that enable the society to access research results without paying any fees. Web tools, on the other hand, are online free tools that can be used to improve the quality of library services and to make the optimum use of Information Technology. Web 2.0 tools are available in various forms such as RSS (Really Simple Syndication), Wiki, Blogs, Social Network, Podcast, etc. Any library with all these tools with open access literature and open source software may provide high quality services to its patrons.

\textsuperscript{25} Open access is discussed in more details in chapter no. seven.
References


