ML Server Pages

Peter Sestoft
sestoft@dina.kvl.dk

Christian Stefansen
cstef@diku.dk

Peter Finderup Lund
firefly@diku.dk

2000-02-10

Abstract

ML Server Pages (MSP) is a server-side web scripting language, a loose integration of Standard ML (SML) and HTML in the style of Sun's Java Server Pages, Microsoft's Active Server Pages, or PHP. This talk will present MSP script examples, details of the proof-of-concept implementation, and ideas for future work. Feedback on the idea and design is most welcome.

Introduction to Server Pages

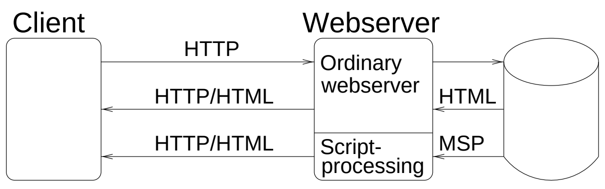
A server page consists of the following:

- Scripts in any supported language of choice. Today Visual Basic, Perl, and Java are most widespread
- Fragments of HTML, CSS, JavaScript, etc.
- Include directives which can include either of the above

When a client requests the page, the webserver compiles and executes the scripts contained in the page and produces the final HTML page which is sent to the client.

Parameters are given to the script through HTTP's `GET` or `POST` and are commonly referred to as the *querystring*.

Architecture



Microsoft Active Server Page (ASP) Source

```
<%
Option Explicit

Function Hello (str)
  Hello = "Hello " + str
End Function
%>

<HTML>
    <HEAD>
    </HEAD>
    <BODY>
        <%=Hello("world")%>
    </BODY>
</HTML>
```

Output

```
<HTML>
    <HEAD>
    </HEAD>
    <BODY>
        Hello world
    </BODY>
</HTML>
```

Sun Java Server Page (JSP) Source

```
<%@ page info="Hello world" %>
<%!
public String hello(String str)
{
  return "Hello " + str;
}
%>
<HTML>
    <HEAD>
    </HEAD>
    <BODY>
      <%= hello ("world") %>
    </BODY>
</HTML>
```

Haskell Server Page (HSP) Source

```
<HTML>
    <HEAD>
        <?H?
hello :: String -> String ;
hello person = "Hello " ++ person ;
        ?>
    </HEAD>
    <BODY>
        <?H=
        hello "world"
        ?>
    </BODY>
</HTML>
```

Also see: `http://losser.st-lab.cs.uu.nl:8080/`

FunTechs ML Server Page (MSP) Source

```
<%
  fun hello str = "Hello " ^ str;
%>
<HTML>
    <HEAD>
    </HEAD>
    <BODY>
        <%= hello "world" %>
    </BODY>
</HTML>
```

The beauty of ASP

```
set maxofperiodeRS=db.Execute("MaxOfPeriodeEmail")

while not maxofperiodeRS.eof
        action=false
        if (dateadd("m",maxofperiodeRS.fields("interval"),
                        maxofperiodeRS.fields("dato").value) < date())
        and (isnull(maxofperiodeRS.fields("naesteudsendt"))) then
                action=true
                %><!-- #include file="e-mailtemplate.asp" --><%
        else
                if dateadd("d", 14,
                           maxofperiodeRS.fields("naesteudsendt")
                            .value)<date() then
                        action=true
                        opgivet=true
                        %><!-- #include file="e-mailtemplate.asp" --><%
                        subject=subjectrykker
                        body=bodyrykker
                end if
        end if
...
```

Advantages of Server Pages

- Much more secure than CGI thereby allowing server administrators to let users create dynamic content without risks. The administrator will still be able to grant varying degrees of access.
- Easy access to parameters, session states, and databases compared to CGI.
- HTML fragments can be generated in more appropriate tools and included afterwards; no need for large inline strings in CGI.
- Wide-spread and understood

Disadvantages of Server Pages

- Being a mixture of many languages and styles, Server Pages tend to become very difficult to read
- More tightly coupled client/server architectures are slowly gaining momentum (e.g. Java servlets).
- Integrates badly with highlevel (e.g. WYSIWYG) web design tools.

Advantages of ML Server Pages

- ML is more elegant and brief.
- The server-side scripting environment is very remote to the programmer. Therefore programming by trial-and-error can be very tedious. ML - through garbage collection, strict type checking, etc. - gives the programmer fewer things to worry about.
- Higher-order functions seem to be a promising tool to perform a variety of tasks including populating tables with data from databases.
- Higher-order functions are convenient and reasonably efficient for generation of HTML code. Also, ensuring correctness of the HTML code is easy.
- Structures and functors allow a degree of modularity and reuse not found in ASP.
- Development time can be drastically reduced.

Disadvantages of ML Server Pages

- Acceptable execution speeds can be a difficult goal to obtain due to the nature of ML and functional programming languages in general.
- Operations on databases and files are inherently imperative.

Current Status

So far we have designed and implemented a usable proof-of-concept system, based on Moscow ML and the Apache webserver.

Future developments Near goals

Near future goals:

- Explore the area and obtain more experience
- Develop the MSP language and the supporting structures/functors
- Aid webprogrammers in changing to ML by creating an environment similar to ASP, but improved significantly
- Implement MSP for Apache using direct APIs (as a module)
- Improve efficiency, scalability, security and functionality (details to follow)

Future developments Far goals

Far future goals:

- Allow and support creation of XML documents as well
- Implement MSP for Microsoft Internet Information Server
- ML Servlets

Security Issues

No script must be able to compromise the webserver on which it is executed in terms of speed, restricted access and stability. The following issues must be addressed:

- Attacks from malicious bytecode
- Source code disclosure
- Infinite loops slowing down the server
- Thread separation
- Restricted file system access
- Many more...

This would make a nice project for someone interested in web security!

Efficiency

The problems involved in obtaining acceptable execution speeds for ML Server Pages are the same as for ML in general (i.e. the need for garbage collection and bytecode interpretation).

- Compile once, execute many times
- Cache frequently used pages in web server memory

Functionality

- A rich set of ML structures for easy HTML/JavaScript/CSS code creation
- Commonly used ASP objects should be available as structures (e.g. `request` and `server`)
- Session and cookie handling
- Access to server API (ISAPI, Apache API)
- Access to other network ports/services (e.g. FTP, SSL, SMTP)
- Database access through ODBC and proprietary APIs
- Third party module interfacing
- Verbose error reporting

Thank you!

Also visit the ML Server Pages website on: `http://ellemose.dina.kvl.dk/~sestoft/msp/index.msp`

Peter Sestoft
`sestoft@dina.kvl.dk`

Christian Stefansen
`cstef@diku.dk`

Peter Finderup Lund
`firefly@diku.dk`