A Survey of Object Oriented JavaScript Language

F. Fawzia Khan
Research Scholar,
Karpagam University
Department of Computer Science,
Coimbatore

R. Mallika, PhD
Assistant Professor,
Deaprtment of Computer Science,
CBM College,
Coimbatore,

ABSTRACT

Many of the programmers move towards object oriented programming languages due to its reusability nature, whereas in procedural programming languages reusability is not allowed, which may contribute to the high cost of development. Most of web browsers make use of JavaScript as a user interactive program that is a client side program due to its active nature. In the client side programming it is practiced to enable user interaction, control browser activities and alters the content changes made by user dynamically. On the server side programming language, it is used for desktop and mobile application growth. JavaScript is mainly applied for maturation of object oriented web application development because of its supportability of inheritance via prototyping and also properties and methods. In this work the various object oriented concepts are discussed. The methods and operation supported by the object oriented JavaScript language is likewise talked about. Its nature of operating methods, type refinement of JavaScript language, various access control mechanisms over JavaScript language is discussed. The enabling of transaction control over JavaScript language is also elaborated.

Index terms: JavaScript, Type Refinement, Graphical user Interface

1. INTRODUCTION

Object oriented programming (OOP) is a programming model based on objects instead of logical representation. The object is generally a bundle of related state and behavior of software. Object oriented programming makes use of user abstraction to create a model which is relevant to the real world. In the OOP, instead of considering the logical operation required to manipulate the program, it will take account of what objects are needed by the programmers to manipulate them. In the OOP processing of operation will be done in the following two ways. Those are

- Data modeling
- Method

1.1. Data Modeling

In the data model, the objects that are needed by the programmers to manipulate will be gathered and the relations between those objects will be identified.

1.2. Method

Once the objects are identified, then those objects will be bundled together, which is called as a class. In the class of objects, the kind of data as well as logical sequence existing among them will be presented. There are many rules and properties that are supported by the OOP. Those are inheritance, class, data hiding, data corruption, interface, package, polymorphism, and encapsulation. There are many programming languages that support OOP concepts today. The first OOP programming language was Simula. Other than Simula, the programming languages that support OOP concepts are Java, Python, C++, Visual Basic.Net and Ruby.

Web based application is the one of the most popular application that is used by everyone today. There are many scripting languages available for web application development. Many of the web pages require flexibility provided to the users with dynamic support. JavaScript is an effective mechanism that is used to write object oriented programming languages. In the client side programming language which requires a user interaction, JavaScript is mostly used. JavaScript libraries make use of OOP concepts to create a code which is easy to manage and maintain.

JavaScript is a prototype based language because it makes use of prototype for inheritance rather than using class for inheritance. OOP is supported by the JavaScript when creating a child object through inheritance, Encapsulation and type polymorphism. The inheritance concept is used to create a child object which will make use of all of the properties present in the parent method.

The internal details of parent object can be hidden by using encapsulation which prevents from external access to collide the properties and methods defined. The three types of encapsulation levels are known as private, protected and public level. The type of parent can be changed by using type polymorphism without any modifications. Among these principles two important principles of using OOP in JavaScript are encapsulation and inheritance.

2. RELATED WORKS

This survey analyses the importance of object oriented concepts used by the JavaScript programming language. The various concepts and mechanisms present in OOP are discussed in order to understand the importance and usage flexibility of the object oriented concepts in today's world. The various programming languages which are used to convey the OOP concepts are also discussed. Among those languages, the importance of JavaScript is discussed.

Oscar Nierstrasz [1] discussed various OOP concepts and the importance of those mechanisms in the real world environment. OOP becomes popular not only by the programmers who use it to solve their problem but also because of its flexibility over user interfaces, operating systems and databases. Smalltalk effort is a main reason for the popularization of the OOP which utilizes both the classes and inheritance concept. Simula was the first programming language which uses an object as a programming construct.

The creation of the object is not only flexible for programmers; it is also useful for prototyping and application development.

The most important concepts that are supported by the OOP are encapsulation and inheritance. Encapsulation is used to hide the behaviour of objects, whereas the behaviour of properties can be shared among objects through the inheritance. This work studies mainly about the reusability properties of objects, its various types and its concurrency properties.

Effective reusability can be achieved by better encapsulation of an object's property and its methods. An object can be reused by instantiating it either statically or dynamically by the users. Programmers can instantiate the objects based on object classes with the visibility properties to provide access control over the objects. The object's type is same as the object's class and in object's type the user will define the static behaviour of objects.

These properties of OOP reduce the development cost of building applications through its reusability properties. The issues present in OOP design are, complexity in decomposition of the entire application into objects corresponding to the problem, selecting most appropriate objects to solve the given problem and selection of objects should adapt with the software base.

Maya Hristakeva et al [2] discussed the various programming languages that are based on OOP concepts, whereas in the previous works only the OOP concepts are discussed. This work argues about the way of interpreting the OOP concept in the real time problems. Some of the object notions that are supported by the OOP programming are classes, inheritance, encapsulation and polymorphism.

Class: It is a logical way of grouping the methods and attributes which are used to solve the specific problem.

Abstraction: It is used to provide a simplified view to the users by hiding the complex details of the classes.

Inheritance: Through this mechanism the behaviour of objects can be shared with each other. The child object which is inherited from the parent object can have the parent properties and as well as its own properties.

Encapsulation: It is used to hide the sensitive information of objects from others.

Following Simula (1967), Smalltalk (1970) is the next programming language which involves object oriented concepts totally. C++ (1980) is the next OOP language which is introduced by incorporating OOP concepts with the C programming language. In 1990 C++ programming is improved and introduces a new programming language called java. In 2005 C# is generated by incorporating the properties of both C++ and Java. In scripting languages Ruby and Python are the one which follows the OOP concepts. Concluding from the analysis of all these languages Eiffel is the effective OOP language which balanced features and provide safe use by the programmers. Each and every OOP language has its own specific feature which turns every language as best at some specific point. Hence no-one can conclude that any programming language as better than other language.

Scripting language which is based on OOP concepts are mainly used for creating the user interactive client side and server side web page programming. Ronald P. Loui [3] discussed importance of scripting language in the real world and how it becomes more powerful than other OOP programming languages. There are more scripting languages like PHP, Perl, Cold Fusion which supports OOP concepts. These languages are most traditional languages and prototypical one. JavaScript and Ruby are the OOP scripting language which can capture the imagination of the real world. Perl becomes more flexible to the programmers to use, which provides best server side programming language. It makes use of common gateway interface to run the server side programming languages. With effective GUI support and reusability properties of OOP scripting languages, it becomes programmer friendly. The early scripting languages allow the programmer to modify the old program with some small modification to create the new scripting program. The main concern present in all programming languages is its syntax and its semantics. The programmers need to understand the syntax and semantics of the programming languages effectively, which will give an effective handling of languages.

Simon Holm Jensen et al [4] discussed about the type analysis of the JavaScript language since it is the main scripting language for the server side programming and web browsers. It is the essential language for the modern web applications which cause the programmers to write easy understandable language rather than writing complex code. The JavaScript language can be understood by reading its flow present among them. The flow of the JavaScript programming language contains following kind of representation:

Declare Variable: it declares a variable needed to solve the program

Read Variable: It will read the value of the programming language into the temporary variable

Write Variable: It is used to write the value of programming variable into the temporary variable

Constant: It is used to assign a constant value to a temporary variable if none of the values are assigned.

Read Property: it is used to read the properties of each and every created object.

Write property: By using this, property can be assigned to object

Delete Property: It is used to delete the properties assigned to the objects which are not needed in future.

Holger M. Kienle [5] discussed about the importance of JavaScript in the today’s world where the usage of user interactive application is increased. JavaScript can be defined as an ugly duckling. JavaScript is supported and became popular in the both sides of browser vendors and web application developers. The compiler and interpreter technologies are combined with the JavaScript language in order to maximize its speed of execution. Web based application needs to develop effectively in order to replace the desktop application. This can be achieved by using JavaScript which needs only the smallest amount memory to execute the JavaScript code with faster execution.

JavaScript languages are mostly written by the programmers which in turn today, has caused the availability of many sophisticated tools to write the JavaScript. For example, Google Web Toolkit, is used to convert the java files written by the programmers directly into the JavaScript. Nowadays a number of projects are making use of JavaScript as a programming language.

Rodolfo Toledo et al., [6] discussed various access control mechanism present in the JavaScript. JavaScript can provide basic access controls to the developers who can change the contents of the web page dynamically. However, for accessing and modifying sensitive resources over a JavaScript coding needs help of some of the external tools. In this work Easy access control (ZAC) library is developed for providing access control over a JavaScript application in an extensible, expressive and modular way. This newly introduced access control library mechanism can execute the programs safely without bug which contains an unsafe construct. It is achieved due to its dynamic enforcement nature.

Due to its dynamic enforcement nature, one can introduce new concepts in the middle of program execution, which can reduce the programming execution time as well as preventing from any scripting bugs. It cannot be implemented in the object capability model because of its dependency with the objects. It also governs large number of access control specifications without consideration of the users varying scenarios.

An access control in JavaScript object is controlled by using two properties. Those are rules and actions. The rule defines the type of access over the specific resources, whereas actions define the access to be taken when specific access control over a JavaScript object occurs.

One of the functionality that can be supported by the JavaScript language is providing the security concern over the untrusted guests. Security concerns over the web application are increased due to its nature of using untrusted third party JavaScript programs in order to increase their advertisements, libraries and widgets. Mohan Dhawan et al., [7] analysed about the transcript mechanism which can provide the security concern from the untrusted JavaScript code. The transaction system introduced in this work consists of the following main features. Those are JavaScript transactions, the transaction suspends/ Resume, Speculative DOM updates. It prevents the system from the malicious guests by understanding the nature of the each and every guest code, and refining the malicious guest code from it and by doing forensic analysis. The system can be protected from the guest OS by embedding a guest OS within a transaction construct. These functionalities of transcript mechanism conclude that the prevention against untrusted guests can be provided by incorporating the functionalities of JavaScript with the transactions.

Nataliia Bielova et al. [8] discussed the various types of security policies that are occurring while using the web applications concurrently. This study mainly focuses to provide the security for the research scholars. The various security relevant components of the web browsers and the variety of security policies are analysed by the security providers. In this work following security enforcement techniques are discussed. Those are dynamic techniques based on run time monitoring and secure information flow control. In the dynamic execution technique, the execution of programs will be checked for whether it satisfies the security policies or not. In the security enforcement techniques, the flow of information within the program is analysed in a secured manner.

In order to prevent the clients from the untrusted guest access, the secured API is developed between the clients and guest hosts. In this work lightweight self protecting JavaScript is implemented which is based on the aspect oriented programming in order to implement the security policies and enforcement mechanism in the remote JavaScript. This protocol can be added to any type of HTML page during runtime in a secured manner.

In JavaScript based applications, static analysis is the one which can provide an effective performance over many of its functionality. However the precision value of static analysis cannot be predicted accurately in the case of JavaScript. Vineeth Kashyap et al. [9] proposed a type refinement strategy for JavaScript which can improve the precision value considerably. In this work, type refinement is done on various levels of implicit condition statement declarations. And this work also evaluates a different strategy for incorporating a type refinement in the application.

The papers discussed above shows that the importance of Object Oriented JavaScript as a scripting language over many of the real time user interactive web applications. And the several ways to provide a security refinement control over the different application interface with the untrusted guest environment is discussed.

3. CONCLUSION

The reusability and the user interactive nature, observe the concentration of programmers turn to the JavaScript side. The effectiveness and applicability of reusability method in many areas are discussed deeply in this paper. The various analyses discussed in this survey, prove the interactive nature of the JavaScript in the client programming and large growth in the server side programming as well. The different functionalities in the JavaScript which is used for controlling access of client side and server side programming are analysed. The various studies are discussed in this analysis about the functioning of the JavaScript language over many web applications. And it also discussed about the security evaluation over application interfaces, which may lead to the cryptographic attacks. The effective type refinement process in the JavaScript is discussed in order to improve the precision of the JavaScript language. Thus the discussion of these studies concludes that Object Oriented JavaScript is the most powerful scripting language which allows users to share their data among the untrusted guest JavaScript coding with better precision value.

4. REFERENCES

[1] Oscar Nierstrasz, “A Survey of Object-Oriented Concepts”, ACM Press and Addison Wesley, pp. 3-21, 1989.

[2] Maya Hristakeva, RadhaKrishna Vuppala, “A Survey of Object Oriented Programming Languages”, Univ. Of California, 2000.

[3] Ronal p. Loui, “In praise of scripting: Real programming pragmatism”, Published by the IEEE Computer Society, 2008.

[4] Simon Holm Jensen, Anders Møller and Peter Thiemann, “TypeAnalysisforJavaScript”, Aarhus University, Denmark, 2009

[5] Holger M. Kienle, “It’s about time to take Javascript (more) seriously”, IEEE Software published by the IEEE Computer Society, 2010.

[6] Rodolfo Toledo and Éric Tanter, “Access control in java script”, In university of chile, 2011

[7] Mohan Dhawan,Chung-Chieh Shan,Vinod Ganapathy, “Enhancing JavaScript with Transactions”, Published in Proceedings of ECOOP'12, 2012

[8] Nataliia Bielova, “Survey on JavaScript Security Policies and their Enforcement Mechanisms in a Web Browser”, Journal of Logic and Algebraic Programming, May 2013.

[9] Vineeth Kashyapy John Sarracinoy, z John Wagnery Ben Wiedermannz Ben Hardekopf, “Type Refinement for Static Analysis of JavaScript”, ACM 978-1-4503-2433 5/13/10, 2013.