Toward the Next Generation Public Traffic Information System Using Internet*

Seokhoon Lee†, Jongill Ahn†, Sungyong Lee†, Tae-Choong Chung†, Hyonwoo Seung††

† Department of Computer Engineering, Kyung Hee University, Korea
†† Department of Computer Science, Seoul Women's University, Korea
{shlee@oslab.kyunghee.ac.kr, slee@nms.kyunghee.ac.kr}

Abstract

This paper introduces a next-generation public traffic information system, which facilitates the WWW(World-Wide Web) to provide users with easier access and use. The proposed system is composed of three subsystems: client/server interface, knowledge based path search system and traffic data storage system. The user interface utilizes Java language to furnish users with multimedia data accessibility and interactiveness. The path search system produces optimal solutions based on dynamic traffic data, while previous search systems are limited to use static traffic data such as bus/subway route information. The storage system is designed to give the search system more efficient access to the traffic information. The consisting sub-systems are interconnected on the WWW using CGI(Common Gateway Interface). A client requests the server to search a path. The server asks the search system through CGI to get a result from the database, and returns it to the client. The system can be extended to an integrated navigation system which includes a variety of information on the Internet as well as the traffic information.

1. Introduction

The notorious traffic congestion in the metropolitan area can be relieved by directing people to the public transportations. One of the many ways to achieve the goal is to provide people with an easy-to-access and user-friendly public traffic information system.

However, most of the current traffic information systems are limited to show bus/subway routes to people standing at stations. Some advanced systems have been proposed with search engines based on the static traffic data such as the distance between two stations to show optimal paths. But, much remains to be desired to satisfy the various users' needs. The system should be accessed more easily by users and have more friendly user interface. Even people who are not familiar with computer environments should be able to use the system at any convenient places such as bus/subway stations, public institutions and home. It should also be equipped with more efficient search engine to produce optimal paths within a reasonable time based on the dynamic traffic information such as current traffic situation or weather condition.

This paper proposes a public traffic information system where users can access the system easier than ever before. It utilizes the WWW, CGI and the Java programming language to provide the 'state-of-the-art' user interfacing techniques, and AI-based path search algorithms to produce results within "real-time" and in a more efficient way. On the Web, through a graphical browser such as Netscape or MS-explorer, users can activate a hyperlink, read multimedia traffic informations, or download a file within a single mouse click. In the path search system, a modified A^* algorithm with a conditional pruning technique is used to consider various search conditions such as the number of routes to search or transfers, instead of using the dynamic programming technique which is generally used to search a shortest path.

The remainder of this paper is organized as follows. In Section 2, we review the client/server interface on the Web. Section 3 discusses the AI-based optimal path search engine. The characteristics of the integrated system are described in Section 4. Finally, conclusions appear in Section 5.

2. Client/Server Interface on World Wide Web

The WWW(sometimes called just 'the Web') is a collection of information stored on computers over the world that are connected to the Internet. The Web is the largest, most comprehensive, and most widely used electronic information system in the world. It is popular. It is easy to use; you move to new topic by

* This paper was partially supported by Information Super Highway Application Project under grant No. 96-147, ETRI, 1996.

pressing a key or clicking a mouse button.

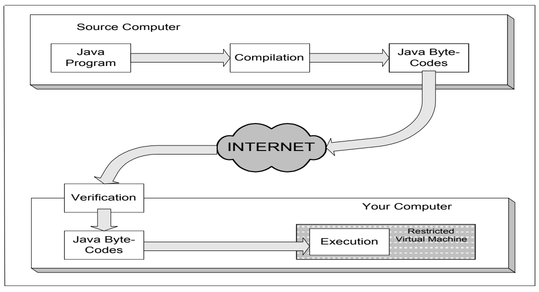
Since the proposed public traffic information system is mounted on the Web through the Internet, it can be used on PCs, NCs(Network Computers) or Internet TVs, while the previous traffic guidance systems use exclusive terminals furnished at bus/subway stations or public institutions.

The system is built on a client-server model. The client and the server communicate with each other using a common protocol, HTTP(Hyper-Text Transfer Protocol) over the TCP/IP. The client requests a traffic information, and the server answers. The client has a Web browser such as Netscape or MS-Explorer as its user interface. The browser provides the graphical user interface(GUI) that enables users to see Web pages and to access other pages with the click of a mouse button. A Web page can have any multimedia traffic information such as formatted or unformatted text, images, sounds and videos included in it. The browser loads such pages and displays them. The traffic information contained on the pages is formatted in HTML(Hyper-Text Markup Language) and Java Applet/Script. In order to provide various multimedia traffic information, many Plug-In programs are used: Real Audio and Toolvox for sound data and Quick Time and Stream Works for graphics. The client(Web browser) interfaces with the path search system and the traffic data storage system through CGI.

2.1 User Interface on Client

Most of the information available on the Web today is formatted in HTML(HyperText Markup Language), which is a set of symbols that specify the structure of a document. However, in a public traffic information system where interactiveness is very important, it is not adequate to use the hypermedia system since HTML itself is static in a sense that you access one page, then click on a link, and another page appears. One solution is to use Java. With Java, data are no longer restricted to a page-by-page display. So, we can have updated traffic information appearing "live" on the browser window. Java, developed by Sun Microsystems, is built as an extension to HTML and can be included in WWW pages. Rather than execution on the HTTP server, a Java application actually is downloaded and executed by the Web browser. When a Java program(called an Applet) is accessed over the Internet, the entire application is downloaded to the browser. The browser then executes the code. In order to do this, the browser must include a Java interpreter. The Java interpreter and the interpreter built into a Web browser act as a virtual machine to run the Java code [see Figure 1]. This means that the language is fast, like a compiled language, but also is platform-independent[2][15][17].

The public traffic information system proposed in this paper uses AWT(Abstract Window Toolkit) classes[7], Windows-version of Java to provide menus, buttons, dialogboxes, to represent traffic maps, to select the source/destination, and to produce the output of search results. Plug-In programs are also used to show traffic conditions in videos at the spots users want to see, or current traffic/weather informations in sound. A

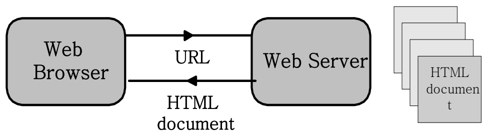


[Figure 1: Execution of Java program]

Plug-In is an API(Application Programming Interface) added to the Web browser to process various kinds of multimedia data. Plug-Ins used in the system are Real-audio and Toolvox for sound and Stream Works and Quicktime for motion pictures.

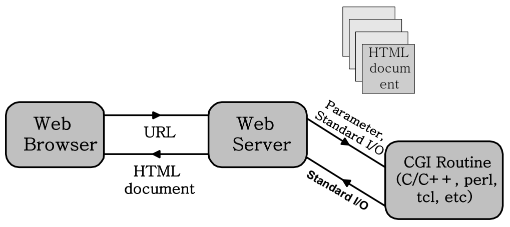
2.2 System Interface on Server

To access HTML documents, Web browsers have to know where to look. They get this information from a Uniform Resource Locator(URL). The URL contains the pieces of information a Web browser requires to locate a page on the Web.



[Figure 2: Data transport between a Web browser and a Server]

However, this mechanism simply allows users to use hyperlinked HTML documents. It does not allow them to use existing databases or dynamically created data. The Common Gateway Interface(CGI) is one way to overcome this limitation. CGI is a mechanism that allows Web clients to execute programs or access databases on a Web server and to receive their output[3][4][9][13]. [Figure3] shows how the mechanism works.



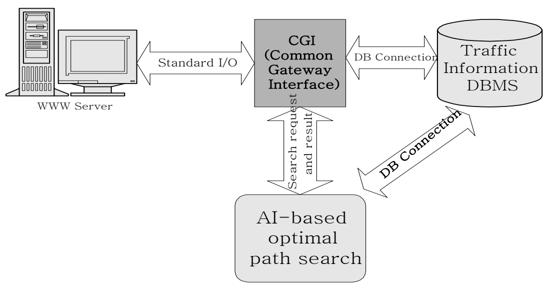
[Figure 3: Data transport between a Web browser, and a Server using CGI]

The CGI programming is used in the proposed public traffic information system in order to interconnect the client(Web browser) to the server where the knowledge based path search system and traffic data storage system reside. Following is the normal process how CGI works in the system:

1) The user calls a CGI program to issue a search request on the Web browser

2) The Web browser contacts the Web server asking for permission to run the CGI program
3) The Web server checks if the requester is allowed access to the CGI program
4) The CGI program executed
5) The resulting paths produced by the CGI program are returned in the HTML format to the Web browser
6) The Web browser displays the CGI output

[Figure 4] illustrates the CGI's role on the server.


WWW Server
Standard I/O
CGI (Common Gateway Interface)
DB Connection
Traffic Information DBMS
Search request and result
DB Connection
AI-based optimal path search

[Figure 4: CGI role on the Server]

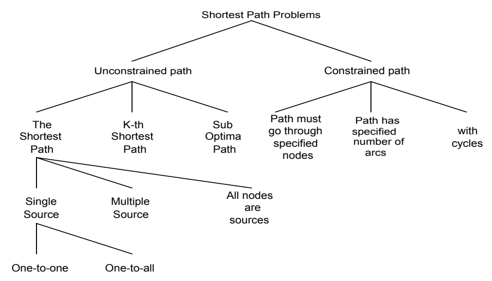
3. Path Search System

The path problem has been a hot research topic for decades not only in academic fields but in the real life applications. Many algorithms have been developed to find optimal solutions. Among them, A^* algorithm has been received more attention than any other algorithms. Although A^* algorithm proved to be successful, it quickly runs out of space even for problem instances of moderate size when searching for optimal solutions since it requires exponential space. Furthermore, it produces the best solution only. It does not produce the alternate solutions sometimes needed in many applications.

To overcome such problems, we propose a modified A^* algorithm which produces more than one optimal solutions and utilizes a conditional pruning technique for the time efficiency[1].

The objective in the system is to produce more than one optimal paths from source to destination in the cost value order. Path problems can be classified into four categories [8]:

1) single source/single destination shortest paths
2) all pairs shortest paths
3) K-th shortest paths(first, second, ...)



[Figure 5: Classification of the optimal path search algorithms]

4) shortest paths going through specified nodes

[Figure 5] shows the classification tree. The kind of algorithm applied in the proposed system falls under the single source/single destination K-th shortest path search algorithm.

Since (vehicle) transfers must be considered in the proposed system, a heuristic search technique, A^* algorithm, has been modified and applied which has been proven time-efficient in certain problem domains[10][11][12][16].

3.1 Modified A* Algorithm

While A^* algorithm finds the least $g(n) + h^*(n)$ value first and uses dynamic programming technique to search a shortest path, the modified A^* algorithm proposed in this paper uses a conditional pruning technique to consider various search conditions such as the number of routes to search or transfers[14]. The differences are illustrated in [Figure 6].

A* algorithm	Best First search + under estimation + dynamic programming
modified A* algorithm	Best First search + proper estimation + conditional pruning + management of changing vehicles

[Figure 6: Comparison between the A* and the modified A* algorithm]

3.2 Heuristic Functions

Two heuristic functions are considered when time or distance is used as search criteria. When the search criteria is distance, the Euclidean Distance(UD) is used. In case of time, UD*TD*R is used as the heuristic function, where TD stands for the average time/distance calculated from the database scoring traffic cost values and R is the weight value between 0 and 1. In addition, transfer cost must be considered as well. Those search criteria and transfer cost can be given by users. [Figure 7] summarizes both heuristic functions[6].

	Time	Distance
g(n)	∑ ime(min) + transfer cost *number of transfers	∑ istance(km) + transfer cost *number of transfers
h*	UD * TD * R	UD

(UD : Euclidian Distance
TD : average time cost/distance in DB
R : real number between (0 ~ 1))

[Figure 7: Differences between the time cost and the distance cost]

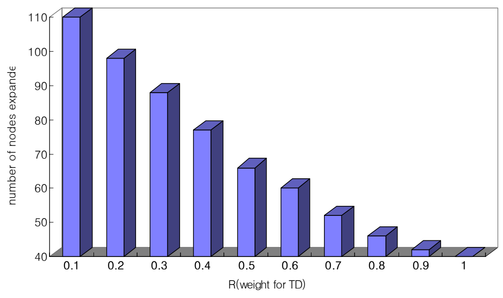
As shown in [Figure 8], when time is used as cost value, g(n) is calculated using the formula : ∑ + (cost(T)*num(T)), where ∑ is summation of time cost of each edge from the start node to the n-th node (min.), cost(T) is time cost of each transfer (min.), num(T) is number of transfers. When cost value is distance, $\sum$ is used instead of $\sum$. In that case, $\sum$ D is summation of distance cost of each edge from the start node to the n-th node (km).

3.3 Simulation of the Heuristic Function

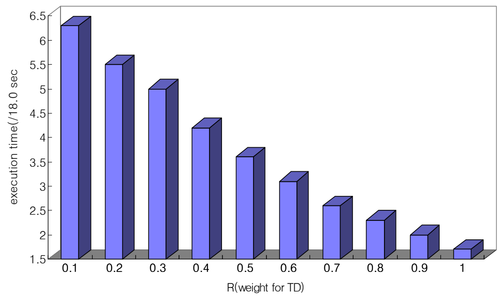
We examine number of nodes expanded, execution time, and relative cost error for values of R(weight for TD) in case time is selected as the search criteria. For the simulation, we created a 300-node graph based on the data collected from real bus/subway routes. The cost between two adjacent nodes n and m is calculated using the following formula:

cost(n,m) = h^*(n,m) * TD * R, where h^*(n,m) is the Euclidean Distance between node n and m. The values of R are real numbers from 0 to 1. For a given value of R, we chose a pair of source/destination at random and executed 200 times. Then we averaged the number of nodes expanded, the execution time, and the relative cost error. The results are shown in [Figure 8, 9 and 10]. The relative cost error, E, was calculated as follows:

$$E = \frac{cost\_of\ searched\ path - cost\ of\ optimal\ path}{cost\ of\ optimal\ path} \times 100(\%)$$

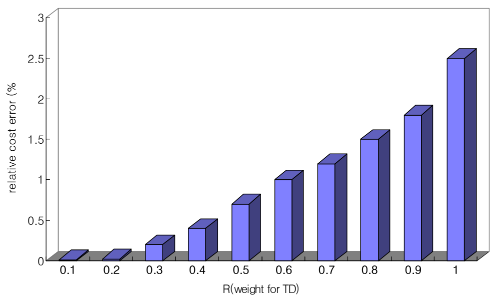


[Figure 8: Average number of nodes expanded]



[Figure 9: Average execution time]

[Figure 8] illustrates a remarkable reduction in the average number of nodes expanded as R comes close to 1. Accordingly, the execution time, proportional to the number of nodes expanded, also reduces as shown in [Figure 9]. [Figure 10] shows the average relative cost error is under 3%.



[Figure 10: Average relative cost error]

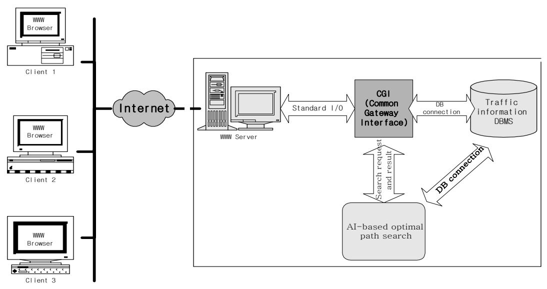
4. The System

4.1 Goals and the whole Picture

The goals we try to achieve in the system are two-fold:

1) to provide users with optimal paths from the places where they are to destinations based on "live" traffic informations through home PCs, NCs and exclusive terminals furnished at bus/subway stations or public institutions.

2) to allow users to access various Web services such as public informations from government institutions or private companies, and on-line shopping, etc.

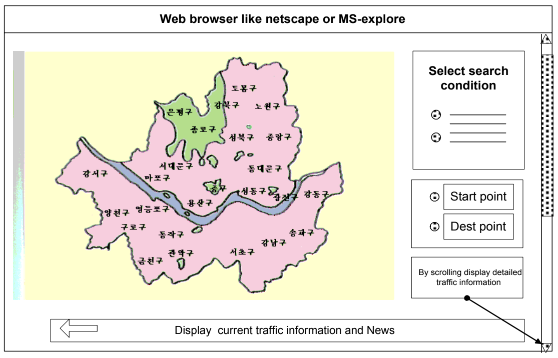


[Figure11: System architecture]

[Figure 11] illustrates the whole system. Clients are connected to the server through Internet platform-independently. Any PCs, NCs or exclusive terminals where Web browsers can be operated on are possible clients. Java and Plug-Ins are used to build the user interface on clients. The server is coupled with the AI-based path search system and the traffic databases through CGI.

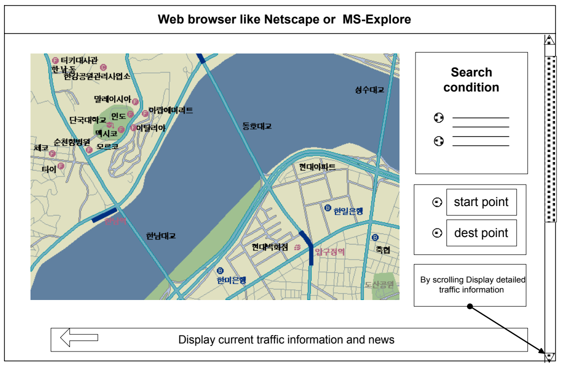
4.2 Using the System

The user from a client sees the first screen [Figure 12] upon connection to the server using a Web browser such as Netscape or MS-Explorer.


Web browser like netscape or MS-explore
Select search condition
Start point
Dest point
By scrolling display detailed traffic information
Display current traffic information and News

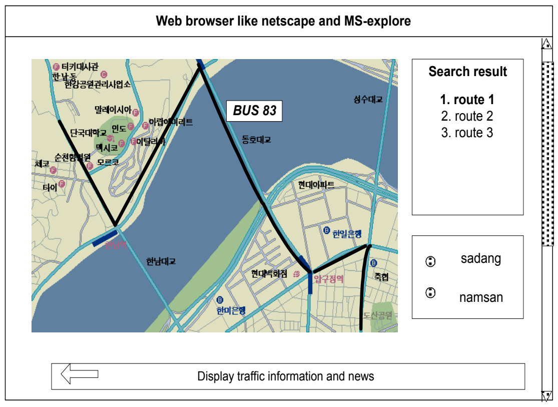
[Figure 12: First screen upon connection]

The picture shows the metropolitan area, Seoul, as a whole. At the bottom of the screen, current traffic spot news such as car accidents, jammed areas, roads under construction or weather informations can be displayed like an electric sign. The user can move down to detailed pictures(maps) to select the source/destination he/she wants. A source/destination is selected by simply clicking a mouse on a spot. While moving around the map, several places are displayed around the mouse pointer to help the user to pinpoint the place he/she wants.


Web browser like Netscape or MS-Explore
Search condition
start point
dest point
By scrolling Display detailed traffic information
Display current traffic information and news

[Figure 13: A detailed map]

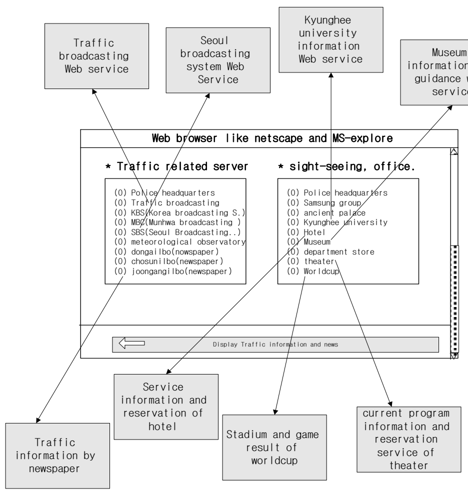
[Figure 13] shows a detailed map. On the detailed map, the user selects the source and destination after he/she sets up search conditions such as shortest distance or time, least cost, number of transfers. The information the user creates is transmitted through URL to the server.


Web browser like netscape and MS-explore
BUS 83
Search result
1. route 1
2. route 2
3. route 3
sadang
namsan
Display traffic information and news

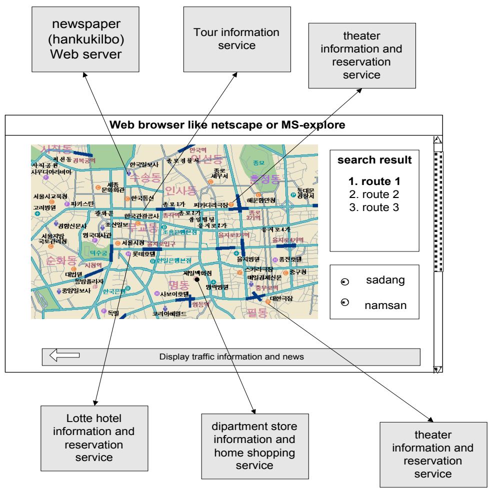
[Figure14: Search results]

Using the information, together with the geographical data stored in the database, the server finds the best path(s) and send the result back to the client. The client displays the result in text and graphics as in [Figure 14]. The bold line shows an optimal path from source to destination under certain conditions.

Not only the traffic information provided by the system itself, but traffic related services provided by the other institutions such as the Traffic Broadcasting Station, newspaper companies or Seoul Metropolitan Police Headquarters can also be accessed using the system [Figure 15].


Traffic broadcasting Web service
Seoul broadcasting system Web Service
Kyunghee university information Web service
Museum information and guidance web service
Web browser like netscape and MS-explore
* Traffic related server
(0) Police headquarters
(0) Traffic broadcasting
(0) KBS(Korea broadcasting S.)
(0) MBC(Munhwa broadcasting)
(0) SBS(Seoul Broadcasting..)
(0) meteorological observatory
(0) dongailbo(nowspaper)
(0) chosunilbo(newspaper)
(0) joongangilbo(newspaper)
* sight-seeing, office.
(0) Police headquarters
(0) Samsung group
(0) ancient palace
(0) Kyunghee university
(0) Hotel
(0) Museum
(0) department store
(0) theater
(0) Worldcup
Display Traffic information and news
Service information and reservation of hotel
Traffic information by newspaper
Stadium and game result of worldcup
current program information and reservation service of theater

[Figure 15: Connecting to the other traffic related sites]


newspaper (hankukilbo) Web server
Tour information service
theater information and reservation service
Web browser like netscape or MS-explore
search result
1. route 1
2. route 2
3. route 3
sadang
namsan
Display traffic information and news
Lotte hotel information and reservation service
dipartment store information and home shopping service
theater information and reservation service

[Figure 16: The system as multi-purpose information provider]

The public traffic information system mainly serves as a traffic information provider, but it can also be a multi-purpose information provider if connected through its map navigating functions to the other Internet services, such as home banking/shopping, hotel/theater reservations, travel services or civil affairs services.

As shown in [Figure 16], users can get informations about the places on the map in a natural way. Let's say a user wants to go a department store. He/she searches paths to the store, and by simply selecting the spot, looks into the information the store may provide, for example, the kinds and prices of the merchandise, opening/closing time, whether they are having a sale or not. Such an integrated navigation system is a possible extension of the system.

5. Conclusion

A public traffic information system is proposed to overcome traffic problems in the metropolitan area. The system aims to provide users with live traffic informations in a user-friendly manner through PCs, NCs, Internet TVs and exclusive terminals on the Internet and WWW. To achieve the goal, an easy-to-use user interface has been introduced utilizing Java, CGI and Plug-Ins. An AI-based path search technique has also been introduced which produces optimal paths within a reasonable time under dynamic traffic situations. The proposed system can be extended to an integrated navigation system where a variety of useful informations on the Internet as well as the traffic information can be accessed.

6. References

[1] A. L. Karl, H. Kaindl, "Bidirectional Best-First Search with Bounded Error: Summary of Results", IJCAI, 1993, pp. 217-223.

[2] A. V. Hof, S.Shaio, O. Starbuck, Hooked On Java, SUN Microsystems, 1996

[3] B. F. Rasmussen, "A Web to Database Interface", Sybase WWW home page, 1995.

[4] Eric Hermann, CGI Programming with PERL in a week, SAMS NET, 1996

[5] IBM, "DB2 World Wide Web Connection", IBM, 1995

[6] J.B.H Kwa, BS*: An Admissible Bidirectional Staged Heuristic Search Algorihm, Artificial Intelligence 38(2), 1989, pp. 95-109.

[7] Lemay, Perkins, Teach yourself JAVA in 21 days, SAMS NET, 1996

[8] Judea Pearl, "Heuristics: Intelligent Search Strategy for Computer Solving", Addison-Wesley Publishing Company, 1984, pp. 64-99.

[9] ORACLE, "Oracle and Internet", ORACLE White Paper, 1996

[10] Hart, Nilson and Raphael, "A Formal Basis For The Heuristic Determination of Minimum Cost Paths", IEEE Transaction on SSC-4(2), pp. 100-107.

[11] J.B.H Kwa, "BS* : An Admissible Bidirectional Staged Heuristic Search Algorithm", Artificial Intelligence 38(2), 1989, pp. 95-109.

[12] A.L. Karl, H. Kaindle,"Bidirectional Best-First Search with Bounded Error : Summary of Results", IJCAI, 1993, pp. 217-223.

[13] Shishir Gundavaram, CGI Programming on the World Wide Web, O'REILLY & ASSOCIATES, 1996

[14] I.Pohl, "First Result on the Effect of Error in Heuristic Search", In B. Meltzer and D. Michile, editors, Machine Intelligence 5, 1970, pp. 219-236.

[15] T. Ritchey, Java, New Rider, 1995

[16] H. Kim, Dvelopping traffic route information system usimg heuristic search and management system, Korea Research Foundation Reprot, 1993

[17] M. Shin, H. Choi, K. Park, “Internet esperanto Java”, Microsoftware , 1995, pp. 262-297.