
An Exploratory Study on Object Oriented Web Development

Janne Umapathi¹, Dr. Vishal Khatri²

¹Research Scholar, Department of Computing and Information Technology, Himalayan University, Itanagar, Arunachal Pradesh, INDIA.

²Research Guide, Department of Computing and Information Technology, Himalayan University, Itanagar, Arunachal Pradesh, INDIA.

Abstract: Web application developers are increasingly advised to adopt an object-oriented programming (OOP) approach due to the growing complexity of application development and the related requirement to efficiently manage resources. While current design practises and development environments abstract away from the details of implementation technology, are not exceptionally versatile and performs a mid-reasonable job of planning for potential future upgrades or retention. This research is an attempt to screen out the significance and influence of OOP in web applications. Contextual research has been performed and the descriptive study has been discussed for the insights. Insight into the features, techniques, and efficient ideas of web application development is provided in this research work.

Keywords: Descriptive study, Object oriented programming (OOP), Web development,

Introduction

The Web has developed into a worldwide platform for delivering a wide variety of applications, from lightweight, transient services to massive, enterprise-level workflow systems spread across a network of servers. The widespread availability of Web browsers makes applications with HTML-based front ends universally accessible across platforms (Mazinanian & Tsantalís, 2016). Web-based software distribution also has the additional benefit of enabling the use of virtualization and centralised maintenance, which allows for the rapid and cheap deployment of programme upgrades. Numerous HTML-based apps have emerged as a result of the Web's popularity and its advantages as a client-server platform, although the development of Web applications is still largely ad hoc (Khan, 2021). The majority of existing Web application development and management techniques rely on the knowledge and experience of individual developers because there is no method that is rigorous or methodical that can be used.

The primary misconception about web development is that it is an authoring challenge, not a software development problem that can be solved by applying established methods and research. This is due to the fact that the Web implementation model doesn't quite map onto the most cutting-edge approaches to creating software. It is challenging to develop frameworks that can record architectural design decisions for the purpose of reusing those decisions in different areas of an application or in different application projects due to the absence of acceptable abstractions (Kim, 2021). Maintenance and advance Web-based programmes turns out to be a tedious task due to lack of structured representations. In a low-level implementation, it is quite difficult to keep track of design decisions; as the programme develops, modifications can very easily result in inconsistencies. Poor maintainability is a significant issue for web-based applications because of their propensity to experience rapid evolution, marked by frequent upgrades and redesigns (Abid et al., 2020).

The objective of this research work is that this article will provide light on fundamentals of the OOP paradigm and its influence in the field of web application development. The findings of our research will also help web developers and designers become familiar with the most common characteristics of web applications as well as the most widely used programming languages, development environments, and techniques.

Literature Review

Development of OOP

In contrast to modern web development practises, early web implementations relied solely on the low-level tools which did not facilitate the high-level absorption for the purposes of sharing and reusing. As proper representations were insufficient, it was difficult to design platforms that enabled the reuse of many reusable modules. Software engineers, nevertheless, became aware of the necessity for modularization of concerns in order to accommodate large and sophisticated systems as web applications matured in the late 2000s (Lumyong et al., 2021). To facilitate object-oriented principles such as modularization, segmentation, and abstraction, technologies

like the Extensible Markup Language (XML) and Document Object Model (DOM) were generated. Through this, the barrier between high-level and low-level implementation was narrowed. The development of modern web application frameworks and tools has been greatly aided by the introduction of object-oriented concepts. Research into this field has typically concentrated on topics like linguistic contrasts and the chronological development of online applications (Jaiswal & Heliwal, 2022).

Web Applications

Article on web application trends, the status quo, security related challenges in web applications, and web framework principles and idea were reported and discussed (Onu et al., 2022). The similarities, distinctions, and weaknesses of PHP, Python, and Ruby were compared and contrasted. Elements such as each language's history, popularity, evolution, features, popularity, syntax, security, performance and semantics in perspective of web application contexts were evaluated. An analysis on PHP was conducted and provided a summary of some of its shortcomings. A paradigm for object-oriented online applications was identified after a survey of object-oriented web apps. Research identified XML's contribution to technology, which enables high-level abstractions for design-level modelling in a markup language. Object-oriented frameworks were analysed, and the positive and negative characteristics of the frameworks were determined with the connection between the technologies used to construct web applications and the level of security. The object-oriented design was analysed using link analysis and conducted research involves the implementation of design patterns in PHP-based web applications.

Design and technology

Web 1.0's existence through the years 1990 and 2000 was considered to be an early stage in the history of the World Wide Web, which personally witnessed some level of growth with support of the multiuser interface, distributed and hyperlinked documents, single point maintenance and updates (Pereira dos Reis et al., 2021). Popularity and user engagement were still quite low at the time, as the majority of web1.0 sites were static and functioned using a brochure-style architecture. Because of the fact that few professional web designers created the content that people could access on websites, this was the case. Web 2.0 has experienced a huge boom in popularity and interest from businesses of all sizes and Internet users all over the world, making it the robust platform it is today. Users are responsible for producing the content, while firms are solely responsible for providing the platform, in the architecture of participation used by Web 2.0 (Koopman et al., 2021). The majority of websites on the internet today, such as wikis, social networking sites and blogs all display user-generated content, bringing about a change from a small number of powerful experts to a large number of users with increased agency.

Rich Internet Applications (RIA) and the AJAX (Asynchronous JavaScript and XML) technologies are two examples of the many tools that have been developed as a substitute for the development (Silva et al., 2022). Both of these technology allow more user interaction. Web applications that mimic the feel and functionality of desktop software can be built using these technologies. The idea of Object-Oriented Programming is enconced at the foundation of the Web 2.0 revolution sparked by technologies such as AJAX, the Document Object Model (DOM), rich Internet application (RIA) frameworks. Understanding these resources and OOP is, thus, essential.

OOP with CSS

The code's organisational structure is the primary emphasis of OOCSS (Robin Nixon, 2012). Users will be able to code reusable objects of styles while using OOCSS. When working with OOCSS, users will be working with reusable objects, and each object will have the style rules for an HTML structure that will carry a certain type of information. Separating the structure from the skin and the container from the content is made easier by adhering to the OOCSS guidelines. The most common referenced instance is the OOCSS media object.

Discussion on the exploratory study

In a typical client-server architecture, a user's web browser acts as the client, sending requests to the server and receiving responses from the server. Internet programme creation can be done in a number of ways. There are several variables to consider when deciding which method of development to use, including the data's complexity, the size of the application, the frequency of content updates, and the developer's level of expertise. The usual steps

in creating software or online applications are research to determine what features are needed, information analysis, architectural design and specifications, team collaboration, coding, testing, bug fixing, implementation, and maintenance. To produce software more quickly with fewer resources, Agile Software Development employs a number of techniques (Savage, 2015). Systems with rapidly shifting user needs are a common fit for agile development approaches.

Further, object-oriented techniques have been proposed by several authors (Dutonde et al., 2022; Pokryshen, 2022) throughout the years to solve current problems in the online world. Decomposing Web applications into their constituent parts is the goal of Web Composition in object-oriented paradigm with specified characteristics of each element like web page, website. Using this method, creators can be assured that web app will have consistent support for software reuse, high-level abstraction, run-time administration, and maintenance. W3Objects wraps web elements as objects with well-defined interfaces. All objects can be inherited to create new ones. W3Objects is a powerful tool for solving the issue of transparent object migration and maintaining referential integrity. Jessica is an HTML based object-oriented abstraction model which has core of the system implemented in an Object-Oriented Language, with other important part is a compiler that translates the system's abstract description into highly dynamic Web services (Wu et al., 2019).

Conclusion

The OOP approach is applicable to both the development of web application and stand-alone software development, making it the current adaptable programming pattern used by any software developer or programmer. As an aid to creators, OOP includes the following: The less code a developer has to write, the better, so long as the application is a developer. Enable the reusing of code functions and design: when creating a web application, a developer does not have to create all of the code from scratch. As it allows for the incorporation of code from a third-party programme, it opens the door to cyber-attacks. The openness of the object-oriented design pattern to incorporating objects from outside sources makes the programme vulnerable to assault. This study facilitated to understand and identify the various group of web application and its development.

Despite the few challenges associated with integrating object-oriented concept in terms of resource recycling, maintenance, extensibility, delegation and scalability, the study suggests that the implicit impacts of an object-oriented approach on web application development are observed. The consequence has been considerably beneficial, as web apps have become increasingly popular. Therefore, OOP has prompted the development of numerous frameworks for creating applications that execute over the Internet.

References

- [1] Abid, C., Alizadeh, V., Kessentini, M., Ferreira, T. do N., & Dig, D. (2020). *30 Years of Software Refactoring Research: A Systematic Literature Review*. <https://doi.org/10.48550/arxiv.2007.02194>
- [2] Dutonde, P. D., Mamidwar, S. S., Sunil Korvate, M., Bafna, S., & Shirbhate, D. D. (2022). *Website Development Technologies: A Review*. *10*. <https://doi.org/10.22214/ijraset.2022.39839>
- [3] Jaiswal, P., & Heliwal, S. (2022). Competitive Analysis of Web Development Frameworks. *Lecture Notes on Data Engineering and Communications Technologies*, *93*, 709–717. https://doi.org/10.1007/978-981-16-6605-6_53/COVER
- [4] Khan, G. (2021). Traditional implementing usage of core programming languages in the object oriented programming. *Asian Journal of Multidimensional Research*, *10*(10), 1167–1173. <https://doi.org/10.5958/2278-4853.2021.00983.6>
- [5] Kim, M. C. (2021). *Learning Early-Stage Web Development at Scale: Exploring Methods to Assess Learning Through Analysis of HTML and CSS* [Drexel University ProQuest Dissertations Publishing]. <https://www.proquest.com/openview/8dc55704c3e9200d5fad777b6c5415d9/1?pq-origsite=gscholar&cbl=18750&diss=y>
- [6] Koopman, P., Michels, S., & Plasmeijer, R. (2021). Dynamic Editors for Well-Typed Expressions. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12834 LNCS*, 44–66. https://doi.org/10.1007/978-3-030-83978-9_3/COVER
- [7] Lumyong, C., Tachasriburapha, N., & Titijaroonroj, T. (2021). Proposed Structural Validation-based Testing for Object-Oriented Programming. *KST 2021 - 2021 13th International Conference Knowledge and Smart*

- Technology, 96–101. <https://doi.org/10.1109/KST51265.2021.9415812>
- [8] Mazinianian, D., & Tsantalos, N. (2016). *An empirical study on the use of CSS preprocessors*. <https://doi.org/10.1109/SANER.2016.18>
- [9] Onu, F. U., Kelechi, D. A., Uchenna, O. F., Kelechi, I., & Okorie, K. M. (2022). Integrated Unified Crime Information Management Sys. *International Journal of Information Technology and Electrical Engineering*, 2278–3075. <https://doi.org/10.35940/ijitee.H9115.0711822>
- [10] Pereira dos Reis, J., Brito e Abreu, F., de Figueiredo Carneiro, G., & Anslow, C. (2021). Code Smells Detection and Visualization: A Systematic Literature Review. *Archives of Computational Methods in Engineering 2021 29:1*, 29(1), 47–94. <https://doi.org/10.1007/S11831-021-09566-X>
- [11] Pokryshen, D. (2022). Finished Software Products in Training Specialists of It Industry. *Open Educational E-Environment of Modern University*, 12(12), 90–102. <https://doi.org/10.28925/2414-0325.2022.129>
- [12] Robin Nixon. (2012). *Learning PHP, MySQL, JavaScript, and CSS*. O'Reilly. <https://www.oreilly.com/library/view/learning-php-mysql/9781449337452/>
- [13] Savage, T. (2015). Componetizing the web. *Queue*, 13(8), 1–20.
- [14] Silva, E. F., Dembogurski, B. J., & Semaan, G. S. (2022). A literature review of computational thinking in early ages. <https://doi.org/10.1080/09669760.2022.2107491>
- [15] Wu, W.-L., Hartono Budianto, I., Wong, C.-F., & Ken-En Gan, S. (2019). A Review of Apps for Programming: programming languages and making apps with apps. *Scientific P Hone Apps and Mobile Devices*, 5(1). <https://doi.org/10.30943/2019/25012019>