1. INTRODUCTION ...................................................................................................... 3

1.1 WHERE THE WEB WAS BORN .................................................................................. 4
1.2 EVOLUTION OF WWW ............................................................................................. 5
1.3 CHAPTER SUMMARY ................................................................................................ 6

2. LIBRARY INFORMATION SYSTEM ........................................................................ 7

2.1 CERN LIBRARY ....................................................................................................... 9
2.1.1 THE SCIENTIFIC INFORMATION SERVICE ............................................................. 10
   The Document Management Section ...................................................................... 10
   The Users Services Section ..................................................................................... 10
   The Periodicals Section .......................................................................................... 11
   The Archives Section ............................................................................................... 11
2.1.2 THE DOCUMENT HANDLING SERVICE ................................................................. 11
   Library Support Section .......................................................................................... 11
2.1.3 ALEPH SYSTEM ................................................................................................. 12
2.1.4 LIBRARY PROJECTS ............................................................................................. 13
2.1.5 CERN DOCUMENT SERVER ................................................................................ 15
2.1.6 WEBLIB ............................................................................................................ 16
2.2 CSF OF THE LIBRARY INFORMATION SYSTEM ........................................................ 16
   Education and Technology Transfer Domain (ETT) .............................................. 16
   Direction of Library Digitalization ......................................................................... 17
   Trends in Data Handling and Information Technologies ....................................... 17
   Grey literature and the Scientific Information Support .......................................... 17
2.3 CHAPTER SUMMARY .............................................................................................. 18

3. INTERNET TECHNOLOGIES IN THE LIBRARY IS ........................................... 19

3.1 INTERNET SERVICES .............................................................................................. 19
3.2 WWW TECHNOLOGY ............................................................................................ 20
3.2.1 STATIC HTML .................................................................................................... 20
   HTML and its future ............................................................................................... 20
   HTTP protocol ........................................................................................................ 22
3.2.2 DYNAMIC HTML ............................................................................................ 23
   ActiveX .................................................................................................................... 23
   Scriplets .................................................................................................................. 23
   DOM model ............................................................................................................. 24
3.2.3 SCRIPT PROGRAMMING ....................................................................................... 24
3.2.4 SERVER-SIDE SCRIPTING AND THE WWW SERVER ............................................. 25
   CGI .......................................................................................................................... 25
   Proprietary Server Interfaces ................................................................................. 28
   Server-side JavaScript ............................................................................................ 29
3.2.5 LANGUAGES FOR WEB SCRIPTING ................................................................... 30
   PERL ....................................................................................................................... 30
   JavaScript ............................................................................................................... 31
   VBScript .................................................................................................................. 31
   Java ......................................................................................................................... 32
3.2.6 WWW APPLICATIONS ........................................................................................ 32
Portals ..................................................................................................................... 32
Search Engines ........................................................................................................ 32
Robots (Crawlers, Wanderers, Spiders) ................................................................. 33
Agents...................................................................................................................... 33
3.3 AGENT BASED TECHNOLOGY .................................................................................. 33
Software Agents ...................................................................................................... 33
Code on Demand vs. Remote Evaluation................................................................ 34
Mobile Agents Technology ...................................................................................... 35
3.4 CHAPTER SUMMARY .............................................................................................. 36

4. LIBRARY APPLICATIONS............... ERROR! BOOKMARK NOT DEFINED.

4.1 GoDirect .............................................. ERROR! BOOKMARK NOT DEFINED.

4.1.1 PROJECT ASSIGNMENT ................. ERROR! BOOKMARK NOT DEFINED.
Proposal of a solution ....................................................................................... Error! Bookmark not defined.

4.1.2 IMPLEMENTATION ............................ ERROR! BOOKMARK NOT DEFINED.
Link builder ........................................................................................................ Error! Bookmark not defined.
Indexing robot .................................................................................................... Error! Bookmark not defined.
GoDirect Data Files .......................................................................................... Error! Bookmark not defined.
Basic Data Files ................................................................................................ Error! Bookmark not defined.
Lookup Tables .................................................................................................... Error! Bookmark not defined.
Data Files Update ............................................................................................. Error! Bookmark not defined.
Search Rule Files .............................................................................................. Error! Bookmark not defined.
WWW input form ............................................................................................... Error! Bookmark not defined.

4.1.3 CONCLUSIONS .................................... ERROR! BOOKMARK NOT DEFINED.

4.2 BIBLIOGRAPHIC INFORMATION UPLOADER ...... ERROR! BOOKMARK NOT DEFINED.

4.2.1 PROJECT ASSIGNMENT ................. ERROR! BOOKMARK NOT DEFINED.
EDS ........................................................................ Error! Bookmark not defined.
INSPEC ..................................................................... Error! Bookmark not defined.
LANL ........................................................................ Error! Bookmark not defined.
UNCOVER ................................................................ Error! Bookmark not defined.
SLAC ........................................................................ Error! Bookmark not defined.

4.2.2 UPLOADER MODULES ........................... ERROR! BOOKMARK NOT DEFINED.
Record matching ............................................................................................... Error! Bookmark not defined.
Knowledge base ............................................................................................... Error! Bookmark not defined.
Mail-to-Aleph gateway ....................................................................................... Error! Bookmark not defined.
Standardization of templates ........................................................................... Error! Bookmark not defined.

4.2.3 IMPLEMENTATION ..................................... ERROR! BOOKMARK NOT DEFINED.

5. CONCLUSIONS AND SUGGESTIONS. ERROR! BOOKMARK NOT DEFINED.

6. SUMMARY ............................................... ERROR! BOOKMARK NOT DEFINED.

FIGURES AND TABLES ................................ERROR! BOOKMARK NOT DEFINED.

REFERENCES .................................................. ERROR! BOOKMARK NOT DEFINED.
APPENDIX
1. Introduction

Internet technologies went through a tremendous development during the past decade. This happened in particular thanks the invention of the World Wide Web, nowadays the most widely used service of the Internet, that has reached during this period a honorable success. This phenomenon has influenced almost all of the spheres of corporate world and its impact on the society is going to rise in the future with even more emphasis.

One of the main factors is that the Internet has been absorbed by the commercial sphere where a great variety of applications emerged. Electronic Commerce (EC), Electronic Business (EB) and Internet Banking are just a few examples that are now being discussed widely. The requirements on the Internet services grow together with this trend which pushes back on the development and standardization of information technologies.

The Internet also brings the opportunity for other spheres, non commercial, where there are applied different factors of organizational success, but with a certain abstraction there is an obvious equivalent. While commercial organizations compete on the open business markets, the academic subjects compete on the intellectual markets.

Naturally, the new technology also opens the opportunity to create business in places where it was not usual in the past. New companies are established and even new branches are founded and the scope of offered services is extending. The commercialization is an effect that brings a new quality to the society, however, it only is the secondary effect. Primarily the goals and objectives are to be recognized within the particular area, which eventually leads to a different actual strategy taken.

This work is focused on the library information systems. First, the critical success factors for library information systems are defined. In particular, the role of Internet/Intranet technologies is pointed out as a result of previous studies. The base of Internet/Intranet technologies is laid down and it is demonstrated on the variety of applications within the library information system. The main focus is brought to the application of WWW technologies and to the aspects of programming that are connected to the development of applications on the Internet. The described technologies are demonstrated on several applications that were implemented in the library information systems at the CERN Library, where I stayed during my educational stay.

This work is dedicated to the projects that I participated and is aimed to present my personal experience gained by this participation. I have concluded that the library IS in general still do not support fully the paradigm of digital library, but the new standards as the XML and the further standardization efforts will bring a new step forward. My opinion is underlined by the fact that at CERN these topics are being discussed continuously in cooperation with other international organizations (e.g. CNRI, W3C etc.). These conclusions and further suggestions are summarized in chapter 5.
1.1 Where the Web Was Born

CERN (European Laboratory for Particle Physics) is the world leading research center where the finest research in the particle physics and the nuclear physics takes place. CERN was founded in 1954 by 12 European countries. Today, the number of members reached 20, including countries of the Central and Eastern Europe. The Czech Republic joined the organization in the year 1993. However, the cooperation in fact is worldwide and scientists participating in the research come from more than 80 countries of the whole world. This makes the place truly international and intercultural.

The activities are focused on experimental and theoretical physics. Using large accelerators and detectors dozens of different experiments aimed to reveal the universe’s hidden secrets take place. Any description of one each deserves an extra attention exceeding the scope of this work. The main ones are located on the Large Electron Positron accelerator (LEP) placed in 27km long ring. In 2005 the accelerator will be replaced by the Large Hadron Collider (LHC) that will allow the researches to perform more detailed experiments than ever before.

The High Energy Physics (HEP) is very specialized area requiring the support of the lead edge technologies. The research could not be happening without a strong technological base involving electronics and electrical engineering, mechanical engineering, magnets, vacuum systems and information technology. Concerning the information technology, the main research concerns simulation techniques, image processing and reconstruction (physics data processing), networking/communications and real time systems.

The research has lead to many achievements that have exceeded the boundaries of particle physics and have contributed to the evolution in other spheres of the science as well as to the industrial development. Among others, the information science belongs to the most important ones.

Invention of the World Wide Web was one of these achievements. In a large community of scientists appeared the need for more efficient communication and data sharing. The need was that information had to be communicated to relatively high number of people within a short time compared to alterative media. Clients were operating in heterogeneous environment and remotely over WAN from their home institutes. In 1989 there was no efficient and consistent way to satisfy this need and this lead to creation of the first WWW browser downloading the HTML pages from the WWW server using the HTTP protocol.
1.2 Evolution of WWW

Within the ten year history of WWW is an observable evolution that can be summarized as a way to more interoperability and complexity caused with the dramatical growth of the number of Internet users and of their demands on this service. When we look at some milestones of this evolution, we can easily see the progress and way to more cooperation with other technologies. In fact the WWW technology today is connected to all other technologies incl. database technology, multimedia or component-based technologies with further attempts to global standardization such as XML. This trend observations approximate the future of this technology. In a little review, I have put down some important milestones starting early, but note that the true world-wide web term belongs first to the late 80’s:

1969  GML - Generalized Markup Language, IBM initiative to deal with its publishing problems. The syntax had introduced the tags that are still used in the same way embraced in <> and </>.
1989  NeXUS, the first WWW browser composed at CERN
       HTML created as a subset of SGML and extension of the links that enabled the hypertext-based structure of documents
1993  The first version of Mosaic browser (GUI) issued at NCSA – the first graphical browser that contributed significantly to the early expansion of the web technology and the web-boom in 1994
1994  The first International WWW conference held at CERN
       W3C consorcium established at MIT with objective to control further standardization attempts in the field.
1995  The first search engines
       Boom of the innovative technologies JAVA, JavaScript, ActiveX, VRML
1996  Standardized Stylesheets – CSS1 (cascading style sheets level one), a standardized way to detach the document contents from its formatting
1997  XML1.0 announced as W3C recommendation
1998  CSS2 (cascading style sheets level two), backwards compatible to CSS1, contains many new features
       DOM makes it possible for programmers to write applications which work properly on all browsers and servers, and on all platforms
1999  XSLT – Transformation language for the XML new generation language on the WWW, aiming to replace the original CSS and HTML
Generally we can observe the following technological stages:

Stage 1:
One way delivery of static documents, first in line-text mode, later on using the GUI HTML browsers.

Stage 2:
Interactive dynamic documents, client/server architecture.
The connection between the hypertext front-end and the database technology is provided by means of Common Gateway Interface (CGI) or Application Programming Interface (API) on the server side. The PERL programming language becomes popular representing an effective tool for the realization of a server-side script programming.

Stage 3:
On the client side, the ActiveX and JavaApplets technology is developed and the client script programming emerges (JavaScript). The CGI is being alternated with other technologies such as proprietary web-server extensions and servlets.

Stage 4:
Enhancement of architectures alternating the classical client/server architecture.
Involving the AI in the space of WWW. Intelligent agents and mobile agents technology.
Greater standardization for more interoperability in both web document handling and web technology domain.

1.3 Chapter Summary

In the first chapter the frame of the work is defined. The role of Internet, particularly of the WWW service is growing and the trend has a strong influence on both commercial and academical spheres. Invention and evolution of the WWW are presented, as well as CERN, the organization where the WWW originated.
Usage of the Internet technologies becomes critical for success of all spheres. This work focuses on the usage of these technologies in the library IS. The structure of the work and its goals are presented.
2. Library Information System

The trend in the modern library information system has two main streams. First the library itself heads towards the digital library, where all documents are available in electronic form and therefore do not necessarily need to be accessed in their paper form. In the ideal state, each user would have his personalized library on the desktop accessing any type of library service transparently. Second, the library tends to be less document oriented and more information and knowledge oriented. The library does not deliver the raw material anymore, but a compact service that is based on the human added value and has extensible character. These two issues define the main focus in an architecture and a design of a modern library system. In this part I will have a look at these aspects of the modern approach to the library IS and I will show its characteristics on the example of the CERN Library.

From the system point of view, the system architecture is based on the database technology. The core of the system consists of the library catalogue - the bibliographic information database that holds the bibliographic records consisting of standardized descriptive information about library documents. The bibliographic information database is usually delivered as a part of a sophisticated and specialized product. This product is tailored for needs of this type of data handling, including enhanced algorithms for database searches, indexing and keywording, large data manipulation in standardized data format and also administrative functions such as loan registrations. The complexity of internal data structure of bibliographic information is typically lower, on the contrary, the volume of stored and processed data is larger.

The trend, as pointed out in the first paragraph, is to provide the documents corresponding to the bibliographic entries in an electronic form. Let’s call the service that takes care about documents handling the document server. So the document server contains the electronic version of catalogued documents. At this point, the completeness is not 100% and the question is, whether it would be efficient to have all catalogued documents stored in a central database. Besides the document database, the server includes a complete set of services for document searching and browsing, data acquisition, document format conversion, library support and other specialized services.

Of course not all the documents catalogued in the bibliographic database are available on the document server or electronically in full text. The trend is to higher this ratio (electronically available documents)/(catalogued documents) to the maximum.

The barriers of achieving this are mainly the type of media used for distribution. Most documents are still distributed using the traditional media and therefore they are not often enabled for electronic access. The most digital documents are issued on disks with large capacities such as CD ROMs or DVDs or accessible via Internet using some secured connection. The Web Publishing domain becomes very strong and more documents will be provided via the Internet access. The chapter 4.3 is focused on the topics of Internet based accessing and Internet Publishing.
Another problem occurs with the documents that existed before the electronic media occurred on the list. Naturally, the archived documents and older documents are more problematic issue, because their electronic version is not available and they need to be scanned for purposes of electronic access. The process is slow mainly due to the volumes of documents that are enormous. The documents are being scanned retrospectively. Also the very old documents that are not allowed for public access due to their historical value are processed (if possible) so they can be provided in an electronic form. The Internet is an international environment. In order to use the potential of it and enable the worldwide market, the legislation concerning Internet has to be unified internationally. Also, the trust in the Internet has not still reached the point and the Internet as a medium seems to have a debt in this direction.

Since the vast majority of documents is still treated in their paper form, the library service is still strongly focused on the bibliographic records maintenance which is of the high priority. This fact determines the priorities for the library IS itself, but I will discuss some points concerning the opportunities and risks that arise from this fact in chapter 2.2 (Critical success factors in the library IS).
2.1 CERN Library

CERN Library has developed a system that encapsulates all core functions of the Library IS called CERN Document server (CDS). In this chapter I will characterize the organizational scope of the CERN library and the architecture of its information system. The figure-2.1 shows the library in the organizational context. During the projects I was involved in sections of Scientific Information Service and Document Handling Service.

![Fig. 2-1 Library in the organizational context](image)

The CERN library is specialized in handling the scientific information that is also known as grey literature. The department has also several scientific projects that are supposed to improve the library service, library interoperability and standardization within the HEP community. To name some scientific projects maintained currently: SOCRATES project deals with the automated indexing and keywording by applying a complex algorithms using artificial intelligence approach; or the LEXI project that aims to build a comprehensive encyclopedia of the HEP domain.

The CERN library is also a leader in the international librarianship community and is the active member of AILIS (Association of International Librarians and Information Specialists) which is an international association established as a cooperative body of CERN, ILO, ITU, UN, WHO, WIPO and many other international organizations. The CERN Library has also started to publish the HEP Libraries Webzine, which is a magazine published on the web and oriented on the management in scientific libraries.
2.1.1 The Scientific Information Service

The Document Management Section

This section takes care of the acquisition and provision of the HEP documents. The majority of documents gathered concerns the pre-prints, the conference proceedings, the theses and the books. Along with the existing documents, also information about future documents is included, such as conference announcements.

The main projects of the section are the collaboration with other institutes that are mainly the SLAC, FermiLab and DESY with attempts to optimize the common documents management. Currently the library catalogues and document servers are disjunct and the bibliographic data is being exchanged on bilateral basis. The main problem that has recently occur are the rising volumes of data that are being processed in this way together with the fact that some percentage is still processed manually.

The information system is expected to support the section mainly concerning the automation of its processes. The goal is to replace those manual processes that can be more efficient when automated. This concerns all database importation processes. The statistics on data processing between years 1997 and 1998 that refer to the evolution of volumes of data that are processed is showed in Appendix B.

The Users Services Section

This section specializes in the reference librarianship and activities that correspond to it. Further, the section controls the interlibrary loan management, reference desk and reading rooms management.

The projects of this section are dealing with topics of efficient information and knowledge delivery, data format standardization, solutions for library interconnection and optimization of the user support tailored for the scientific community. Being a „front desk“ of the scientific information services, a very strong emphasis is given to the analysis of the library users feedback. The innovative ideas that come from this feedback are analyzed and realized in numerous projects that lead to better service for the library users. More specifically, the time of response of the library IS is being lowered constantly by using modern means of IT and the quality of service is maintained. To name some of the achievements, the service has a compact users interface involving the full text access to certain documents and access to other types of documents such as graphical, video and multimedia documents.

All library services are being monitored. For the thorough survey, the questionnaire has been organized as a web form and its results are evaluated periodically. Also other web pages are under control of the section providing all useful information for the library users.
**The Periodicals Section**

The CERN library journal collection is very rich, contains more than 450 journal titles in different areas of science. Around 50% of journals are available electronically in full text and nearly 90% of journals have an electronic table of contents available. The main publishers that provide the electronic access to their databases in some form are the AIP, APS, Elsevier Sciences, IOP and Springer.

The section has organized the electronic shelves so any user can access the articles from their desktops and read/print the articles from their offices instead of making photocopies in the library reading rooms. This feature also has an impact on requirements of the IS that is supposed to update the shelves periodically, customize users needs and provide a user friendly access to the periodicals, particularly to their full texts.

**The Archives Section**

The CERN Archive was created in 1980 as part of CERN’s commitment to the CERN history study. In 1989 its mandate was extended to create the CERN historical and scientific archive. The archive serves as an information source for CERN management, as a place of research, and as a repository for documentation of the work of the organisation.

2.1.2 The Document Handling Service

**Library Support Section**

The actual IS/IT computer support for the library and scientific information services is provided by the library support section of the document handling group. This section controls the management of the whole library IS.

The system is build on the unix platform as shown on the figure 2-2. As the library system containing the library catalogue is used the ALEPH in version 330 produced and serviced by Ex-Libris, the company based in Israel. The CDS system was developed as a document server. The main components of the IS will be described in next chapters.

The main activities of the section are

1. maintenance of the library catalogue database, currently the ALEPH 330, planning to upgrade to ALEPH 500
2. maintenance of the document server
3. development of new tools tailored to the needs of the library users and library staff
4. scientific information sections computer and technology support
2.1.3 ALEPH System

The system provides the solution for the library catalogue (the bibliographic information database) and many services that together build up a complex system for the library catalogue management.

At CERN the system consists of several databases, one of these is the library catalogue, others function as support databases. The main catalogue contains records about books, periodicals, conference announcements and proceedings, pre-prints, e-prints, scientific comitee papers, video tapes and yellow reports. The Besides this main database, ALEPH system also contains other databases such as HEP Institutes, CERN Archives, CERN press cuttings and CERN photos. The testing database is present for the library staff to test the new features.

The ALEPH system itself had been created in Israel, however has a strong support for local differentes all over the world. The latest version issued recently is the ALEPH 500. The system has some extensible features, supports the MARC and Z39.50 standards (Libraries around the world can be quickly and effectively searched by means of a Z39.50 broadcast search function) and enables the access to the database using the WWW interfaces.

Z39.50 is an internationally recognized standard for information retrieval that specifies the rules and procedures for the behaviour of two systems communicating to search databases and retrieve information.

The programming interface ALEPH API is available that enables the developers to access ALEPH databases. ALEPH API provides an interface for an access to data stored in the ALEPH system. The basic functions are the select of data set from the database, get the record selected from the set and fetch the record from the database. (i.e. SelectScope(), GetField(), MultiFetch())
2.1.4 Library Projects

The description in previous chapters implies the need for a further development that would use the potential given by the current state of technology and enhance the system so it would comply with the demands. The following projects were recognized as the key projects:

1. CERN document server (CDS)
2. Web interface to the bibliographic information database (WEBLIB)
3. Electronic Document Submission (EDS)
4. The Link Manager application (SetLink, GoDirect)
5. The Agenda Maker and Conference Manager
6. The Bibliographic Information Uploader
7. Socrates, automatic indexing and keywording
8. Lexi Project
In this work I would like to describe closer the two projects that I participated, which are The GoDirect Link Manager and The Bibliographic Information Uploader.

The GoDirect project has originated in the year 1997 as an impulse from scientists accessing often the electronic journals. The main idea is to provide the access to any article from CERN pre-pain journal accessible electronically to the of the user independently of its location. The first prototype was done in the year 1997 and contained a dozen of most important journal titles including a cross-journal experimental search. The project header:

<table>
<thead>
<tr>
<th>Project</th>
<th>GoDirect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>12 weeks</td>
</tr>
<tr>
<td>People</td>
<td>2</td>
</tr>
<tr>
<td>Requirements</td>
<td>WWW based</td>
</tr>
<tr>
<td></td>
<td>link manager based</td>
</tr>
<tr>
<td></td>
<td>publisher independent</td>
</tr>
<tr>
<td>Versions(^1)</td>
<td>2</td>
</tr>
</tbody>
</table>

The Bibliographic Information Uploader project objective is to establish a modular system of bibliographic record importation. The project was assigned due to the rising both number of the information resources and number of imported bibliographic records in total. The initial six main sources were handled independently using separate programs. The idea is to have one generic tool that would be able to import any record in textual form.

<table>
<thead>
<tr>
<th>Project</th>
<th>Bibliographic Information Uploader</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>26 weeks</td>
</tr>
<tr>
<td>People</td>
<td>5</td>
</tr>
<tr>
<td>Requirements</td>
<td>Modular system</td>
</tr>
<tr>
<td></td>
<td>Defined data interfaces</td>
</tr>
<tr>
<td></td>
<td>Defined functions and current systems</td>
</tr>
<tr>
<td>Versions(^1)</td>
<td>2</td>
</tr>
</tbody>
</table>

The projects 1 and 2 are the main part of the CERN Library IS and I will describe them shortly in following chapters. The projects of „GoDirect“ and „Uploader“ are the subject of the chapter 4.

\(^1\) Number of versions in production apart from the prototype
2.1.5 CERN Document Server

The idea of CDS corresponds to the definition of a document server as defined earlier in this work. The purpose is to gather, store, maintain and provide a wide range of the available documents in full text - when textual or in another form – when graphical, audio, video, or multimedia. The types of available documents are:

- E-prints - electronic pre-prints
- Scanned Pre-prints - paper pre-prints that were scanned into the TIFF format
- Published Articles - typically not stored locally, but accessed at the publisher’s server
- Conference Proceedings - a few conference proceedings are also available

The architecture of the CDS system is showed on the figure 2-3. The system is running on the UNIX operating system.

Fig. 2-3 Architecture of the Library IS, implementation at CERN

As pointed out, the server does not store all full text documents but in some cases it provides it as a pointer to some other location (usually external) that can be accessed over the Internet. Therefore the CDS contains a set of services that enable this functionality. The services are:

- Various interfaces to catalogues, incl. the web interface
- Electronic Document Submission (EDS)
- The Link Manager application (SetLink, GoDirect)
- The Agenda Maker and Conference Manager
- CERN Conversion Server
- Holdings Management
2.1.6 WEBLIB

The web interface for all services of CDS is called the WEBLIB which has a comprehensible and user friendly navigation system. More than an interface, it should be called the library portal since it consists of many different types of interfaces to various services that are not always linked to the local databases only, but rather to the particular scientific domain. The system enables to search the databases using the search tool or by hypertext navigations by grouping certain categories together. In this case the categories are organized according to the type of document. Second the system contains a powerful searching tool with similar functionality as the search engine.

2.2 CSF of the Library Information System

The library usually functions as a part of broader society and does not stand alone as an independent subject. The strategy of library is bound to the global strategy of the whole community. The influence of recent trends in IS/IT has enabled the libraries to extend their services so they can be competitive in an open market and some of them already are. Generating business in libraries is a trend, however, it remains as a secondary issue. The primary value of a library service is visible in a greater context of the society and usually in a long term horizon. In this chapter I will discuss the main critical success factors for the library IS. As a library model I will discuss the Scientific Library in High Energy Physics Society (HEP).

Critical success factor is a factor that significantly influences the final behavior and results of the concerned system area [Vor97]

The CSF are to be determined in an early stage of system development in order to recognize needs and avoid risks connected to the projects that evolve. Within my work on the system analysis for several projects that took place in this field I have recognized several factors that have such characteristics, even when they are derived from the scientific library, they are applicable on any other library sooner or later.

Main factors and trends that influence the library systems are: Education and Technology Transfer, Direction of library digitalization, Trends in data handling and information technologies and Grey literature and the scientific information support.

Education and Technology Transfer Domain (ETT)

As I described in the beginning of the chapter 2, the educational (knowledge) object is the main target of the library function. In the scientific environment, these objects are extended and understood under a common term Education and Technology Transfer (ETT).
**Direction of Library Digitalization**

In the modern library, the documents tend to be stored using some flexible media. The trend is to switch from the traditional paper form to the electronic form. In the future, the whole libraries and their services will be mediated electronically, i.e. more efficiently.

**Trends in Data Handling and Information Technologies**

The library has to comply with new trends concerning the new data formats and standards created in the information technology domain such as XML. Particularly for the topic discussed in this work – the application of Internet technologies, it is important for the library to comply with the new Internet standards. The XML format is an example of a critical factor as a the data handling trend.

**Grey literature and the Scientific Information Support**

The “grey literature” as defined on the 4th International Conference on Grey Literature (GL’99) is that which is produced on all levels of government, academics, business and industry in print and electronic format, but which is not controlled by commercial publishers. Since this kind of literature in uncontrolled commercially, the libraries become an important mediators of this information.
2.3 Chapter Summary

The chapter defines the scope of Library IS, specifies its object and the architecture. The role of the CERN library was characterized and its internal structure was described. The CERN library has an important position in the scientific domain, first by serving the scientific field of the High Energy Physics, second by participating actively in the development of the library science field. The main contribution is the organization of the „Library Science Talks“ that are organized usually four times a year as a cooperation with other entities gathered in the AILIS association.

The organizational structure of the CERN library is shown and the four core sections are presented – the document management section, the user support section, the periodicals section and the archives section. The IS/IT support is controlled externally by the specialized sections within the document management group. Besides the maintenance function, the section has a strong development activities and the main projects are described. The architecture of the CERN library IS and its main components (ALEPH library catalogue, CDS document server and WEBLIB) are explained.

The critical success factors of library IS were determined as follows:

1. Education and Technology Transfer
2. Direction of library digitalization
3. Trends in data handling and information technologies
4. Grey literature and the scientific information support

The corresponding technological issues were implied. These issues will be described in detail in the following chapter.
3. Internet Technologies in the Library IS

In the Library IS the variety of Internet applications is growing with new services that are being established. The main application remains the library catalogue access that using the Internet can be performed from distant locations as well as from within the local system. The original systems were based on the terminal access that used the Telnet session with an invoked program on the server that mediated the actual database access. This service was text oriented and required the knowledge of the manipulation language used in the mediating program in order to communicate with it. In the second step, the programs were improved to be more user friendly, for example the ALEPH OPAC that works in the ALEPH system and its databases (see chapter 2.1.3). The significant change comes with the WWW and its GUI that makes the system much more user friendly. In addition, the WWW system enables to extend the functionality of the interface from the basic functionality of library catalogue searches and browses to interoperability between the catalogue and other resources available in the HTML or otherwise via the WWW (e.g. remote databases). The great advantage is that the user can electronically access the full text of documents together with the record in the library catalogue. Currently, the Internet services enable the interconnection of the library catalogues, full texts and library services, where users can manage their personal shelves, order documents, check their loans or have the new issues of their popular documents sent automatically.

In this chapter I will present the technologies, standards and protocols that are used for Internet applications with particular focus on the database access, World Wide Web service and building the applications on this services. The parts that correspond to the projects that I have been involved will be emphasized, such as script programming, creation of WWW applications and comparison of available programming languages. In the end I will describe some new technologies that are being implemented in the library IS, namely the agent-based technologies.

3.1 Internet Services

The Internet is based on the TCP/IP communication protocol. Instead of TCP is sometimes used the UDP protocol, which is quicker, because it does not require the confirmation of the packet reception.

```
+-----------------+  +-----------------+
| SERVICES        |  | Session          |
| TCP             |  | Transport        |
| UDP             |  | Network          |
| IP              |  +-----------------+
```

*Fig. 3-1 The TCP/IP architecture and Internet services*
The services build on top of this architecture are commonly called the internet services (or the Internet services). Among these services, the most used ones are the Telnet, FTP, SMTP, NNTP or NFS.

3.2 WWW Technology

3.2.1 Static HTML

*HTML and its future*

The HTML was established as a language for creating the hypertext documents that can be interconnected via the network. The standard is a simplified version of SGML and has many limitations that are now recognized as substantial. The main problem of HTML is a join expression of the document’s data contents and its presentation. Both are expressed in one HTML file that makes it difficult to maintain.

In newer versions of HTML it is possible to use some enhanced techniques that partially solve these problems. In the HTML 4.0 version the standardized stylesheet support is included, known as CSS (Cascading Style Sheets).

The true distinction between the data contents and its presentation and the true solution comes with the XML standard proposed in 1996. The styles, formatings and programming elements are expressed in a separate XML document that represents the stylesheet in a standardized form using the XSL language. Since XML is more strict than HTML, it allows more comfortable application development that is more efficient and portable. XML is a standardized format designed specifically for transmitting the structured data to the web applications. Consider the following record of an electronic journal in HTML notation:

```html
<DL>
  ...
  <DT>
  <img src="../red_ball2.gif" alt="o">
  <font color="red"><b>
  Physical review D online
  </b></font>
  </DD>
  <a href="http://prd.aps.org/">Full text (1997- v 55 - )</a>
  <br>
  <a href="http://prola.aps.org/browse.html">Full text archives (v 31 - 54 1985-96) </a><br>
  <a href="../format/showfull?base=CERCER&amp;sysnb= " > Library printed collection</a>
  <br>
  </DT>
  ...
</DL>
```
Using XML, the record would contain only data elements, i.e. all HTML tags such as DT or FONT would be filtered out:

```xml
<LIST>
  ...
  <JOURNAL>
    <TITLE>
      Physical review D online
    </TITLE>
    <FULL>
      Full text (1997- v 55 - )
    </FULL>
    <RESTRICTED>
      1
    </RESTRICTED>
    <FULLLINK>
      http://prola.aps.org/browse.html
    </FULLLINK>
    <LPCLINK>
      ../format/showfull?base=CERCER&amp;sysnb=
    </LPCLINK>
  </JOURNAL>
  ...
</LIST>
```

Comparing these two notations, one can clearly see that the XML structure is more readable and comprehensible. The source is divided into several parts, so data is not mixed with the control structures. Instead of saying, for example, display the image denoting certain attribute (e.g. the red dot stored in "../red_ball2.gif" stating that the journal is restricted) the record contains directly the attribute in the correct semantic form. The actual way of how the attribute is presented is not important at this level, but expressed by means of stylesheets and stylesheet languages such as the XSL.

The XML is a standard particularly important for the libraries and other institutes oriented on the structured data handling, since it allows the standardization of an exchange of any kind of information using the WWW service and process it in a standard way.
HTTP protocol

The protocol used for the transfer of an HTML page between the server and the client is the HTTP protocol. The client in this model is also called the HTTP agent. The most frequent client (or agent) is the web browser. The HTTP protocol uses the MIME standard.

The HTML page can contain other objects that are bound to it. During the transfer, these objects (e.g. pictures, audio or video) are downloaded in their own threads.

Some additional machine can be placed between the client and the server, which is called the proxy server or the firewall. Using these additional machines the technology enable to create physically detached subnets that are called intranets. The intranet, however, can be built on the logical level by application of certain techniques that enable to build the Intranet (especially extanet) over the Internet. The most frequent technique is the IP tunnelling.

The HTTP protocol uses a few methods only, the method POST, GET, HEAD, PUT, DELETE, LINK, UNLINK. The first three methods are used most frequently and are supposed to appear in each implementation of the HTTP protocol, whereas the others are very rare and not always supported.

GET vs. POST method

Using the Method GET, the script can be called directly from the URL using so called “canned queries”. The method is used only to request some information from the server (a file, an image or a database query). The amount of information passed this way is limited. Netscape/X on Solaris at CERN library (Sun Unix) and NCSA Mosaic/X on OSF/1 (Digital Unix) can pass about 8000 bytes of form data. This should be sufficient for most of forms processed, therefore method GET is widely used. When passed via the URL the query appears behind the „?” sign:

http:///bin/scriptname?par1=value1&par2=value2&par3=value3

The POST method, on the contrary, implies the form processing may update information on the server (adding or changing information in a database, placing an order, etc.) Using the POST method, the contents is passed to the script as an input file causing much wider range of data to be passed, since there is no limit on the length of a URL or the length of a command line or command line argument on the server.

GET: via QUERY_STRING and PATH_INFO
POST: via CONTENT_LENGTH
URL

The URL (Uniform Resource Locator) is the address of a resource located within the internet network. The URL enables the hypertext structure of the WWW. The format of URL is following:

<protocol>://[<user>:<passwd>@]<host>[:<port>]/<path>

Each protocol has pre-defined its port number, therefore it is not usually needed to enter the port number in the URL.

3.2.2 Dynamic HTML

Dynamic HTML is a set new features that were introduced by Microsoft’s Internet Explorer version 4.0. From the definition, dynamic HTML can be anything that makes the resulting page to be different regarding the time, place or some other factors or input values. First, the „dynamic“ is understood as an enhancement of the page presentation, by employing some practical techniques such as CSS and of course some graphical effects from whose the commercial pages take advantage. Second, the dynamic HTML is employed to alternate the matter – the information.

At first I would like to mention several object-oriented programming approaches, that are tightly bound to the dynamic attribute of the HTML.

ActiveX

ActiveX is based on the Component Object Model and therefore a feature of the Windows platform and is more general. For HTML it is important that the ActiveX code can be inserted into the document as uncompiled and will be interpreted upon the processing of that particular document.

Scriplets

Scriplet is a web page written according to certain conditions that allow it to be referenced from outside. The scriplet is an HTML document that behaves as an object that can share its methods with other documents in the same way as any other objects do. The scriplet object is called as follows:

<object

type="text/x-scriplet"
data="Scriplet URL.html"
>
</object>
**DOM model**

Document Object Model defines the standard set of objects that are supposed to be independent on the platform used. The document object model (DOM) is a simple, hierarchical naming system that makes all of the objects in the HTML page, such as images, forms, and even CSS properties, accessible to scripting languages like JavaScript. This way the object oriented approach is brought to the traditional HTML structure. If somebody wants to access an object using the DOM model, it has to be done using the specific way.

### 3.2.3 Script Programming

In the large field of Internet/Intranet technologies I have focused on the WWW technology. In this part I will focus on the topics of script programming that is the core part of the development on the WWW and creating WWW applications. These aspects were applied in the solutions of projects described earlier in this work and are important for the detailed descriptions of some of these projects later in this work in chapter 4.

The script is a program that is not compiled, but interpreted. The interpretation takes place repeatedly each time the script is called. This has some advantages and disadvantages depending on the technology, where it is used. Sometimes it is profitable to work rather with scripts than compiled programs. The scripts used in the WWW applications can be distinguished from several points of view (Tab-3.2), according to the place where they are executed, programming language, platform and environment where they can be applied and others. In the next chapters I will characterize the script programming with regard to these categories. Usually, the type of script determines the programming language and tools that have to be used, or at least narrows the set of choice. The decision for particular programming languages and tools is important regarding the future plans of the organization.

<table>
<thead>
<tr>
<th>place</th>
<th>server side script</th>
<th>client side script</th>
</tr>
</thead>
<tbody>
<tr>
<td>language</td>
<td>procedural language</td>
<td>object oriented language</td>
</tr>
<tr>
<td>platform</td>
<td>platform dependent</td>
<td>platform independent</td>
</tr>
<tr>
<td>compilation</td>
<td>pre-compiled</td>
<td>interpreted</td>
</tr>
</tbody>
</table>

*Tab. 3-2 Script types*

Client-side means that the script is executed inside of the browser (or generally, inside of any HTML interpret) on the side of a client. The script code can be either written as a part of the HTML document or stored separately and called. In the first case the script is contained in the SCRIPT element or it can be included as an attribute of some other element.
The SCRIPT element has following syntax:

```html
<SCRIPT LANGUAGE="LanguageName"
[TYPE="MIMEType"]
[SRC="filename"]>

//script code

</SCRIPT>
```

The other option is to keep the script outside and include only the file name that contains the script code. The advantage of external code storage is the possibility of creating the script libraries that enable an easier maintenance and better organization. Also, there is another difference that may be important for the final effect. Only the main script code contained in the SCRIPT element is executed immediately, together with the HTML page analysis by the browser. In all other cases the HTML page is first analyzed and then the script is interpreted.

As the script is always executed on the side of the client, the HTTP connection can be interrupted after the page with the code has been loaded to the client.

### 3.2.4 Server-side Scripting and the WWW server

Unlike in the previous case, the server side scripts are executed on the server. The HTTP protocol transfers the HTML page upon the script has finished the execution of its script. The majority of scripts are so called CGI scripts that provide the gateway between the HTTP protocol and other protocols and services. Some other technologies were developed with similar functionality and will be described later.

**CGI**

CGI is the most common way of creating dynamic documents on the WWW. The application of CGI is relatively easy, because it already is a part of the WWW server. This implies that the CGI script programming is a server-side programming. Its purpose is to interpret the calls stated in the HTML pages passed to the server, execute the particular scripts and return the final HTML to the client, which is usually the WWW browser.

Besides the fact that CGI allows to make the WWW server behave dynamically it also allows the data flow in the opposite way than only transfer of the created HTML page to the client. Therefore the CGI is widely used in applications where the interactive behavior is desired or when the data is supposed to flow from client to server.
The general sequence that follows the CGI call is as follows:

1. The HTTP server gets a request for a URL that points to a CGI script.
2. The script is invoked on the server side.
3. The method is distinguished and the data input is parsed and decoded (if any).
4. If necessary, the script can access the external data, files or access databases via some API.
5. The script can generate some data internally.
6. The data obtained in steps 3-5 is processed and the resulting page is produced in HTML.
7. The HTTP server passes the HTML document to the browser.

For creation of a CGI script it is possible to use a variety of programming languages. Most often they are PERL, C/C++, Unix shell, Visual Basic, TCL or Fortran.

Scripts are located in a special directory usually called cgi-bin. The name of this directory and other parameters are stored in a configuration file (e.g. srm.conf or httpd.conf). The following entries define two cgi-bin directories:

```
ScriptAlias /cgi-bin/ /user/local/etc/httpd/cgi-bin/
resp.
Exec /cgi-bin/* /user/local/etc/httpd/cgi-bin
```

The script is also expected to have a special extension that had been configured on the server side:

```
Addtype application/x-httpd-cgi .pl .cgi .sh
```

Location of the WWW server itself is stated as:

```
ServerRoot /usr/local/etc/httpd
```

As stated in the general sequence, the CGI has two kinds of inputs, the environment variables (UNIX) i.e. REMOTE-ADDR, HTTP-REFERER (where did you come from) and from the client input. Extra information can be provided by datafiles stored on the server. This datafile can be accessed via „extra Path Information“ – PATH_INFO environment variable + PATH_TRANSLATED.

The output of the CGI is always the HTML page that is sent back to the client as a result. The way how the page is obtained can vary significantly – e.g. existing documents can be varied or the pages do not need to exist at all.
Basic applications with CGI are: forms, gateways, virtual documents and server redirection.

**Forms**

In this case the following header has to be included in the HTML file:

```html
<FORM ACTION="script_name" ENCTYPE="encoding_type"
METHOD="GET">

... form body

</FORM>
```

where:

**ACTION** specifies the location of CGI script to be called

**ENCTYPE** is optional. Specifies the type of coding for POST data. Can have only two values:

- Application/x-www-form-urlencoded (default) - the data are URL-encoded
- multipart/form-data - multipart document

**METHOD** specifies the mode, either “GET” or “POST”. Both methods are supported, usually the parser is ready to handle both.

Example of data parsing procedure (PERL):

```perl
sub parse_from_data {
    local (*FORM_DATA)=@_;
    local ($request_method,$query_string,@key_value_pairs,$key_value,$key,$value);

    $request_method = $ENV{'REQUEST_METHOD'};
    if ($request_method eq "GET") {
        $query_string = $ENV{'QUERY_STRING'};
    }
    elsif ($request_method eq "POST") {
        read(STDIN,$query_string,$ENV{'CONTENT_LENGTH'});
    }
    @key_value_pairs = split (/&/,$query_string);
    foreach $key_value (@key_value_pairs) {
        ($key,$value) = split (/=/,$key_value);
        $value =~ tr/+/ /;
        $value =~ s/%([0-9A-Fa-f])/pack("C",hex ($1))/eg;
        $FORM_DATA{$key} = $value;
    }
}
```

**Gateways**
The CGI scripts including forms are often used as gateways to other services. Most frequently the interface is a gateway to the database. The model enables to publish the internal content of database worldwide and also gather information in the other direction with regard to the security aspects. The typical example of this kind of gateway can be the publishers database of full text articles accessed via the web. The gateway checks the accessing user for his identity and provides the document stored safely in the database.

The database access is the most frequent case, but the gateway can be used for variety of services. Among others, the mail gateway or

**Virtual Documents**

At the simplest, the web pages do not have to be stored as HTML documents. The pages do not actually exist, but are created repeatedly on demand and are therefore virtual. When such a document is requested, the script that is responsible for its delivery actually creates it „on the fly“ and sends it back to the client. This approach allows the particular page to have customized contents and reflect the individual needs of that particular request.

**Server Redirection**

It is possible to use the CGI script as a resolution machine that instead of creating the HTML page only returns the Location in the HTTP header. The client receives the instruction to repeat the action using the new URL that has been determined by the resolution mechanism. The client sends the request repeatedly until it gets the final document. The server redirection can be used as a mechanism for the link manager (see chapter 4.3)

**Proprietary Server Interfaces**

As an alternative to the standard CGI some web-server developers have included their own interfaces with an idea of the solution of some weak points of the CGI. In some applications the performance of CGI can drop. The CGI scripts execute externally what can be quite exhaustive concerning the server resources when the script is called frequently since a new process is created for each request.

The two proprietary server interfaces are discussed in this chapter, the Microsoft’s ISAPI and Netscape’s NSAPI. The disadvantage is that scripts developed to these interfaces can only be used within the particular server.
Server-side JavaScript

Another alternatives that are not bound to some web server and therefore are portable to other machines are based on the Java technology - the server-side JavaScript, JSP. The SSJS is an object oriented scripting language that compared to the CGI runs as an in process application which improves its performance. The core of the technology is the same as the JavaScript generally (JavaScript core). Although called the script, the code is pre-compiled into the executable *.web file. SSJS applications are independent on the browser but require the JavaScript enabled server to run.

The system also provides the API to the database, originally called the LiveWire. The interface enables to access variety of databases (Oracle, Informix, Sybase, DB2) or any database via ODBC and even at the same time. The technology has some attributes of a middleware, for example concerning the database connections it enables to perate with the „connections pool“ that saves the system resources and number of licences to the database.

The SSJS is used mainly for the Intranet/Extranet Development, Workgroup-Scale Applications for smaller systems (upto thousand of users). For large systems of thousands users and the whole Enterprise-Scale Applications the Application server is suggested.
3.2.5 Languages for Web Scripting

**PERL**

The PERL (Practical Extraction and Report Language) is a modern language that is an appropriate choice especially for the purposes of CGI scripting. The PERL is also called the scripting language because of its characteristics. The main advantage with PERL is that the script itself is relatively easy to maintain, therefore all necessary changes into the code are feasible in shorter time.

**PERL is an interpreted language optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information.**

This language was created for UNIX but is is possible to use it in an other environment as well. It was created in the year 1986 by Mr. Larry Wall, who maintains the continuous development. The PERL Interpreter is distributed as a freeware.

Compared to the other languages, PERL contains following features:

1. associative arrays feature (indexed with literal keys)
2. automated conversion between data types (among number types plus strings, scalar variables)
3. dynamic arrays (push, pop, shift, ......)
4. binary data conversion
5. extensive support for regular expressions
6. system level functions
7. list-oriented data types
8. modular architecture (various modules, e.g. database connectivity in )

For these characteristics, PERL seems to be a good tool for CGI script programming, text-processing scripts, database front-ends

The main contribution to the text handling optimalization is the feature of regular expressions (RegExp). This feature enables the programmer to handle string values more efficiently in a condensed syntax and is UNIX based (is shared by UNIX utilities). The regular expression has following structure:

```
$var =~ /RegExp/
```

The variable $var is compared to the Regular Expression and with TRUE or FALSE result. The actual power of regular expressions lies in the generalization of string values using specific control signs, for example:

```
. # One character
^ # The beginning of the line or string
$ # The end of the line or string
* # Repetitive character (optional)
+ # Repetitive string (mandatory)
```
So the condition can look as follows:

```javascript
if ($var =~ /^Title/)
{
  print "\n";
}
```

**JavaScript**

The JavaScript is being considered as a standard scripting language. It is based on JAVA and has been standardized by ECMA. The Javascript can be used both on the client and on the server. The SSJS (Server-Side JavaScript, also known as LiveWire) was created later, although the chronological order of the technology evolution was the other way around. This special case is solved by introducing a special element `<SERVER>` for this purpose that says that the code has to be executed on the server. The technique is used as an option for other alternatives of the CGI programming. The language originates from Netscape, however, Microsoft has implemented its own version that complies the norm ECMA which is called JScript. The language is supported by both Netscape and Internet Explorer browsers and seems to be a good choice for these reasons.

**VBScript**

VBScript is a product that is dedicated to the platform of Windows and runs on the MSIE with VBScript support. It is a subset of the Visual Basic and is upwardly compatible with Visual Basic for Applications. The tool includes many features such as using the OLE controls, ActiveX components, multimedia support, and other Windows services.

<table>
<thead>
<tr>
<th>Place</th>
<th>Langue</th>
<th>Platform</th>
<th>Compilation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERL</td>
<td>SS</td>
<td>procedural</td>
<td>cross-platform</td>
</tr>
<tr>
<td>JavaScript</td>
<td>CS/SS</td>
<td>OO</td>
<td>cross-platform</td>
</tr>
<tr>
<td>VBScript</td>
<td>CS/SS(ASP)</td>
<td>Windows, MSIE</td>
<td>Interpreted</td>
</tr>
<tr>
<td>Java Applets</td>
<td>CS</td>
<td>OO</td>
<td>cross-platform</td>
</tr>
<tr>
<td>C/C++</td>
<td>CS/SS</td>
<td>proc/OO</td>
<td>UNIX</td>
</tr>
<tr>
<td>shell script</td>
<td>SS</td>
<td>procedural</td>
<td>UNIX</td>
</tr>
</tbody>
</table>

---

2 OO = object oriented, CS = client side, SS = server side
**Java**

The revolutionary JAVA language was introduced by SUN Microsystems in 1996. The language is cross-platform and therefore well suits the needs of heterogenous Internet. The JAVA differs from JavaScript by being its superset. Unlike with JavaScript, JAVA enables creation of applications that run on the client detached in the reserved memory space. These applications are called *Applets*. Applets are downloaded to the client and then executed as a local application. The platform independence is based on the idea of a *byte-code*, which is a pre-compiled version of the JAVA source code. This byte-code can be then transferred over the network into any other node and interpreted. For the node to be able to perform the interpretation of a byte-code it is neccessary to have a JAVA interpreter included in its operating system – the JAVA Virtual Machine (JVM).

The multi platform architecture of JAVA trades off the speed, which is low because of high requirements on the system resources.

### 3.2.6 WWW Applications

**Portals**

The portal is a complex service that covers a set of WWW applications running below, such as search engines, subject catalogues or meta-search engines using the agent technologies. There are many well known commerical Portals that can be taken as an example. Generally, the portal is a web meeting point for everybody to find their needs. In the library IS the portal would be a parallel, having specialized on providing the library services.

**Search Engines**

The search engine is a tool for an efficient web-search based on the keywords in the same way as the library systems do, just substitute the library catalogue with the web space. The documents are the web pages that are spread over the web space. The difference is, that on the web the pages are not organized properly and therefore the results cannot be precise enough.

The searches are not performed real time, the indexing technique is used to monitor the web space. The engine then only uses the indexed information. For indexing itself must be developed another application, the indexing robot. In order to improve the results, the meta-search engines can be build, which is simply a cover over several search engines and a comparison of their results.
Robots (Crawlers, Wanderers, Spiders)

A web robot is an HTTP agent that works with the HTML documents. The robot is able to traverse the hypertext structure of the web space and in an automatic and efficient way find new resources and treat them in some way. Most frequently, the process is aimed to index the documents available on the web by recursive retrieval of located documents. This type of robot is called an indexing robot.

Agents

The Agent-based technology is a new paradigm that is being well used in WWW applications because of its characteristics. I will describe the details of these characteristics in the next chapter.

3.3 Agent based technology

The agent-based technology uses the software agents instead of regular programs to perform the tasks. On the Internet the software agents are used to handle the gigantic volumes of data on the Internet instead of the user. The agent technology is useful in that case, when the user has a complex personal task and is not able to complete it by traditional tools such as search engines and portals. In fact, agents are being included into these portals in order to satisfy the users needs. The most frequent usage nowadays is in the Electronic Commerce field (shopping bots, commerce bots), but it is expected that the technology will expand into many other areas including the library ISs. The characteristics of library information processes (as pointed out in chapter 2) can be optimized using this technology. The main points of such ideas are:

1. agents can be personalized, which enables the users to configure their preferences in detail. This can be used in the customization of library services such as reference services, alert services, personal shelves maintenance etc.
2. agents can be intelligent, which means they are able to handle not only data or information itself, but the knowledge. Therefore they are in a position of experts and they can be used as knowledge mediators in the knowledge-based Library IS.
3. agents can be communicative, that is, they can share the information (or knowledge among themselves) by constituting the multi-agent systems and by performing tasks commonly called „Personal Assistance“

The project that suggests the usage of agent-based technology in the library IS has been proposed at CERN Library as a mobile agent system and will be described later in this work (Chapter 4.5)

Software Agents

Software agent, as defined in [Bra97] is a
software entity which functions continuously and autonomously in a particular environment (context), often inhabited by other agents and processes.

The agents are further distinguished according to their basic characteristics and capabilities. These are the autonomy, cooperation and learning capabilities\(^3\) (Fig-3.5).

From the technological point of view, agents are constructed as objects, hence the object oriented approach is applied.

![Fig. 3-5 Typology based on Nwana’s primary attribute dimension](image)

**Code on Demand vs. Remote Evaluation**

Code on Demand approach is one step towards the code mobility. The system requires some code in order to be able to complete some task and this code is not available locally. The code is demanded from other system on a shared network, which provides it. The example of this approach can be the Java Applets or Servlets (chapter 3.2.5).

In the library, this approach can be useful for example to enable clients to view different kinds of document formats without having stored all kinds of viewers on their local systems that may be inefficient in the situation when the system supports wide variety of formats. In this case, the agent operates as a manager that is responsible for obtaining the missing code.

\(^3\) The Autonomy is a required attribute from the definition, however, in some cases the agent is characterized as semi-autonomous, which is the case of Collaborative Learning Agents showed in Fig-3.5
**Mobile Agents Technology**

The mobility of agents is another optional characteristic that basically means that the agent can change the context within its execution. The agent mobility has been introduced because of some technological limitations that stationary agents have. The main advantages that mobile agents have are:

4. network load reduction (fig-3.6)
5. network latency solution for real-time systems
6. protocols encapsulation
7. asynchronous and autonomous execution
8. dynamic adaption
9. work in heterogeneous environment
10. fault tolerance

**Fig. 3-6 Stationary agent – RPC vs. Mobile Agent – RP (Remote programming)**

**MASIF (MAF) standard**

The mobile agents technology has been implemented in several different systems. In order to enable the particular systems to interoperate, the standardization efforts have lead to an OMG proposal of MAF standard, later renamed as MASIF.
3.4 Chapter Summary

In this chapter the web technologies were categorized and the core elements of the web technologies were described. The web technology is being reformed by establishing the new standards. The XML standard is a new markup language based on SGML that is expected to solve the problems encountered with the original WWW language, the HTML.

The emphasis has been given to the web programming domain, describing the different approaches and solutions to creating . The server-side scripting paradigm is pointed out and all alternatives of it were presented. The original CGI scripting and the PERL language are emphasized as it is the traditional approach used most frequently. Other alternative technologies, proprietary web server extensions and servlets are also mentioned.

A special topic is the JAVA language which has been developed with the aim to support the needs of the Internet programming and other heterogenous and multi-platform systems. The JAVA can be run on any system that has installed the Java Virtual Machine, independently on the operating system located underneath. The JVM processes the pre-compiled JAVA byte-code.