Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 3 Issue 12, December - 2016

Beyond the Hype: Issues and Perspectives in the Software Paradigm for Internet Computing

Francisca O. Oladipo, PhD
Computer Science Department
Federal University Lokoja
Lokoja, Nigeria
francisca.oladipo@fulokoja.edu.ng

***Abstract*— The demand for open and dynamic internet had given rise to a new software paradigm for computing on the internet: Internetware. The software for internet computing is expected to exhibit certain desirable properties and conform to specific software models. The dynamic nature of the present day internet however had given rise to changing and non-deterministic component-based software system parading unclear functional and non-functional requirements. As the internet is in every facet of our society life, there is the need for different software services and components (most with unpredictable requirements and component behaviours) to collaborate in solving societal problems. This state of the art in Internetware was examined in this paper. All approaches and architectural blueprints required to define a software paradigm in a dynamic internet environment were emphasized and the requirement for self-adaptation and transposing from conventional software architecture were explored. This work revealed that beyond the hype, the paradigm of Internetware is basically a contemporary software architecture designed to be autonomous and adaptable, trustworthy, situational and emergent while being cooperative.**

Keywords—Internetware; software, behavior; evolution; adaptive-systems; modeling

I. INTRODUCTION

Internetware is a paradigm that came to being as a result of the demand for a model of software system that could drive the dynamic present day internet, involving collaborations between different internet-based services needed for different software services and components (most with unpredictable requirements and component behaviours) to collaborate in solving societal problems. This could involve the development of new applications or a transformation of legacy applications to new systems that are able to emerge, be autonomous, cooperative, be able to evolve, trustworthy, etc.

Historically, this software paradigm took his root from works by researchers from Chinese Peking and Nanjing Universities in the early part of the millennium [1]; and the research has since then grown across Chinese institutes resulting in the publication of several research articles and position papers as well as organized conferences and symposium on the subject [2]. Several methodologies, approaches, engineering adaptations and platforms have been developed for Internetware by several researchers.

This aim of this paper is to provide an introduction to Internetware. The author examines the state of the art in Internetware research, highlighting all the approaches and architectural blueprints for the development of and the overall required properties for a software paradigm in a dynamic internet environment.

II. INTERNETWARE: FROM TAXONOMY TO STATE OF THE ART

Between 2003 when Chinese Researchers from Peking and Nanjing Universities pioneered research into defining a software paradigm for internet programming and providing software engineering support for internet computing, several research perspectives had emerged and fundamental challenges uncovered.

A summary of the findings of the research results by several researchers and members of the Internetware communities over Europe, USA, China and Japan, covering over 10 of research activities were presented by [3]. The researchers investigated the fundamental challenges to Internetware and concluded that in order to continue to support Internet computing from a software engineering point of view, three perspectives of software model, operating platform and engineering sensitivity-based design must continue to be considered and addressed.

Working forward from a previous research on agent-based software model, [4] proposed a software-structuring model for environment-driven Internetware applications. The basis of this research was provided by [5], which involved proposing an agent-based approach to a model to coordinate open internet programming. While the primary focus of the research was to improve object orientation and resource sharing in an open network system through provision of supports for structural, developmental and evolutional aspects of Internetware; the present research involved analyzing the behaviours and technical hiccups of environment-driven applications, and provision of an initial framework for the development of context-aware and self-adaptive software application systems in the open network environment. A summary of the achievements of both researches involve the structuring and separation of the logic from the software services at the structural level, provision of

third-party coalition applications and services at the developmental level and reconfiguration of the applications' coordination models at the evolutional aspects. Building a structuring model within the context of the environment in order to address the problems of proving an open and dynamic paradigm in a network environment characterized by uncertainties.

[6] introduced and coined a novel definition for autonomous component in Internetware by showing the feasibility of Internetware based on finding a group of autonomous components that can collectively achieve providing a software paradigm for internet computing. Their research involved a tuple of goals-based autonomous model for Internetware in a goal-driven (top to bottom) refinement and co-operation-based (bottom to up) composition. Future research directions by the authors will involve building an environment that would provide support for automated assembly of Internetware from autonomous components. Similar researches include a provision of a description of how a set of interfaces provided to an environment or required by an environment can be specified and composed at both interface and design levels by [7]; proposition of a common component description language to unify and integrate the structure and behavior of software components and components-based system by [8]; and the definition of a framework for unifying interface signature, interface constraints, interface packaging and configurations, and non-functional properties of software components in the programming paradigm for the internet by [9].

A new technique for Internetware was introduced in a research by [10], this new technique named architecture-based component composition (ABC), which originally provided support for traditional software construction was found to be able to effectively support Internetware as a result of the features which demonstrated its suitability for conventional software engineering. The work adapted ABC to Internetware by deploying its feature oriented domain modeling to structure the software entities (which were originally unordered) to ordered Internetware, adapting the architecture centric design and analysis method to provide support for the design of self-adaptive Internetware and finally, using the component operating platform to provide self-adaptation and runtime infrastructure middleware for Internetware.

After identifying self-adaptation as one of the key features of Internetware, [11] developed a software architecture (SA) centric self-adaptation Internetware. The system could monitor its runtime behavior and adjust based on some pre-define parameters –the SA. The approach involves representing and changing the runtime state and behavior of Internetware in the form of runtime software architecture. Consequently, the system was able to achieve automatic analysis and decision making by capturing, organizing and presenting the knowledge for self-adaptation in the form of software architecture.

A similar research by [12] provided a brief summary of the state of the art in Internetware research. From the analysis of various authors, the work proposed a technical framework comprising of four components based on four identified aspects for Internetware. These components are software model, the middleware, the methodology and quality assurance. In addition to the proposed paradigm, issues and future directions in Internetware arising from a study of ongoing research were discussed. This position is shared by [13] who equally identified four enabling technologies that are critical to Internetware to be the lifecycle model, relevant software ontology system, modeling and simulation and social ranking which can be used to improve on the traditional software testing and validation approaches to establish the trustworthiness of any Internetware application.

III. Internetware Framework

Resulting from all the research results above, it is evident that the framework for Internetware applications is based on architecture (software); this architecture framework is made up of three distinct segments;

1. The software model which is concerned with modeling the Internetware entities and their relationships, specifying the collaborations between them as well as the operating environments
2. The middleware is the autonomous part responsible for provision of all operations and services of the Internetware applications. Common middleware is the Software Architecture (SA) which had defined with varying characteristics by providers to enable the self–organization and self-adaptation of Internetware applications.

The development methodology of Internetware is based on the "Software Architecture of the Whole Lifecycle" and every stage of the development of the Internetware application is controlled by this architecture [12].

IV. Discussions and Conclusion

The definition of Internetware has been expanded beyond the paradigm for Internet computing to include all self-adaptation software systems that can detect and self-adapt to changes in their operating environments, in addition to being able to dynamically alter its structures and exhibited behavior. Internetware design should allow the system to be open and cooperative to environment software in order to permit modification of runtime behaviours and an ability to come with unexpected stimuli from its entities coupled with the strong fault tolerance.

Internetware systems are generally required to exhibit the behavior and goals specifically defined by their designers. The systems though distributed are autonomous entities but can be self-adapting according to changes in operating environments. They may equally be required collaboratively mutate to for new applications and pervasive. For a successfully development and deployment of Internetware systems, four major considerations are key: specification of the software model, which may be based on existing and

well-known methodologies like OOADM; the operating environment or runtime space where the systems' entities can collaboratively evolve and the specific engineering sensitivity which will be based on strictly following the stages of a development approach. Finally, there must be a definition of metrics to measure the success of the Internetware applications.

REFERENCES

[1] Y. Fuqing, J. Lv, and H. Mei, "Some Discussion on the Development of Software Technology," Acta Electronica Sinica (in Chinese), vol. 26, iss. 9, pp. 104-115, December 2003.

[2] H. Mei, H. Gang, and T. Xie, "Internetware: A Software Paradigm for Internet Computing," IEEE Computer, vol. 45, iss. 6, pp. 26–31, June 2012.

[3] H. Gang, M. Xiaoxing, and T. Wei-Tek, "A new software paradigm for Internet computing," Oxford University Press on behalf of China Science Publishing & Media Ltd, vol. 1, iss. 2, pp. 168-169, June 2014.

[4] J. Lu, X. Tao, and X. Ma, "An agent-based software model for Internetware," Science China Series E-Tech Sci (in Chinese), vol. 35, iss. 12, pp. 1233-1253, 2005.

[5] J. Lu, X. Ma, X. Tao, C. Cao, Y. Huang, et al., "On environment-driven software model for Internetware," Science in China Series F-Information Sciences 51(6), 683-721, Springer. June 2008.

[6] W. Jiao, P. Zhu, and H. Mei, "Modeling Internet-Based Software Systems Using Autonomous Components, "Chinese Journal of Electronics, vol. 15, iss. 4, pp. 593–598, October 2006.

[7] Z. M. Liu, J. F. He, and X.S Li, "Contract-Oriented Development of Component Software," In the proceedings of IFIP WCC-TCS2004, Toulouse, France, pp. 349–366, April 2004.

[8] T. Teschke and J. Ritter, "Towards a Foundation of Component-Oriented Software Reference Models," in Gregory Butler, Stan Jarzabek (Eds.), Generative and Component-Based Software Engineering, Second International Symposium, GCSE 2000, Erfurt, Germany, October 9-12, 2001 Revised Papers. Lecture Notes in Computer Science 2177 Springer, pp. 70-84.

[9] J. Han, "A Comprehensive Interface Definition Framework for Software Components," in the proceedings of the 1998 Asia-Pacific Software Engineering Conference, Taipei, Taiwan, 110–117, 1998.

[10] H. Mei, G. Huang, H. Zhao, and W. Jiao, "A software architecture centric engineering approach for Internetware," Science in China Series F: Information Sciences, vo. 49, iss. 6, pp. 702-730, December 2006.

[11] H. Mei, G. Huang,L. Lan and J. Li, "A software architecture centric self-adaptation approach for Internetware," Science in China Series F: Information Sciences, Springer, vol. 51, iss. 6, pp. 722–742, June 2008.

[12] H. Mei, and XZ. Liu, "Internetware: An emerging software paradigm for Internet computing," Journal of Computer Science and Technology, vol. 26, iss. 4, pp. 588-599, July 2011.

[13] WT. Tsai, Z. Jin, and X. Bai, "Internetware Computing: Issues and Perspective," International Journal of Software and Informatics, Institute of Software, Chinese Academy of Sciences, vol. 3, iss. 4, pp. 415-438, December 2009.