An Outlook of State-of-the-Art Approaches in Functional Testing of Web Application

M.Y. Suhaila, W.K. Wan Mohd Nasir, *Member, IAENG*

***Abstract*— The numerous recent contribution of web application testing approaches reflect the rising awareness and concern for assuring web application's functionality and performance. The functionality of a web application cannot be compromised, especially if dealing with critical web-based applications such as online banking system, ticketing system and file hosting services. This paper aims to systematically investigate state-of-the-art functional testing approaches for web applications. Focus will be given on testing techniques that address the functional aspect of a web application. In this paper we firstly perform exhaustive literature reviews on state-of-the-art approaches in functional testing of web application. Next, a comparative evaluation framework that consists of several related criteria is developed, and subsequently applied to the selected prominent approaches. Finally, we present a critical discussion on the results of the comparative evaluation. Based on our observations, the evaluation results may indicate the current and future direction in web application functional testing and the viability of these approaches. Through the results, we will be able to determine the direction of functional web application testing and which area within the functional requirement aspect of web applications that has not duly being addressed.**

***Index Terms*—Functional testing, Software testing, Test automation, Web application testing**

I. Introduction

WEB applications (WAs) are immensely popular, due to their capability to reach to a mass of users instantly once the applications are deployed. Due to the popularity of WA testing, the need of an effective testing approach is indispensable. A WA represents the image of its organization to the mass Internet users. Errors discovered on an organization's WA may compromise its public image and reputation. To some extent, it may even cause a considerable amount of financial loss to the organization if the errors are related to the critical processes in the WA. Therefore, it is imperative that WAs are tested adequately before they were released, and tests were done periodically after deployment in order to discover hidden, unexpected errors that were previously undiscovered.

The dynamics of WAs have motivated various different testing approaches to be developed and introduced by researchers. While some approaches are taken from traditional testing approaches and later adapted to suit WA testing, other approaches are introduced specifically for testing WAs. Nevertheless, they all share the same expectation, which is discovering errors and 'bugs' that may compromise a WA's conformance to its specified requirement [4].

There have been numerous approaches introduced to test web applications. Of course, the main aim of these test approaches are to discover failures in the required services/functionality, in order to verify the conformance of the application behavior with the specified requirement [4].

To achieve this, various tools have been introduced. The testing of web applications has been led by industry, whose techniques have been oriented primarily toward validation of non-functional requirement [9]. However, there has been constant effort by several researchers to produce testing approaches and techniques that deals with the functional requirement of web applications. Thus, in this paper, we explore several testing approaches that have been introduced in the last five years. Focus will be given on testing techniques that address the functional requirement aspect of a web application.

This paper is organized as follows. First, we describe the general issues that are inherent in WA testing recently. Then, an overview of the state-of-the-art of WA testing is presented. These approaches are specific to functional testing of WA only where works from several researchers will be discussed. Subsequently, we will evaluate these approaches meticulously so that appropriate classes can be used to categorize them.

II. GENERAL ISSUES IN WA TESTING

Even though testing is important in ensuring an application's conformance with its requirement, developers tend to disregard the testing aspect in favor of early software deployment, out of fear of losing out to their competitors. This tendency often led to a higher cost incurred in correcting failures uncovered in the software much more later which, if detected early will not cost as much. Therefore, negligence in testing is not feasible in reducing software development cost.

Furthermore, Rapid Application Development (RAD) and Agile Software Development method is now a preferred software development methodology by developers due to its

Manuscript received January 7, 2011; revised February 7, 2011. This work was supported in part by the University Tun Hussein Onn Malaysia under the Short Term Grant Fund and University Teknologi Malaysia under the Fundamental Research Grant Scheme.

M.Y. Suhaila is with the Department of Software Engineering, Faculty of Computer Science & Information Technology, University Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia (phone: 60-7-4538198; fax: 60-7-4532199; e-mail: ysuhaila@ uthm.edu.my).

W.K. Wan Mohd Nasir is with the Department of Software Engineering, Faculty of Computer Science & Information System, University Teknologi Malaysia, 81310 Skudai, Johor, Malaysia (e-mail: wnasir@utm.my).

shorter developing time. To test an application that has been developed using this method is considered a challenging task, mainly due to frequent changes in user requirement and lack of the necessary documentations during the development phase [11]. The inherent issue is whether performing testing every time after a change is made is considered practical.

Hieatt and Mee [12] propose that test-first programming is the best approach for this development scenario. This type of agile testing favors unit testing to drive application's development, it offers advantage in ensuring code reusability. A particular open-source test tool called Selenium utilizes the same approach [14]. The popularity of this approach can be seen through the constantly growing of Selenium's community of users. Alternatively, testers may also implement other agile testing approaches. The next section will present recent works related to functional testing of WAs.

III. OVERVIEW OF WA TESTING

Over the last few years, much progress have been discovered in WA testing albeit there is still room for introducing new techniques and improving the ones that are readily available. Why is there a need for a diverse number of testing approaches? Is the current ones are not adequate enough?

In order to be able to accommodate the constantly rising number of users, a WA has to be regularly maintained, and tested every time changes are made. The functionality of a WA cannot be compromised, especially if dealing with critical applications such as online banking system, ticketing system and file hosting services. Much is at stake here, in terms of financial resources and information that are of utmost importance to its users. Thus, the rapidly evolving of web technologies calls for a diverse implementation of test approaches and techniques to uncover errors and failures in various aspects and finding ways to minimize or even resolve them.

Test approaches are developed to satisfy the testing of functional and non-functional requirements of a WA. The functional requirement testing would refer to testing from the application's point of view, in order to uncover flaws that may inhibit an application from achieving its functional purposes. Whereas testing the running environment (i.e. hardware, software and middleware components) perspective of a WA would satisfy the non-functional requirement that is usually attributed with issues such as performance, compatibility, security and other issues that reflects the well-being of an application [4]. However, it is unfair to say that one perspective is more important than the other as both types have different test aims and objectives and cover different test areas. Rather, both types are of equal importance in WA testing, thus a combination of both test types will generate a better test results than using only either one. However, the test approaches discussed here will focus on the functional requirement aspect of testing WA.

Each day, the complexity of application increases, while the resources and testing time decreases [17]. The ever increasing challenges faced in testing WA have brought forth many new and state-of-the-art testing approaches. According to [4], these state-of-the-art testing approaches generally falls into three types of strategies; white box testing (implementation based), black box testing (responsibility based) and gray box testing (a combination of implementation and responsibility based). Each of the strategy requires different set of software testing skills and resources. However, sometimes these three can be implemented simultaneously to complement each other. Nguyen et al. [17] propose that gray box testing is essential for an effective WA testing which normally consist of a large number of hardware and software components, due to the ability of gray box testing in uncovering problems that are not easily perceived by black box or white box testing, especially problems of end-to-end information flow and distributed hardware/software system configuration and compatibility.

IV. WEB APPLICATION TESTING APPROACHES

The previous section provided an overview of WA testing. This section will describe recent state-of-the-art works on WA functional testing approaches.

A. Input Validation Testing

The first work proposed a white box technique that implemented input validation testing of WAs [1]. User inputs are partly essential in determining the accuracy of an application's output. Even if each input in a WA is validated, the control mechanism itself needs to be tested to ensure that it is performing as expected. Thus, the need of test approach in this particular area arises. This approach produce test cases for input validation testing using TIVT (Tool for Input Validation Testing), a prototype testing tool from Georgia Institute of Technology. The test cases are obtained by recovering input validation model in the form of Validation Flow Graph from the respective WA's program source code automatically. The approach effectively detected most of the expected errors, with a few tester managed to detect all expected error in a WA. While the input validation area remains as one of the important features in a WA testing that has received fair amount of attention from researchers, other feature-oriented WA testing also has potentials for exploration.

B. Navigation Flow & Control Flow Model Testing

Ricca and Tonella [2] propose an approach for white box testing that is more suitable for static WAs. This approach focused on the navigation model at the higher level and control flow model for the lower level of a WA. Test cases are defined through structural coverage of a WA obtained by tracing its navigation and control flow. However, the work is far from producing a definitive conclusion, as there is still room for further improvements and exploration. Nuo et al. [18] later extended the approach in the aspect of utilizing navigation model as test model in a proposed model-driven methodology. A framework is also proposed to support the methodology and mainly features a modeler and tester. Even so, automatic generation of test data is only partly achievable and the flexibility of the test model can be emphasized through enabling rules establishment for

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

modifying models.

C. Event Flow Testing

Another proposal introduces a new testing coverage for WA testing using event flow testing technique [5]. Data flow testing technique is applied to an enhanced dependence graph called event-based dependence graph model which is newly introduced. However, this technique exploits the event-driven feature in the .NET environment and is only particular for testing WAs in the said environment.

D. Object-Oriented Model Testing

Di Lucca et al. [6] propose a two-stage black box testing approach which comprises of a unit test (testing a single page) as the first stage and an integration test (testing a set of Web pages) as the second stage. The approach utilizes a WA object-oriented model as test model through the implementation of decision tables. The test models are then tested using a testing toolkit called WAT (Web Application Testing) which automated the testing process. Although the tool managed to automate testing process and reduce test time, this approach is only applicable for pages with feasible paths.

E. Finite State Machine (FSM) Model

Andrews et al. [7] utilizes FSM for modeling software behavior and deriving test cases from them. Called FSMWeb technique, it is realized by building FSM model of a WA's subsystem, the subsequences of a state in FSM is combine to generate the complete test. Furthermore, the author proposes an automated tool that can support certain phases in the approach. However, the proposed tool is only conceptually sound. Therefore realization of the proposed tool is requisite before its effectiveness in supporting the approach can be determined empirically.

F. System Requirement Model

Another work proposes combinatorial testing, which is a low-level functional-test technique that has a wide application across the software development lifecycle [8]. The proposed approach generates several sets of test inputs automatically from system requirement models developed by testers. The potentials of this approach are present in revealing failures that are otherwise failed to be noticed in self-developed models. A further advantage is due to its nature that ensures each value is sufficiently covered.

G. Input, Output, Preconditions and Effects (IOPE) based Testing

Paradkar et al. proposed a test approach that is specified at testing web services [10]. Still, the possibility that certain aspect of the approach can be used in a general WA testing must not be ruled out. This approach makes use of the Precondition and Effect pairs in the IOPE information that is usually present in a web service and refines them based on fault models to generate testing goals. The testing process is automated using the planner component and the test cases are then validated by a verification sequence generated in the approach. The adequate savings in effort for requirement coverage and effectiveness of the fault detection is apparent in the approach. The work can be extended by making the approach adaptable to general WA to some extent.

H. User Session Data Manipulation

Gray box testing is a hybrid of white box and black box testing. Its primary aim is testing a piece of software against its specification but using some knowledge of its internal working [4]. This testing strategy is well-utilized in a WA functional test approach in the aspect of manipulating captured data from user sessions to produce test suites and includes them into a white box testing implementation for fault detection purposes [9]. By comparing and combining both techniques, the effectiveness of the techniques is presented through finding different types of fault.

I. Test Suite Reduction

Sampath et al. [13] expands the works by [9] further, by proposing a technique on minimizing the size of test suite by clustering the user-sessions using concept analysis. Compared to the original test suite that has not been minimized, the reduced test suite produces lesser result. However, the author states that it is not a significant loss and the notion of reducing test suites can be explored further with other alternate algorithms which may produce a higher fault detection capability.

J. Functional Scenario Testing

Huang et al. [15] proposed that the UML extended activity diagram is utilized to produce test cases. The work is focused on the functional scenario testing aspect of a WA as functional scenario can be depicted by the activity diagram. The test cases are essentially in the form of testing codes based on HttpUnit, a web testing tool. The generation of the test cases is done using Web Application Scenario Automated Testing Tool (WASATT). The test codes can then be compiled and run by the testers. Huang et al. proposed that this approach can potentially reduce artificial errors which are usually present when testers try to self-defined test models inaccurately. The work can be further extended to completely model all functional properties of WA, accurately validate web page's dynamic semantics and accommodate the behavior of concurrent user access.

K. Test-First Design Approach

Antawan and Marc provide an acute description of Selenium, a functional test tool for WA [14]. Selenium, an open-source project for in-browser testing uniquely offers a possible test-first design of WA for its users, a customer acceptance test and an automatic regression test bed for web-tier. As an agile testing tool, Selenium's capabilities in writing and maintaining test scripts surpasses other testing tool such as Canoo WebTest, HttpUnit and QuickTest Professional. Furthermore, Selenium allows its users to write test codes in a variety of programming languages, thus eliminating the need of having different test tools for WAs developed in different languages and eases testing task for testers. The author further reports that while Selenium's current performance is adequate, its future potential in becoming a powerful WA testing application is reflected by the constant growth of Selenium's active community.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

L. Test Case Design Approach

The test case design approach which is proposed by [16] utilizes UML sequence diagram of a WA to generate three types of test cases which will be tested level by level, starting with the smallest unit (single web page testing), followed by mutual web page testing and lastly integrated web page testing. The information of the test case was derived solely from sequence diagram and testing was performed with the aid of a testing tool called OnlineTestWeb. The approach demonstrated was fairly simple and straightforward and easy to comprehend. Unfortunately the tool used was no longer available on the Internet.

V. Comparative Evaluation Framework

This section describes the framework for systematic evaluation of the WA testing approaches discussed in the previous section. The evaluation framework consist of criteria that are organized into three aspects namely test approach strategies, test coverage and comparative evaluation of state-of-the-art WA functional testing approaches. The evaluation is imperative in order to distinguish features which are unique in every approach.

A. Test Approach Strategies

The objective of this comparison is to discover which aspect or area of WA testing has the highest concentration in terms of research works lately. Analysis is done in order to identify other potential parts/area of WA testing that has yet to be researched.

In terms of test strategies, white box testing scores a high level of concentration ([14], [1], [2], [5] and [3]). This type of WA testing is highly recommended for testers who are well versed in the target application's internal structure as they can make use of the internal knowledge to produce highly credible test results. For black box testing, the works produced are equally numerous ([6], [7], [8], [10], [15] and [16]). Black box testing is recommended for testers who are the end-user or those who are not familiar with the internal knowledge of the target application.

In the case of gray box testing, only several work are produced ([9] and [13]). In order for a tester to perform gray box testing, a combined external and internal knowledge of a target application is prerequisite. Few testers possess both the internal and external knowledge of a target application. Hence the source of difficulty in producing test approach in this area. This leads to hidden and under-discovered potentials for this testing strategy [19].

B. Test Aspects

All of the test approach mentioned covers different aspects of WA testing. The evaluation performed in this section is to assess the conduciveness of areas in WA testing.

Most of the approaches introduced have covered almost all aspects of system testing and component testing in WAs. The conduciveness of the research area in WA testing can be seen through numerous works produced. Integration testing, release testing, performance testing, test case generation and several other areas have seen their fair share of contributions. However, interface testing can be classified as an under-developed research area, as there are still several issues that are not addressed adequately due to the presence of volatility in interface communications. For example, the output value of a WA may also require a suitable testing technique as opposed to input values.

Researchers also need to address more issues arising in dynamic WA testing such as page navigation, infinite looping, validation control, data flow and other taxing issues. As more and more of interconnected dynamic web pages emerge, the needs of a highly flexible yet efficient test approach are also definite. Greater efforts are required in addressing testing issues for dynamic WA [4].

The key feature that should be present in most state-of-the-art approaches is economic value, due to the rapid nature of producing WA nowadays. Reduction of test suite/case but greater coverage is highly desirable. Other techniques that may also contribute to cost and resource saving in WA testing is in demand by developers and should be focused on by future works.

C. Comparative Evaluation Criteria

In order to systematically evaluate the viability of the current approaches in WA functional testing, it is important to define a number of relevant criteria used in the evaluation. Therefore we suggest that the comparative evaluation framework to include the following criteria under two key components: breadthness and depthness for test coverage, while practicality, usability and simplicity under test effectiveness. Further detail is presented in Table I.

Test coverage determines that the testing approach covers these aspects of web application functional testing (i.e. navigation testing, validation control testing, interface testing, and JavaScript testing). Two criterions that are associated with this component are breadthness and depthness.

TABLE I
Rubric of the Comparative Evaluation Criteria

Components of Comparative Evaluation	Criteria	Description
Test Coverage	Breadthness	Measures if all the aspects of web application testing are covered.
	Depthness	Measures whether each aspect is explored intensely or fully.
Test Effectiveness	Practicality	Capability of the test approach in incorporating features that will completely or partially resolve the testing issues that it is considering
	Usability	Easiness of applying the process.
	Simplicity	How straightforward the testing processes are.

Test Effectiveness evaluates the test approach in terms of the degree of success in producing its intended result. The criteria considered include practicality, usability and simplicity.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

VI. COMPARATIVE EVALUATION RESULTS

From the comprehensive evaluation of state-of-the-art WA testing approaches, the results are obtained and summarized in Table II. Through the results, appropriate conclusion is formulated.

TABLE II
COMPARATIVE EVALUATION ON STATE-OF-THE-ART WA FUNCTIONAL TESTING

Test Approach / Features	Testing Criteria				
	Test Coverage		*Test Effectiveness*		
	Breadthness	***Depthness***	***Practicality***	***Usability***	***Simplicity***
Input Validation [1]	Low	High	High	Medium	Medium
Navigation Flow Model [2]	High	Medium	High	Low	Medium
Control Flow Model [3]	Medium	Medium	Medium	Low	Low
Event Flow [5]	Low	High	High	Medium	Medium
Object-Oriented Model [6]	High	Medium	Low	Medium	High
Finite State Machine Model [7]	High	High	Medium	Low	Low
System Requirement Model [8]	Medium	High	Medium	High	Medium
IOPE-based [10]	High	Low	High	Low	Low
User-Session Data [4]	High	Medium	High	Low	Low
Test Suite Reduction [13]	High	Medium	High	Low	Medium
Functional Scenario [15]	High	Low	High	Medium	Low
Test-first design [14]	High	High	High	Medium	Medium
Test Case Design [16]	High	Medium	Medium	High	High

In reference to test coverage, most of the approaches have adequate coverage of breadthness, with the exception of two approaches only ([1] and [5]). As for the intensity of the test approach, only two approaches ([10] and [15]) does not reach a satisfactory depthness level. However, the test coverage comparative evaluation results does not indicate a weakness of the approaches mentioned above, rather it indicates that every approach have a different testing aim and specific testing issues that it wishes to address.

Regarding to practicality, only [6] is identified as quite complex and thus reduce its practicality. Fortunately this approach scores higher in other criterions. But for other approaches, practicality is not their weakest point. Therefore most of them indicate a high economic value.

In terms of usability, half of the approaches ([2], [3], [7], [10], [9] and [13]) are quite difficult for first time implementation and requires a certain level of experience and knowledge before implementing them.

For simplicity, some of the test approaches requires little amount of time and resources and are quite straightforward ([6] and [16]). While others are not as straightforward ([1], [2], [5], [8], [13] and [14]), and several other approaches are low on simplicity based on the process involved in the implementation of the approach ([3], [7], [10], [9] and [15]). Testing process needs to be straightforward and simplified in order to minimize testing cost and time. Therefore further works for simplifying or automating parts or all of the complex test process are imperative to enhance the viability of the approach.

VII. CONCLUSION AND FUTURE WORK

This paper has presented the state-of-the-art approaches introduced for functional testing of WAs. We also propose the comparative evaluation framework that consists of various criteria classified under two key components namely test coverage and test effectiveness. Based on the evaluation results, we found that in terms of testing strategies, hybrid testing (gray box testing) lacks adequate contribution.

Regarding test aspects, several aspects of WA's functional testing namely interface testing and dynamic WA testing requires further contributions. The key feature for future contribution is the economical value of the approach.

Concerning the overall evaluation result of state-of-the-art WA functional testing approaches, we found that the test-first design approach [14] and test case design approach [16] are better than other approaches based on the overall result. However, finite state machine model approach [7] is superior in test coverage. For test effectiveness, input validation approach [1], event flow approach [5], and system requirement model approach [8] are on par with [14]. However, [16] are the most effective state-of-the-art functional testing WA approach .

We hope that the results presented in this paper may provide an outlook of the current research and indicate the direction for future research. Future efforts to continuously map the unexplored aspects of WA testing is highly encouraged as WA will possibly remain as the primary choice of developing application for a long time due to its dynamics and widespread capabilities.

We are currently continuing our research in several directions. First, we aim to develop our own functional testing approach for WA application that will contribute to the enhancements and improvements of existing approaches. Next, we attempt to develop tools that support the automation of the approach. For our research, we hope to produce a functional test approach for WA that satisfies most of the criteria incorporated in our comparative evaluation framework.

REFERENCES

[1] Liu, H. and T.H.B. Kuan, *Testing input validation in Web applications through automated model recovery.* Journal of Systems and Software, 2008. **81**(2): p. 222-233.

[2] Paolo, T. and R. Filippo, *A 2-Layer Model for the White-Box Testing of Web Applications*, in *Proceedings of the Web Site Evolution, Sixth IEEE International Workshop*. 2004, IEEE Computer Society.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[3] Filippo, R. and T. Paolo, *Analysis and testing of Web applications*, in *Proceedings of the 23rd International Conference on Software Engineering*. 2001, IEEE Computer Society: Toronto, Ontario, Canada.
[4] Di Lucca, G.A. and A.R. Fasolino, *Testing Web-based Applications: The state of the art and future trends.* Information and Software Technology, 2006. **48**(12): p. 1172-1186.
[5] Mansour, N. and M. Houri, *Testing web applications.* Information and Software Technology, 2006. **48**(1): p. 31-42.
[6] Di Lucca, G.A., et al. *Testing Web Applications*. in *Proceedings of International Conference on Software Maintenance*. 2006: IEEE Computer Society Press.
[7] Andrews, A.A., J. Offutt, and R.T. Alexander, *Testing Web applications by modeling with FSMs.* Software and Systems Modeling, 2005. **4**(3): p. 326-345.
[8] Lott, C., A. Jain, and S. Dalal, *Modeling requirements for combinatorial software testing*, in *Proceedings of the 1st international workshop on Advances in model-based testing*. 2005, ACM: St. Louis, Missouri.
[9] Elbaum, S., S. Karre, and G. Rothermel. *Improving web application testing with user session data*. 2003. Portland, OR, United States: Institute of Electrical and Electronics Engineers Computer Society.
[10] Paradkar, A.M., et al. *Automated Functional Conformance Test Generation for Semantic Web Services*. in *Web Services, 2007. ICWS 2007. IEEE International Conference on*. 2007.
[11] Lei, X., X. Baowen, and J. Jixiang, *Testing web applications focusing on their specialties.* SIGSOFT Softw. Eng. Notes, 2005. **30**(1): p. 10.
[12] Hieatt, E. and R. Mee, *Going faster: Testing the Web application.* IEEE Software, 2002. **19**(2): p. 60-65.
[13] Sampath, S., et al. *Web application testing with customized test requirements - An experimental comparison study*. 2006. Raleigh, NC, United States: IEEE Computer Society, Los Alamitos, CA 90720-1314, United States.
[14] Antawan, H. and K. Marc, *Automating Functional Tests Using Selenium*, in *Proceedings of the conference on AGILE 2006*. 2006, IEEE Computer Society.
[15] Huang, C.-H. and H.Y. Chen. *A tool to support automated testing for Web application scenario*. 2007. Taipei, Taiwan: Institute of Electrical and Electronics Engineers Inc., New York, NY 10016-5997, United States.
[16] Cho, Y., W. Lee, and K. Chong, *The Technique of Test Case Design Based on the UML Sequence Diagram for the Development of Web Application*, in *Computational Science and its Applications – ICCSA 2005*. 2005. p. 1 - 10.
[17] Nguyen, H.Q., *Testing Applications on the Web: Test Planning for Internet-Based Systems*. 2000: John Wiley & Sons, Inc.
[18] Nuo, L., et al., *A Framework of Model-Driven Web Application Testing*, in *Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC'06) - Volume 02*. 2006, IEEE Computer Society.
[19] Nguyen, H.Q., B. Johnson, and M. Hackett, *Testing Applications on the Web: Test Planning for Mobile and Internet Based Systems*. 2nd ed. 2003: Wiley Publishing, Inc.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)