

## Developing High Performance Web Execution Model Using OOPS and Procedural Programming

Mansee Jain<sup>1</sup>, Devashish Puri<sup>1</sup> and Ahmad Shah Abdali<sup>2</sup>

<sup>1</sup>Department of Computer Science, Chitkara University, Punjab, India

<sup>2</sup>South Asian University, Delhi

<sup>1</sup>[mansee.jain@chitkara.edu.in](mailto:mansee.jain@chitkara.edu.in), <sup>1</sup>[manisha.oswal56@gmail.com](mailto:manisha.oswal56@gmail.com),

<sup>1</sup>[devashishpuri@gmail.com](mailto:devashishpuri@gmail.com), <sup>1</sup>[mannatnanda1@gmail.com](mailto:mannatnanda1@gmail.com),

<sup>1</sup>[rainavreet@gmail.com](mailto:rainavreet@gmail.com), <sup>2</sup>[ahmadshah.2008@hotmail.com](mailto:ahmadshah.2008@hotmail.com)

### Abstract

Now, human progress is on the verge of virtual world with the introduction of YouTube, Wikipedia and Facebook. In this work, we are using the best feature of both OOPS and Procedural Programming in order to achieve the goal of developing high performance web execution model. Client and server are integral part of any web-based model. Here, we are integrating compiler and program in web based model and finally come with high performance web execution model. High performance system means taking less time in compare to other existing traditional models. Many web based service like YouTube and Wikipedia are benefitted by taking advantage of our laziness. We want everything on the go, just in our hand-held computer. This work is also solving the beginners' laziness problems like setup of Programming Environment such as compilers etc. If our education system is being shifted online, then this may turn out a big setback in learning process. Thus, we are providing compiler on the go so that the learning process is not interrupted by unavailability of resources (Our whole system is client-side) and of course by the laziness of people.

**Keywords**—High performance, Web Execution Model, Server, Client, Procedural, Object Oriented Programming

### 1. Introduction

A compiler is an intermediate that helps to convert source code to any other computer language.

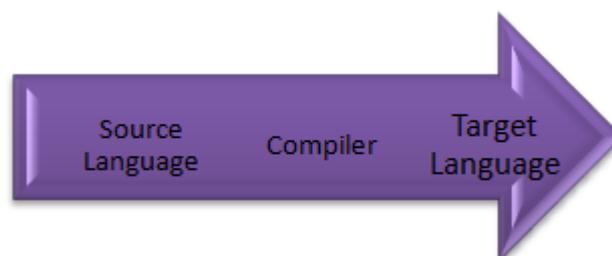
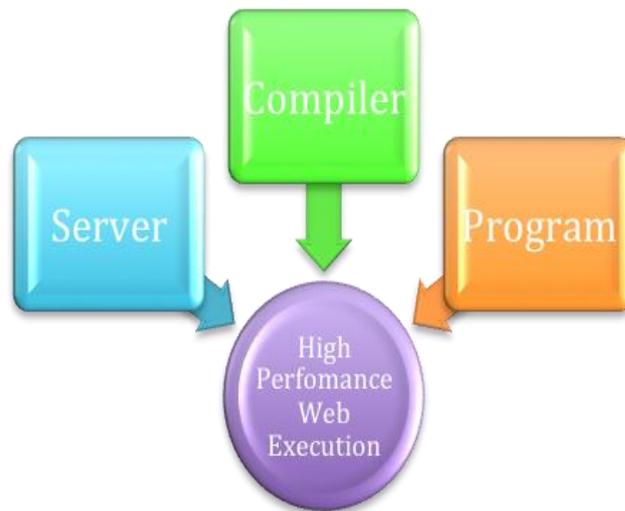


Figure 1. Basics of Compiler

The program, we are creating will transform source codes from many languages to the language that a browser or a similar program (app) understands, *i.e.*, JavaScript. So, it can be termed as a COMPILER. Translating these source-codes can be beneficial for web execution of programs written in situations like some on-line test series or coding contests, on-the-go implementation of ideas *etc.* Web execution of such programs is done

by calling compilers of respective language at server-side which takes a considerable amount of time but translating the code would speed up the process which is must for educational purposes.



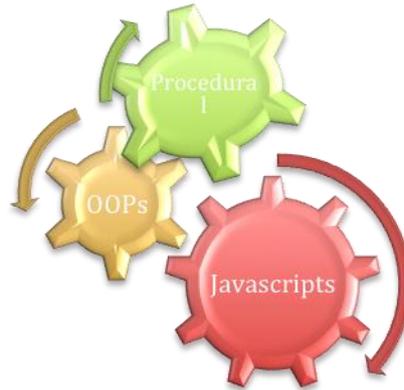
**Figure 2. Components of High Performance Web Execution Model**

This service may benefit a large set of people such as beginners who don't know how to setup a coding environment and also the output can be treated as a normal string and the user experience continuous monitoring instead of trying everything without a direction and suffer intense frustration.



**Figure 3. Constituents of Program in Computing**

A program consists of majorly three steps *i.e.*, Input, Operations and Output. For most languages the operation part is similar but there is variation in Input and Output. Thus translating would include replacing user's code's INPUT/OUTPUT with JavaScript's INPUT/OUTPUT. Some languages follow the concept of OOPS while some don't. Thus JavaScript can be good Alternative as it follows the concept of OOPS and can also simulate PROCEDURAL Programming System.



**Figure 4. Techniques Behind our Proposed High Performance Model**

## 2. Related Work

The execution of composite services is integral part of service-oriented systems [1]. In practice, service invocation is performed by client components (stubs) that are generated from service descriptions at design time [1]. It requires an object representation (*e.g.*, Java classes) of the XML data types specified in service descriptions to be generated and meaningfully integrated in the client code at design time [1]. Mapping from PLC ontology model to function block model and code generation is implemented based on the ontological knowledge base and semantic query-enhanced web rule language is discussed in [2]. Web script crashes and malformed dynamically generated webpages are common errors, and they critically impact the usability of Web applications [3]. The proposed approach in [4] not only composes Web services by adding mediation net to deal with message mismatches, but also checks the compatibility w.r.t. Temporal constraints by generating modular timed state graphs. The capability to develop bendable business applications is one of the eventual objectives behind the use of Web services [5]. Selecting an optimal web service among a list of functionally equivalent web services still remains a challenging issue [6]. The Sensor Web connects live sensors and Earth Science models (ESM) [7]. “A workflow is one approach for designing, implementing, and constructing a live link between sensors and ESM[7]”. The Sensor Web consists of many individual Web services [7]. In distinguish to conformist multi-page Web applications, an Ajax application is frequently developed as a single-page application in which content and formation are changed at runtime according to user connections, asynchronous messages acknowledged from the server and the current state of the application [8]. Efficient coding is used for design of energy efficient hardware [9] and simulation of high performance human brain [10]. We are using this efficient coding approach for web execution model. Basic design of electronic circuit [11].

## 3. Working

Web Languages like HTML and JavaScript are compiled when we open the webpage in our browser. Thus if we want to execute some code on our Browser, it can be easily done by translating the language used to JavaScript. JavaScript has a function eval (argument) which evaluates the JavaScript Expression given as argument. Thus we can evaluate user-defined string and we can translate the code with provided language to JavaScript. Translation Process is done as:

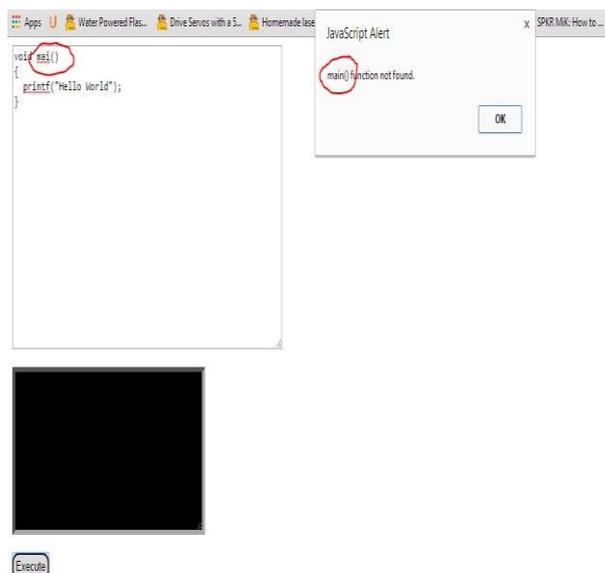
```
while(1){  
if(exe.indexOf('int ') == -1)  
break;  
else  
exe = exe.replace("int","var");  
}  
while(1){  
if(exe.indexOf('float ') == -1)  
break;  
else  
exe = exe.replace("float","var");  
}
```

**Figure 5. Replacing Data-Types to “var”-(JavaScript Variables)**

Checking for Function Return type and replacing them with “function”-(Describes JavaScript Function).

```
while(1){  
if(exe.indexOf('void ') == -1)  
break;  
else  
exe = exe.replace("void","function");  
}  
if(exe.lastIndexOf('main(') <4)  
{  
alert("main() function not found.");  
return 0;  
}
```

**Figure 6. Checking for Function Return type and Replace with “function”**



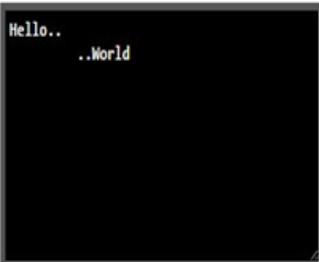
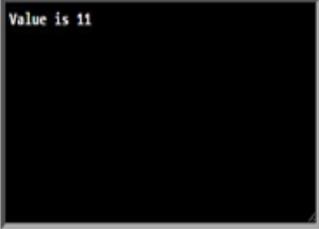
**Figure 7. Translating Predefined Functions with User-Defined Arguments**

```
function printf(arg)
{
  for(var i=1;;i++){
    if(arguments[0].indexOf('%d') == -1)
      break;
    else
      arguments[0] = arguments[0].replace("%d",~~arguments[i]);
  }
  document.getElementById('output').value += arguments[0];
}
```

Figure 8. Argument Object of JavaScript Works

#### 4. Output

We all know JavaScript selects the data type of variables automatically depending on the value defined to variable. But we can typecast variables using some symbols like `~~` typecasts to 'INT' etc. This approach is used when `%d` is recognized but value is not Integer Type. Figure 9 shows six different types of JavaScript code and its related output.

Execute	
<pre>void main() {   printf("Hello..\n\t ..World"); }</pre>	
<pre>void main() {   int a=1,i;   for(i=0;i&lt;10;i++)     a++;   printf("Value is %d",a); }</pre>	
<pre>void main() {   print(); } void print() {   printf("Hello World!!!"); }</pre>	

<pre>int a=10,i; for(i=1;i&lt;10;i++) {   a++; } printf("value of a %d\n",a); printf("Size of Int %d",sizeof(int));</pre>	<pre>value of a 19 Size of Int 4</pre>
<pre>void main() {   int a=1,i;   for(i=0;i&lt;10;i++)     a++;    printf("Integer value of 12.55 : %d",12.55); }</pre>	<pre>Integer valu of 12.55 : 12</pre>
<pre>void main() {   int a=1,i;   for(i=0;i&lt;10;i++)     a++;    printf("Size of Int %d\nSize of String %d",sizeof(a),sizeof("hello")); }</pre>	<pre>Size of Int 4 Size of String 5</pre>

**Figure 9. Sample Output of Web Execution Model**

## 5. Conclusion

Our social environment is heading towards a virtual world with the introduction of YouTube, Wikipedia and Facebook. Whole world is going online and many people are participating in Online Sessions for getting education. User-Interaction can be highly increased and made fun with ready-to-use snippets or modules where user can practice online what he's learning. This highly increases the chance of user's sticking to the course'. As everything is done client-side, the user doesn't have to wait for a part of code to compile at server-side which can take much time. Users can even merge languages and notice the professional, productive and practical aspect of course which gives them a reason to immerse in learning as they can see a product coming (such as game-mechanics with simple '+' operation).

## 6. Future Scope

Many web developers are benefitted by taking advantage of our laziness like YouTube and Wikipedia. We want everything on the go, just in our hand-held computer. My Project is solving the beginners' laziness problems like setup of Programming Environment such as compilers *etc.* If our education system is being shifted online, then this may turn out a big setback in learning process. Thus we are providing compiler on the go so that the learning process is not interrupted by unavailability of resources (Our whole system is client-side) and of course by the laziness of people. Portability can be highly increased by providing the full language support and merging the language features, like a module system. This would also increase the pace of development as some things are easily implemented in *some* languages like Implementation of 2D arrays and multidimensional arrays is simpler in C than JavaScript. We would have our developer tools on-the-go and way more powerful than normal tools and this opens the doors of many unimaginable possibilities as beginners are way more imaginative than experienced and with more power they can surely change the world.

## References

- [1] F. Paganelli, and D. Parlanti, "A dynamic composition and stubless invocation approach for information-providing services", *IEEE Transactions on Network and Service Management*, vol. 10, no. 2, (2013), pp. 218-230.
- [2] W. Dai, V. N. Dubinin and V. Vyatkin, "Migration from PLC to IEC 61499 using semantic web technologies", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 3, (2014), pp. 277-291.
- [3] S. Artzi, A. Kiezun, J. Dolby, F. Tip, D. Dig, A. Paradkar and M. D. Ernst, "Finding bugs in web applications using dynamic test generation and explicit-state model checking", *IEEE Transactions on Software Engineering*, vol. 36, no. 4, (2010), pp. 474-494.
- [4] Y. Du, W. Tan and M. Zhou, "Timed compatibility analysis of web service composition: A modular approach based on Petri nets", *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, (2014), pp. 594-606.
- [5] Q. Sheng, Z. Maamar, H. Yahyaoui, J. Bentahar and K. Boukadi, "Separating operational and control behaviors: A new approach to Web services modeling", *IEEE Internet Computing*, vol. 14, no. 3, (2010), pp. 68-76.
- [6] W. Ahmed, Y. Wu and W. Zheng, "Response Time based Optimal Web Service Selection", *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, (2015), pp. 551-561.
- [7] G. Yu, L. Di, B. Zhang and H. Wang, "Coordination through geospatial web service workflow in the sensor web environment", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 3, no. 4, (2010), pp. 433-441.
- [8] A. Marchetto, P. Tonella and F. Ricca, "Reajax: a reverse engineering tool for ajax web applications", *IET software*, vol. 6, no. 1, (2012), pp. 33-49.
- [9] B. Pandey and M. Pattanaik, "Low Power VLSI Circuit Design with Efficient HDL Coding", *IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, Gwalior, India, (2013) April 5-8.
- [10] L. Kumar, "Simulation of High Performance Energy Efficient Human Brain on 28nm FPGA", *IEEE International Conference on "Computing for Sustainable Global Development (INDIA COM)*, Bharti Vidyapeeth, Delhi, (2015), March 11-13.
- [11] K. Kartik, "HSTL Based Low Power Thermal Aware Adder Design on 65nm FPGA".

