

J. Natural Sci., Vol. 6, 1999

Sangmyung University

## Prolog

,

,

7

1999 12

# Prolog

,  
,  
Prolog 가  
. Prolog  
Prolog  
. Prolog 가 가  
1.  
가 가  
, ,  
, 가  
. Prolog[2] 가 가  
Prolog [6] Prolog 가  
. Prolog  
[1][3][4][5][9][13]가 Prolog Prolog  
Prolog Prolog  
2 Prolog  
, 3 , 4  
. 5 Prolog  
가 [7][14][15]  
. 6

2.

[14] Prolog [1][3][4][5][9][13]

Prolog

4 가 , CGI[8] , HTTP

[12], Prolog Prolog Prolog

2.1

Prolog Prolog Prolog

가

Prolog Prolog

Prolog Prolog

5 가 (type)

가 , , ' '

HTML[11]

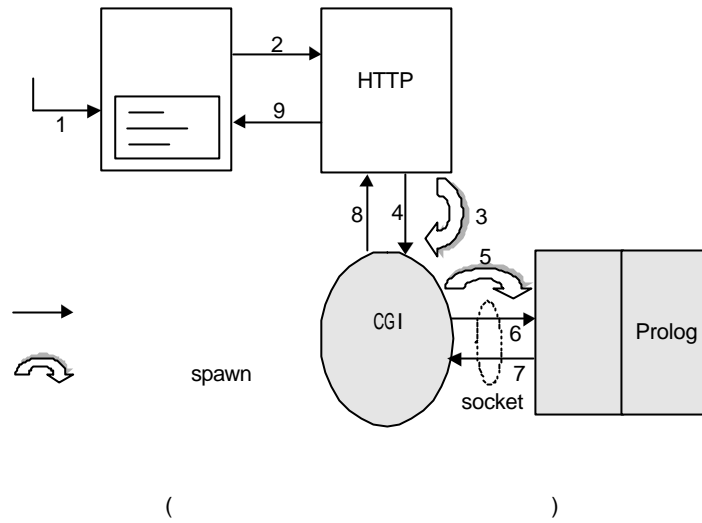
가

2.2.

2.2 , HTTP , CGI ,

Prolog 4 가 Prolog

2.2.



2.2.

1. 가 Prolog CGI  
 URL Prolog . ( :  
<http://poplar.sangmyung.ac.kr/~jskwon/ods-prolog-web/ods.cgi>
2. Prolog Prolog  
 가 URL Prolog  
 HTTP  
 Prolog 가 .
3. HTTP CGI spawn  
 CGI . HTTP Prolog  
 Apache HTTP  
 Netscape HTTP HTTP .
4. HTTP CGI  
 가 가 HTTP CGI .  
 GET POST 가 가 .
5. CGI Prolog spawn .
6. CGI Prolog HTTP  
 Pipes, FIFOs, Semaphores, Shared Memory, Socket  
 [10] .
7. Prolog Prolog CGI  
 5 가 . CGI

Prolog

8. CGI Prolog HTML

HTTP

9. HTTP 가 HTML

5 가

3.

2. CGI Prolog CGI

HTML CGI

Prolog

3.1.

Prolog 가 socket

[10] CGI[8]

CGI Prolog

[14]

- Type 0: 가 가
- Type 1:
- Type 2:
- Type 3: ‘ ’ ‘ ’
- Type 4:

3.2. CGI

CGI Prolog HTTP

socket HTTP

- Type 0: HTML CGI
- Type 1: HTML (control type) pull down menu
- Type 2: HTML text input
- Type 3: HTML pull down menu +100 -100
- Type 4: HTML radio button (recursively) HTML

4.

CGI C Prolog ECLiPSe Prolog  
Sun Ultra 5(Solaris 2.7)

4.1.

CGI socket  
socket\_begin/0, socket\_end/0  
socket ask\_null/1, ask\_integer/6, ask\_symbol/4, ask\_yesno/4,  
ask\_list/5

```
socket_begin :-
    socket(unix, stream, sd),
    bind(sd, 'ods.socket'), listen(sd, 1),
    accept(sd, _, socket), read(socket, _).
```

```
socket_end :-
    write(socket, _), close(socket, _).
```

```

ask_null(begin) :-
    write(socket, 0), !.
ask_null(end) :-
    close(socket), accept(sd, _, socket),
    read(socket, _), !.
ask_null(Message) :-
    write(socket, Message).

ask_integer(_, V, C, Question, I1, I2) :-
    write(socket, 1), write(socket, Question),
    write(socket, I1), write(socket, I2),
    close(socket), accept(sd, _, socket),
    read(socket, V), read(socket, C).

ask_symbol(_, V, C, Question) :-
    write(socket, 2), write(socket, Question),
    close(socket), accept(sd, _, socket),
    read(socket, V), read(socket, C).

ask_yesno(_, V, C, Question) :-
    write(socket, 3), write(socket, Question),
    write(socket, V), close(socket),
    accept(sd, _, socket), read(socket, C).

ask_list(_, N, C, Question, [H|T]) :-
    write(socket, 4), write(socket, Question),
    length([H|T], Length), write(socket, Length),
    ask_list_write([H|T]), close(socket),
    accept(sd, _, socket), read(socket, Choice),
    read(socket, C).

```

## 4.2 CGI

CGI	C	[10]
	socket_client_begin()	socket_client_end(),
	HTTP	

```

        socket_client_io()    print_html()
socket_client_io()    print_html()

```

```

void socket_client_io()
{
    ...
    value=get_value_from_http();
    certainty=get_certainty_from_http();
    sprintf(write_buffer, "%s. %s.", value, certainty);
    write(client_sd, write_buffer, strlen(write_buffer));
    n=read(client_sd, &read_buffer, sizeof(read_buffer));

    read_buffer[0]=0;
    ...
}

```

```

void print_html_message(char *str)
{
    ...
    message=strtok(str, ":");
    printf("%s\n", message);
    printf("<BR>\n", message);
    printf("<BR>\n", message);
    printf("<INPUT TYPE=\"submit\" VALUE=\"Continue\">\n");
    ...
}

```

```

void print_html_integer(char *str)
{
    ...
    question=strtok(str, ":");
    start=strtok(NULL, ":");
    end=strtok(NULL, ":");
    printf("%s\n", question);
}

```



```

printf("<SELECT NAME=\"value\" SIZE=1>\n");
printf("<OPTION VALUE=\"%i\" SELECTED> %i\n", i1, i1);
for(i=i1+1;i<=i2;i++)
    printf("<OPTION VALUE=\"%i\"> %i\n", i, i);
...
}

void print_html_symbol(char *str)
{
    ...
    question=strtok(str, ":");
    printf("%s\n", question);
    printf("<INPUT TYPE=\"text\" NAME=\"value\" SIZE=\"10\">\n");
    ...
}

void print_html_yesno(char *str)
{
    ...
    question=strtok(str, ":");
    value=strtok(NULL, ":");
    printf(question, value);
    ...
}

void print_html_list(char *str)
{
    ...
    question=strtok(str, ":");
    length=strtok(NULL, ":");
    printf("%s\n", question);
    for(i=2;i<=len;i++){
        list=strtok(NULL, ":");
        printf("<INPUT TYPE=\"radio\" NAME=\"value\" VALUE=\"%i\">%s\n",
            i, list);
    }
    ...
}

```

}

5.

가  
가  
가

Prolog

가

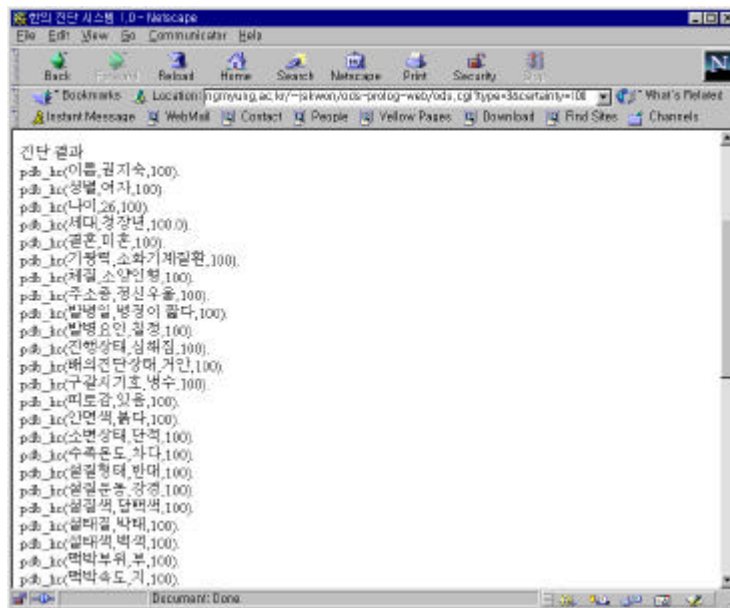
Prolog

Prolog

ETRI

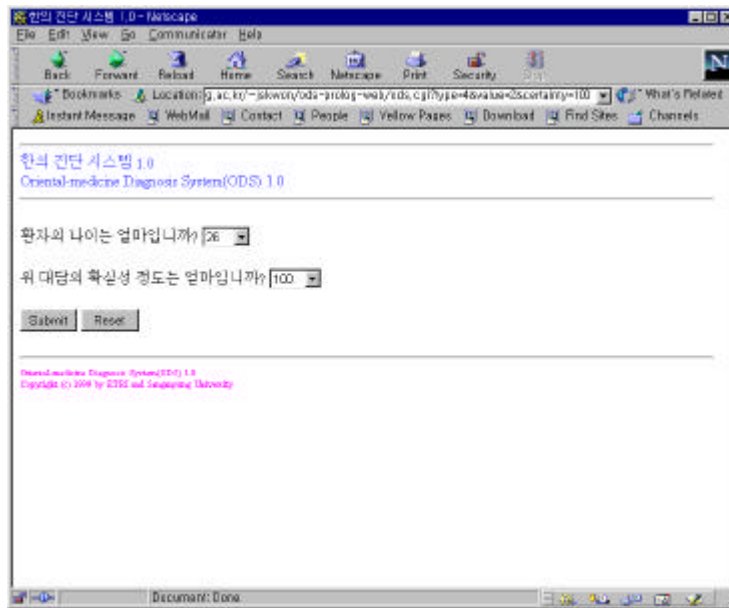
가 [7][15]

5.1.



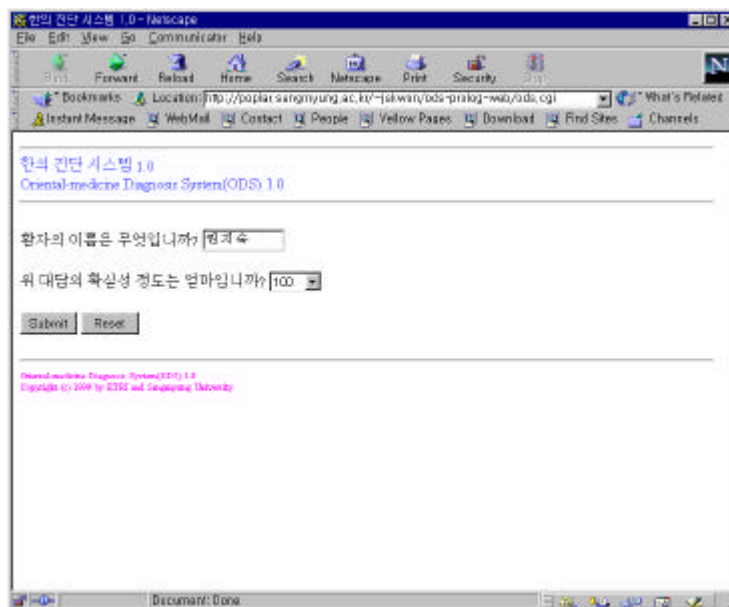
5.1.

5.2.



5.2.

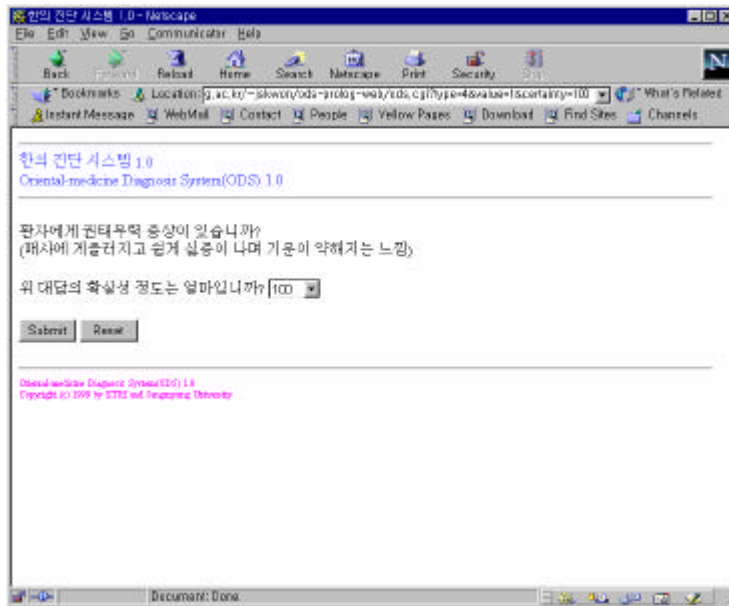
5.3.



5.3.

5.4.

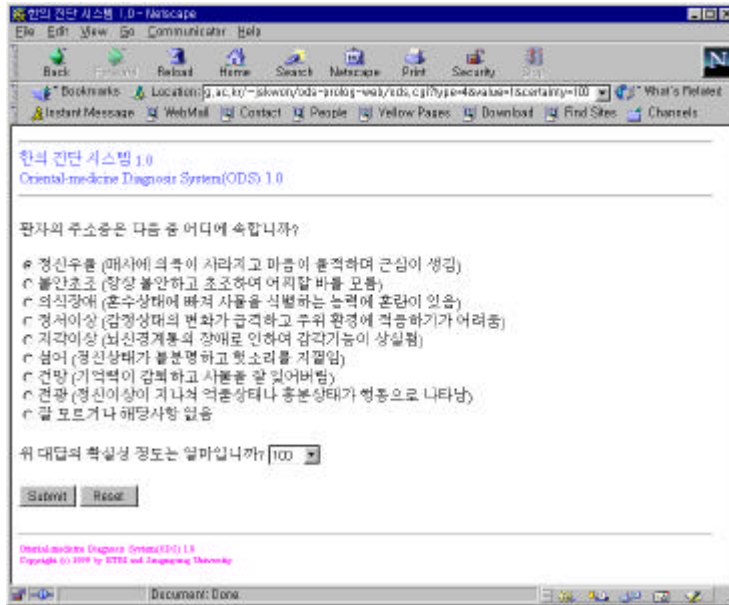
가 +100 가 -100  
 +100 가 가 -100 가  
 가



5.4. ' ' , ' , '

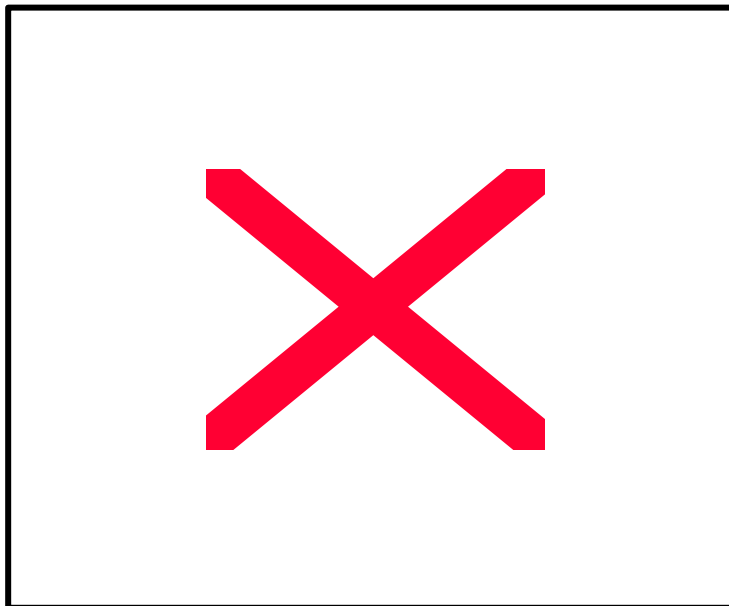
5.5.

'Submit' HTTP  
 Prolog  
 가  
 'Submit' Prolog



5.5.

1



5.5.

2

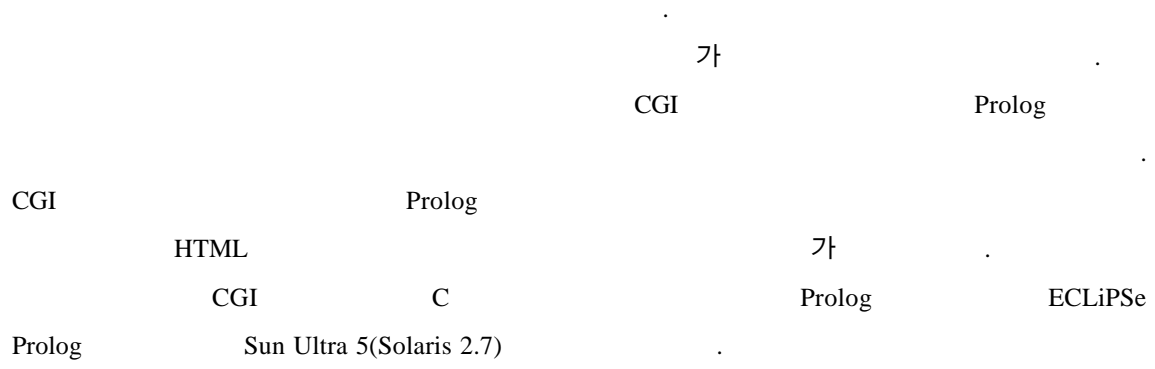
6.

Prolog

Prolog

Prolog

Prolog



- [1] Ph. Bonnet, S. Bressan, L. Leth and B. Thomsen, Towards ECLiPSe Agents on the INTERNET, *Proceedings of the 1<sup>st</sup> Workshop on Logic Programming Tools for Internet Applications*, JICSLP '96, 1-9, Bonn, September 1996.
- [2] I. Bratko, *Prolog Programming for Artificial Intelligence*, Second Edition, Addison-Wesley Publishing Co., 1990.
- [3] D. Cabeza, M. Hermenegildo and S. Varma, The PiLLoW/CIAO Library for Internet/WWW Programming Using Computational Logic Systems, *Proceedings of the 1<sup>st</sup> Workshop on Logic Programming Tools for Internet Applications*, JICSLP '96, Bonn, September 1996.
- [4] S. R. El-Beltagy, M. Rafea and A. Rafea, Practical Development of Internet Prolog Applications using a java Front End. *Proceeding of the 2<sup>nd</sup> International Workshop on Logic Programming Tools for Internet Applications*, ICLP'97, Leuven, July 1997.
- [5] K. Clark and V. S. Lazarou, Distributed Information Retrieval using a Multi Agent System and the role of Logic Programming, *Proceedings of the 2<sup>nd</sup> Workshop on Logic Programming Tool for Internet Applications*, ICLP'97, 87-104, Leuven, July 1997.
- [6] W. F. Clocksin and C. C. Mellish, *Programming in Prolog*, Fourth Edition, Springer-Werlag, Berlin, 1994.
- [7] Y. Lim, D. Shin, S. Kim, K. Kim, S. Park, G. Oh and W. Lee, Knowledge Representation and Acquisition Methods For Oriental Medicine Liver Diagnosis System: OLDS, *The Proceedings of the 2nd International Conference on Computer and Applications*, 770-777, 1987.
- [8] NCSA, The WWW Common Gateway Interface 1.1, <http://Web.Golux.Com/coar/cgi/>, 1997.
- [9] A. Sehmi and M. Kroening, WebLS: A Custom Prolog Rule Engine for Providing Web-based Tech Support. *Proceedings of the 1<sup>st</sup> Workshop on Logic Programming Tools for Internet Applications*, JICSLP '96, Bonn, September 1996.
- [10] W. R. Stevens, *UNIX Network Programming, Networking APIs: Sockets and XTI*, Vol. 1, Second Edition, Prentice-Hall Inc., 1998. Inc., 1998.
- [11] World Wide Web Consortium, HTML 4.0 Spec, <http://www.w3.org/TR/1998/REC-html40>, 1998.
- [12] World Wide Web Consortium, Hypertext Transfer Protocol 1.1, <http://www.w3.org/Protocols/Specs.html>, 1997.
- [13] , , , Prolog , '99 , 26 , 1 , , 81-83, 1999.
- [14] , , , 가 Prolog , '99 , 6 , 2 , , 16-21, 1999.
- [15] , , , , 가 , 37060, , 1990.