

6. DESIGN

6.1 *Introduction*

6.1.1 Purpose

This chapter deals with the basic design of the system. It specifies some of the basic design approaches available for designing iBOS and compares different approaches so as to select the best possible design under the given constraints.

6.2 *Scope*

iBOS system aims at providing an operating system for the lower end machines so as to enable web browsing. The user of iBOS can view a given piece of Information available on the web. An iBOS user can only view a web page and cannot save it on the local disk.

Scope of the system has been considered under the following categories:

- System Objectives
- Major Requirements
- Design Constraints and Limitations

6.2.1 System Objectives

Functionality:

Main Aim : To design and implement an information kiosk dedicated for internet browsing only.

Major Capabilities :

- ♣ Parsing HTML 1.0. complaint web pages.
- ♣ Caching of web pages on local machine.
- ♣ Laying out HTML formatted text.
- ♣ Allowing form submissions.

Performance Objectives:

- Support on a low end machine.
- Functionally correct behavior.
- Disallowing saving of web pages on local disk.

Monetary Cost:

To keep the software free and open for development.

6.2.2 Major Requirements

Following is the compendium of the major requirements of an *iBOS* user:

- Internet browsing on a lower end machine.
- Support text only browsing.
- View web pages compliant with HTML 1.0.
- Support basic HTML tags.
- Support various browser controls such sending a request, scrolling, back, forward, navigating between various web page elements.
- Support various system controls such as shutting down the system or restarting the system

6.2.3 Design Constraints and Limitations

User-Interface

- *iBOS* supports only text based web browsing.
- At any instance only a single browser window is active.

Sessions

- *iBOS* does not implement HTTP sessions.

HTML support

- Only the most common HTML tags are supported.
- Frames are not supported.
- Following are the Major HTML tags supported by *iBOS* :

- ♣ <A>
- ♣
- ♣ <BODY>
- ♣

- ♣ <CENTER>
- ♣ <FORM>
- ♣ <HEAD>
- ♣ <HTML>
- ♣ <HR>
- ♣
- ♣ <INPUT>
- ♣
- ♣
- ♣ <OPTION>
- ♣ <SELECT>
- ♣ <TABLE>
- ♣ <TD>
- ♣ <TEXTAREA>
- ♣ <TITLE>

- ♣ <TR>
- ♣ <U>
- ♣

- Following are the specific features which are not supported by iBOS :
 - ♣ <table> tag inside <table> tag.
 - ♣ Column and row span with in a cell of table.
 - ♣ <table> tag cannot be enclosed within a <A> tag.
 - ♣ Vertical alignment with in a cell of a table.
 - ♣ Horizontal scrolling not supported.
 - ♣ <table> tag within <center> tag not supported.
 - ♣ Hyperlinks only for image and text.
 - ♣ Tags such as < >, < <, <blockquote>, <p> etc are not supported.
- iBOS does not support cookies.
- Security services such SSL etc.have not been implemented.
- iBOS does not support the functionality of History.
- iBOS does not provide the user with the facility of customizing the browser settings.
- iBOS does not support scripts ex. JavaScript, VBscript.

6.3 DESIGN ARCHITECTURE

6.3.1 Components of a web-browsing system

A web-browsing system is basically a hypertext content-based network information system. It comprises of the following major software components (in a typical Local area network environment):

- Software on a local host:
 - ♣ Operating system services
 - ♣ TCP/IP transport-network layer
 - ♣ Web-browser
 - HTTP Client
 - HTML Parser
 - Cache Manager
 - Layout engine
 - Rendering Engine
 - Event Handler
- Software in local network environment
 - ♣ Proxy servers
 - ♣ Domain Name Servers
- Software on the Internet
 - ♣ Web-servers

With respect to the above stated components, a model of the system can be described by identifying:

- i) The location of each component
- ii) The responsibilities of each component
- iii) The interaction between the components, and
- iv) The boundaries defining the systems and sub-systems.

Also, the semantics of each model, meaning the advantages, the disadvantages, the consequences and the effects of using each model must be described. Again, the requirements, in terms of hardware, must also be stated for each of the models.

6.3.2 Symbols and Notation

The following symbols are used in describing the various architectural models:


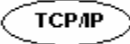
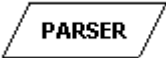
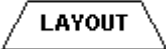

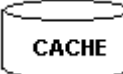
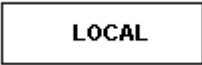
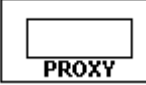




	HTTP Client
	TCP/IP Layer
	HTML Parser
	Layout Engine
	Rendering Engine
	Cache Manager
	Local Network Server
	Proxy Server
	Event Handler
	Operating System
	Domain Name Server
	Web-server

Figure 1 Notation Used

Arrows show the direction of communication. Empty rectangles represent system boundaries. If a component is depicted as being inside another component, then it means, that it is in fact a part of that system itself, or that it is located as a sub-system of that system.

6.3.3 Architectural Models

An architectural model is concerned with the placement of components in a system, and identifying boundaries and relationships amongst the components. The following architectural models represent different approaches to the design of the web-browsing system.

- 1) Dial-up model
- 2) LAN-Proxy server based model
- 3) Variations on the LAN-Proxy server model
- 4) Thin-client model

6.3.3.1 *Dial-up model*

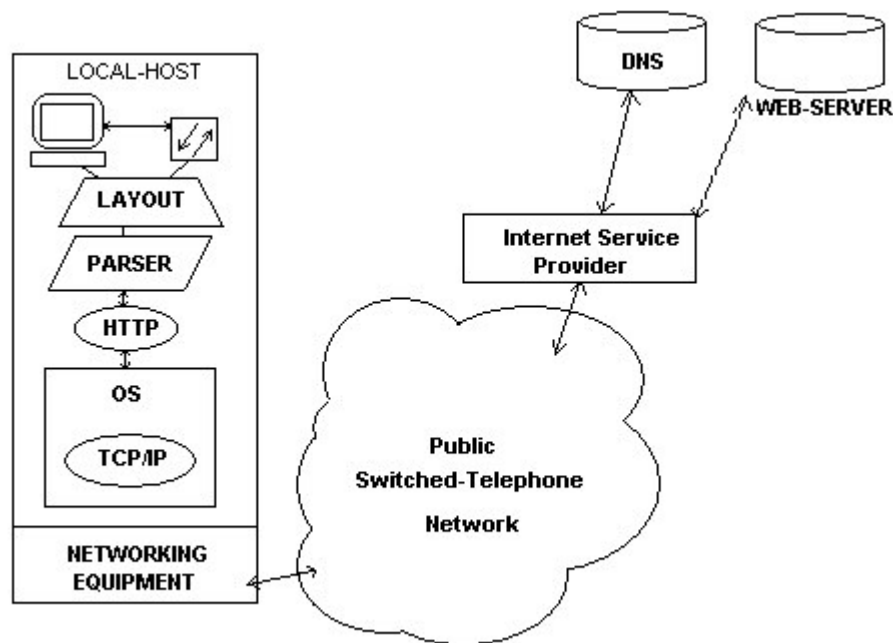


Figure 2 Dial-up model

Description:

In the dial-up model, each host is given a temporary IP address by the Internet service provider. The host computer uses networking equipment such as modems, etc. to communicate over the public switched-telephone network with the ISP. The ISP (Internet service provider) owns a gateway to the Internet. The ISP forwards name-resolution requests and other HTTP requests to their respective destinations. It also accepts the responses on behalf of the dialup host, and forwards them in a transparent fashion.

6.3.3.2 *LAN-Proxy server model*

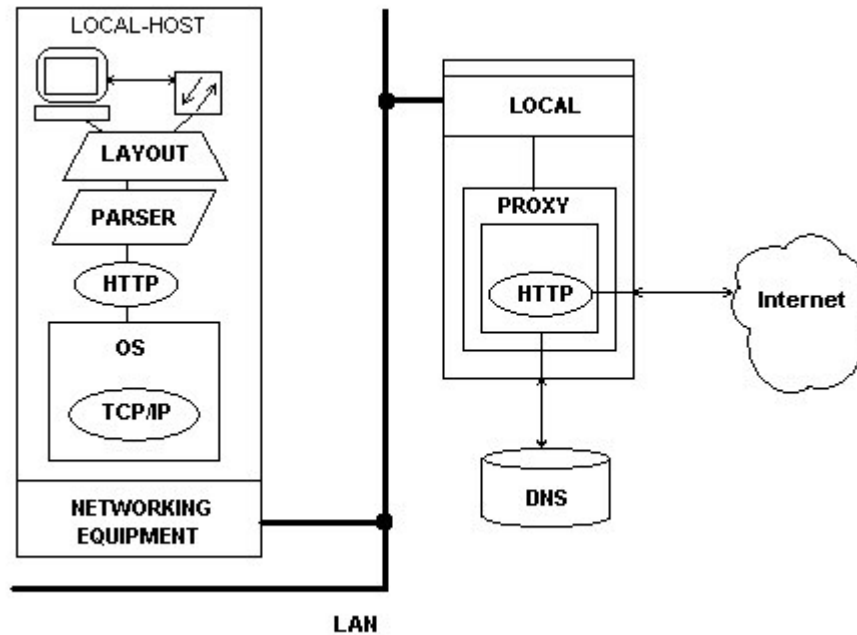


Figure 3 LAN-Proxy server model

Description:

The LAN-Proxy server model refers to an arrangement in which a server machine in a local-area network hosts a proxy server daemon that accepts requests from local hosts on the LAN and forwards it to the Internet. It also accepts the responses to the web-requests and returns the resources back to the hosts in the LAN.

6.3.3.3 *Variations on the Client-server model:*

The client server model can be varied in many different ways. Following are few common variations:

- i) **No HTTP client on the workstation**
In this model the operations of the Hypertext transfer protocol are delegated to the proxy server. All HTTP operations of a web-client are handled by the proxy server. This implies a greatly reduced burden for the client machine.
- ii) **Caching at the proxy-server**
In this variant of the client-server model, pages are cached at the local server site besides the workstation. This allows many workstations to share the same cache. There is some strength in the conjecture that some web-sites are accessed quite regularly and therefore a common cache for such sites is quite useful.

- iii) Filtering at the proxy-server
 Since many text-browsers are not able to view images, there is really no point in transmitting large-sized raster data to the workstations. Images can be stripped from the HTML pages being transmitted to the workstations. Other such content which is not cognizable at the receiving end needs to be excluded from data exchanged between the proxy server and the workstations.
- iv) No name-resolution queries from the workstation
 The task of resolving domain and host names can be delegated to the proxy server machine. This would reduce the complexity of the network interaction between the proxy and the network workstations. The idea is that the proxy server has to request the resources from the remote hosts anyway. So if the task of resolving host names into IP addresses is taken up at the proxy server, the workstation does not need to transmit a DNS query, accept the response to it, and then send a new HTTP request.

6.3.3.4 Thin-Client Computers

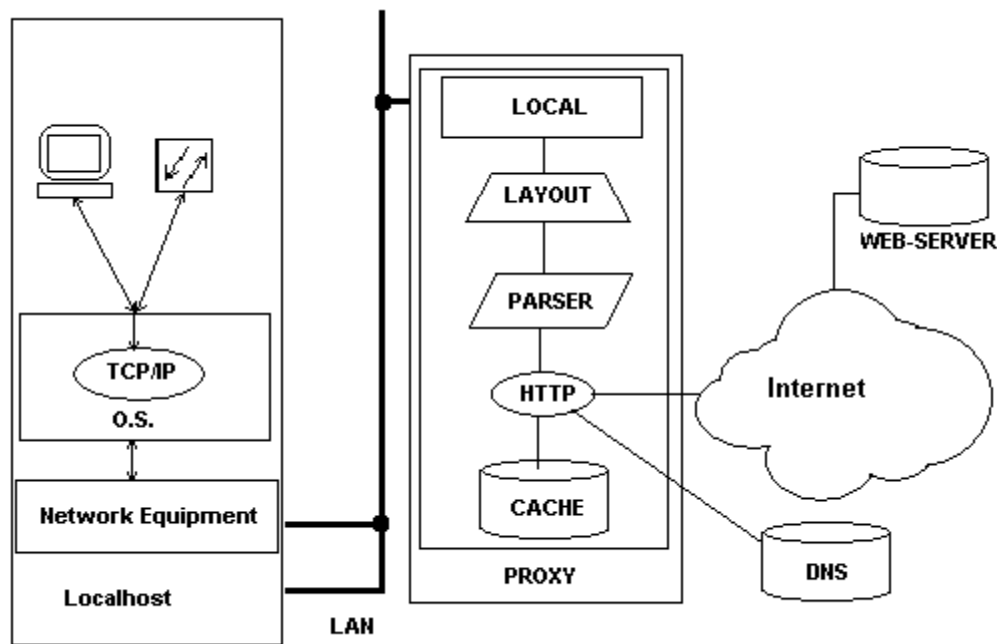


Figure 4 Thin-Client Computers

Description:

A networked computer workstation is one which downloads all software and data that it requires, and then executes it. A thin client has a set of bare minimum programs that handle the devices (display and rendering, and input devices) while all processing is done at some remote machine (in this case, the proxy server). In this model, all processing of

HTML is done at the proxy server and laid out frames of text/images are transferred over the network to the browsing end. Only the user-events, mainly inputs such as mouse-clicks etc. need to be handled locally, because they are required to produce a quick feedback. All other processing is delegated to the proxy server machine.

CHAPTER 7

Contents

7	Feasibility of Design Options	
7.1	Objective	...119
7.2	Focus	...119
7.3	Purpose	...119
7.4	Major design constraints and limitations	...119
7.5	Comparison of Design Options	...121
7.5.1	Dialup Model	
7.5.2	LAN Proxy Server Model	
7.5.3	Thin Client Computers	
7.6	Conclusion	...122

7. FEASIBILITY STUDY

7.1 *Objective:*

To design and implement an operating system that enables Internet access by providing the world-wide web services on low-end machines..

7.2 *Focus:*

The project aims to focus at the following main issues:

1. Enabling the web
2. Workability on low-end target systems
3. Light-weight, in not demanding high hardware/software requirements.

7.3 *Purpose*

The purpose of this document is to compare and contrast various design option available for iBOS. The document gives a detailed analysis of each option, evaluates pros and cons of each and accordingly suggests the best possible design under given constraints and limitations.

7.4 *Major Design Constraints and Limitations*

User-Interface

- iBOS supports only text based web browsing.
- At any instance only a single browser window is active.

Sessions

- iBOS does not implement HTTP sessions.

HTML support

- Only the most common HTML tags are supported.
- Frames are not supported. Following are the Major HTML tags supported by iBOS :
 - ♣ <A>
 - ♣
 - ♣ <BODY>
 - ♣

 - ♣ <CENTER>
 - ♣ <FORM>
 - ♣ <HEAD>
 - ♣ <HTML>
 - ♣ <HR>
 - ♣
 - ♣ <INPUT>

- ♣
- ♣
- ♣ <OPTION>
- ♣ <SELECT>
- ♣ <TABLE>
- ♣ <TD>
- ♣ <TEXTAREA>
- ♣ <TITLE>
- ♣ <TR>
- ♣ <U>
- ♣

- Following are the specific features which are not supported by *iBOS* :
 - ♣ <table> tag inside <table> tag.
 - ♣ Column and row span with in a cell of table.
 - ♣ <table> tag cannot be enclosed within a <A> tag.
 - ♣ Vertical alignment with in a cell of a table.
 - ♣ Horizontal scrolling not supported.
 - ♣ <table> tag within <center> tag not supported.
 - ♣ Hyperlinks only for image and text.
 - ♣ Tags such as <>,<<,<blockquote>,<p> etc are not supported.
- *iBOS* does not support cookies.
- Security services such SSL etc. have not been implemented.
- *iBOS* does not support the functionality of History.
- *iBOS* does not provide the user with the facility of customizing the browser settings.
- *iBOS* does not support scripts ex. JavaScript, VBscript.

7.5 Comparison

The following section evaluates the pros and cons of various possible architectures of iBOS :

7.5.1 Dial-up model

Advantages:

The advantages of the system include:

- 1.) Low-cost setup in terms of the initial investment in hardware and software.
- 2.) Resources are not shared, and thus are isolated and hence secure.

Disadvantages:

The disadvantages of the dial-up approach to browsing the world-wide web are many:

- 1.) Hidden and recurring costs in terms of the telephone calls made.
- 2.) Slow speed of access.
- 3.) Since all browser components are located at the host, the host is overloaded with responsibility.

7.5.2 LAN-Proxy server model

Advantages:

The advantages associated with a LAN and proxy-server setup are manifold:

- 1.) Sharing of a single web-connection for multiple workstations.
- 2.) Greater speed than is achievable in a direct dialup connection.
- 3.) The machine hosting the proxy server daemon can be hardened to act as a bastion host.
- 4.) All network traffic may be monitored at the proxy server site, although this sometimes results in greater delays.
- 5.) Caching can be performed at the proxy server causing the delays to be reduced.

Disadvantages:

The disadvantages associated with a LAN and proxy-server arrangement includes:

- 1.) The client side machines still have to handle all the HTTP operations.
- 2.) The setup requires extra hardware including network interface modules, and cables.
- 3.) There is a single point of failure for the browsing system. However, for an Internet browsing system, which is mostly stateless (except for some applications that require sessions), failure is not a very important issue.

7.5.3 Thin-Client Computers

Advantages:

The advantages in this model are few:

- 1.) The workstations are all saved the burden of laying down the rules for representing information which is transferred as is to be displayed.
- 2.) No caching is required at the workstation and it is all handled at the proxy.

Disadvantages:

The disadvantages of this model outweigh the advantages:

- 1.) High-end machines are required for the role of the proxy server machines.
- 2.) Network traffic increases directly as the number of requests on the proxy server.
- 3.) Since the server must process requests and handle the HTML parsing etc. for each workstation, it is always under tremendous loads.

7.6 Conclusion

Based on above analysis the following architecture has been chosen for *iBOS*:

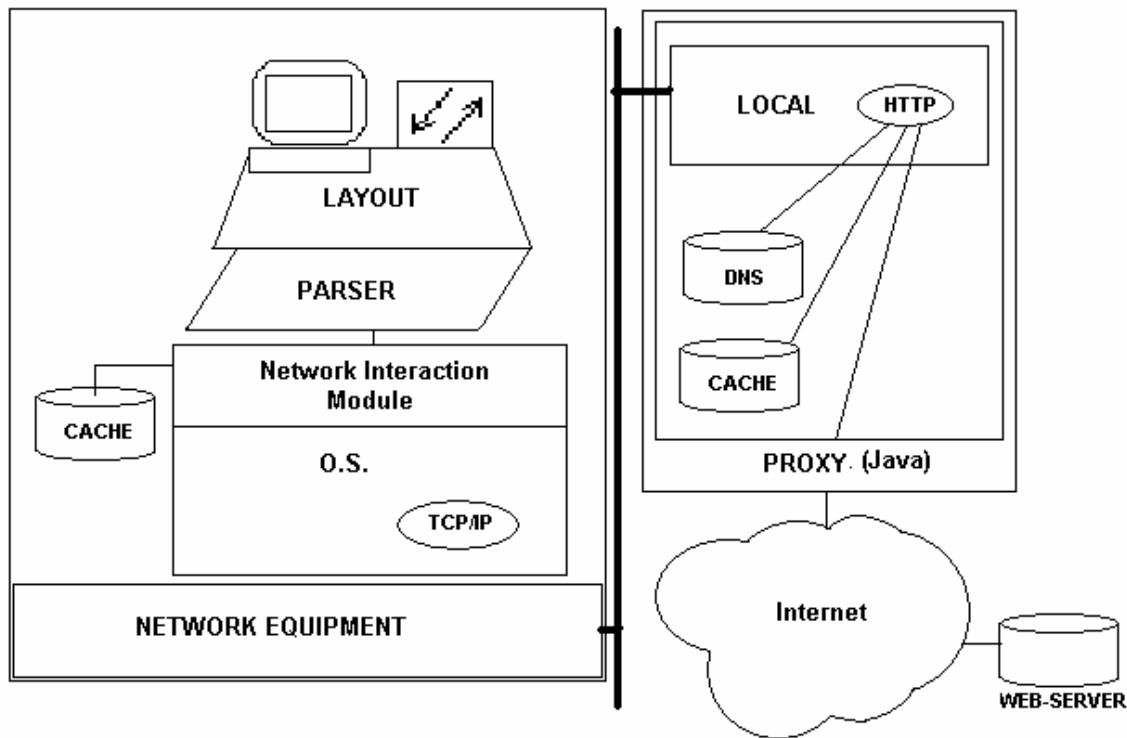


Figure 5 *iBOS* Architecture

Description:

The *iBOS* web-browsing system is based on a variant of the client-server model in which:

- 1.) All HTTP-operations are delegated to the proxy server.
- 2.) All HTML-related operations are assigned to the workstation.
- 3.) Caching is done both at the proxy and the workstation.
- 4.) The parser, the layout engine and the rendering engine work in close coordination, and even overlap.
- 5.) The event handling module is a part of the parsing and rendering modules.

- 6.) The proxy server is a collection of threads that perform various functions including name-resolution, server-side caching, forwarding client-messages as HTTP-requests and responses, etc.
- 7.) Filtering of HTML content takes place at the proxy server site.
- 8.) The client software works as a shell for the operating system.

Advantages:

- 1.) iBOS inherits all the advantages of the LAN-proxy server model and its variants.
- 2.) The system requires no further software to be established on the workstations.
- 3.) iBOS is modeled as a set of objects which makes it easily extensible.

Disadvantages:

- 1.) The primary disadvantage of this model is the inability of the workstations to directly establish sessions with remote machines.
- 2.) The proxy server is needed to perform a lot of functions other than just the plain forwarding of requests and responses.

CHAPTER 8

Contents

8 Detailed Design	
8.1 Introduction	...125
8.1.1 Purpose	...125
8.1.2 Scope	...125
8.1.3 Major Design constraints	...125
8.2 Overview of Detailed Design	...125
8.3 Structural Modeling	...129
8.3.1 Class Diagrams	...130
8.3.1.1 Browser Classes	
8.3.1.1 Network Classes	...131
8.3.1.2 Parser Classes	...137
8.3.1.3 Event Handler	...210
8.3.1.4 Caching Classes	...213
8.3.1.2 Proxy Server Classes	...217
8.3.2 Object Diagrams	
8.3.2.1 Overview	..227
8.3.2.2 Notation Extension	..227
8.3.3 Component Diagrams	...254
8.4 Behavioral Modeling	...266
8.4.1 Sequence Diagrams	...267