Editorial

Emilia Mendes

Computer Science Department
University of Auckland
Private Bag 92019, Auckland, New Zealand
E-mail: emilia@cs.auckland.ac.nz

Abstract: Empirical studies are an essential component of web engineering research and practice. They provide the means necessary to understand, assess, control, and improve web management and development practices and their use of technologies. The results of empirical studies can be used to inform practitioners and researchers alike, and are the means for building a body of knowledge containing sound theories.

This special issue focused on studies where results are based on qualitative and/or quantitative data, which has been used to describe, investigate, and assess relationships between products, processes, and resources. Six studies are presented, four of which are case studies, and two formal experiments.

Keywords: empirical studies; web engineering; case studies; surveys; formal experiments; scientific method.

Reference to this paper should be made as follows: Mendes, E. (2007) 'Empirical web engineering', *Int. J. Web Engineering and Technology*, Vol. 3, No. 3, pp.217–226.

Biographical notes: Dr. Emilia Mendes is a Senior Lecturer in Computer Science at the University of Auckland (New Zealand), where she leads the Web Engineering, Technology and Applications (WETA) research group. She is the principal investigator in the Tukutuku Research Project,[1] aimed at developing and comparing web effort models using industrial web project data, and benchmarking productivity within and across web companies. She has active research interests in web measurement and metrics, and in particular web cost estimation, web size measures, web productivity and quality measurement, and web process improvement. Dr. Mendes is on the Programme Committee of numerous international conferences and workshops, and on the editorial board of the *International Journal of Web Engineering and Technology*, the *Journal of Web Engineering*, and the *Journal of Software Measurement*. She has collaborated with web companies in New Zealand and overseas on web cost estimation and usability measurement. Dr. Mendes worked in the software industry for ten years before obtaining her PhD in Computer Science from the University of Southampton (UK), and moving to Auckland.

1 Introduction

The World Wide Web (WWW) was originally conceived in 1989 as an environment to allow for the sharing of information (*e.g.*, research reports, databases, user manuals) amongst geographically dispersed individuals. The information itself was stored on different servers and was retrieved by means of a single user interface (web browser). The information consisted primarily of text documents inter-linked using a hypertext metaphor[2] (Offutt, 2002).

Since its original inception, the web has changed into an environment employed for the delivery of many different types of applications. Such applications range from small-scale information-dissemination-like applications, typically developed by writers and artists, to large-scale commercial, enterprise-planning and scheduling, collaborative-work applications. The latter are developed by multidisciplinary teams of people with diverse skills and backgrounds using cutting-edge, diverse technologies Gellersen and Gaedke, 1999; Ginige and Murugesan, 2001; Offutt, 2002). Numerous current web applications are fully functional systems that provide business-to-customer and business-to-business e-commerce, and numerous services to numerous users (Offutt, 2002).

Industries such as travel and hospitality, manufacturing, banking, education, and government utilised web-based applications to improve and increase their operations (Ginige and Murugesan, 2001). In addition, the web allows for the development of corporate intranet web applications, for use within the boundaries of their organisations. The remarkable spread of web applications into areas of communication and commerce makes it one of the leading and most important branches of the software industry (Offutt, 2002).

To date the development of industrial web applications has been in general *ad hoc*, resulting in poor-quality applications, which are difficult to maintain (Murugesan and Deshpande, 2002). The main reasons for such problems are unawareness of suitable design and development processes, and poor project management practices (Ginige, 2002).

As the reliance on larger and more complex web applications increases so does the need for using methodologies/standards/best practice guidelines to develop applications that are delivered on time, within budget, have a high level of quality and are easy to maintain (Taylor *et al.*, 2002; Ricca and Tonella, 2001; Lee and Shirani, 2004). To develop such applications web development teams need to use sound methodologies, systematic techniques, quality assurance, rigorous, disciplined and repeatable processes, better tools, and baselines. Web engineering aims to meet such needs (Ginige and Murugesan, 2001). The term 'web engineering' was first published in 1996 in a conference paper by Gellersen *et al.* (1997). Since then this term has been cited in numerous publications, and numerous activities devoted to discussing web engineering have taken place (*e.g.*, journals, workshops, conference tracks, entire conferences).

2 Web engineering and science

Web engineering is described as (Murugesan and Deshpande, 2001):

> "the use of scientific, *engineering*, and management principles and systematic approaches with the aim of successfully developing, deploying and maintaining high quality web-based systems and applications."

Engineering is widely taken as a disciplined application of scientific knowledge for the solution of practical problems. A few definitions taken from dictionaries support that:

> "Engineering is the application of science to the needs of humanity. This is accomplished through knowledge, mathematics, and practical experience applied to the design of useful objects or processes." (Wikipedia, 2004)

> "Engineering is the application of scientific principles to practical ends, as the design, manufacture, and operation of structures and machines." (Houghton Mifflin Company, 1994)

> "The profession of applying scientific principles to the design, construction, and maintenance of engines, cars, machines, *etc.* (mechanical engineering), buildings, bridges, roads, *etc.* (civil engineering), electrical machines and communication systems (electrical engineering), chemical plant and machinery (chemical engineering), or aircraft (aeronautical engineering)." (Harper Collins Publishers, 2000)

In all of the above definitions, the need for 'the application of scientific principles' has been stressed, where scientific principles are the result of applying a scientific process (Goldstein and Goldstein, 1978). A process in this context means that our current understanding, *i.e.*, our theory (hypothesis) of how best to develop, deploy, and maintain high-quality web-based systems and applications, may be modified or replaced as new evidence is found through the accumulation of data and knowledge. This process is illustrated in Figure 1 and described below (Goldstein and Goldstein, 1978):

- Observation

  To observe or read about a phenomenon or set of facts. In most cases the motivation for such observation is to identify cause and effect relationships between observed items, since these entail predictable results. For example, we can observe that an increase in the development of new web pages seems also to increase the corresponding development effort.

- Hypothesis

  To formulate a hypothesis represents an attempt to explain an *Observation*. It is a tentative theory or assumption that is believed to explain the behaviour under investigation (Fenton and Pfleeger, 1997). The items that participate in the *Observation* are represented by variables (*e.g.*, number of new web pages, development effort) and the hypothesis indicates what is expected to happen to these variables (*e.g.*, there is a linear relationship between number of web pages and development effort, showing that as the number of new web pages increases so does the effort to develop these pages). These variables first need to be measured and to do so we need an underlying Measurement Theory.

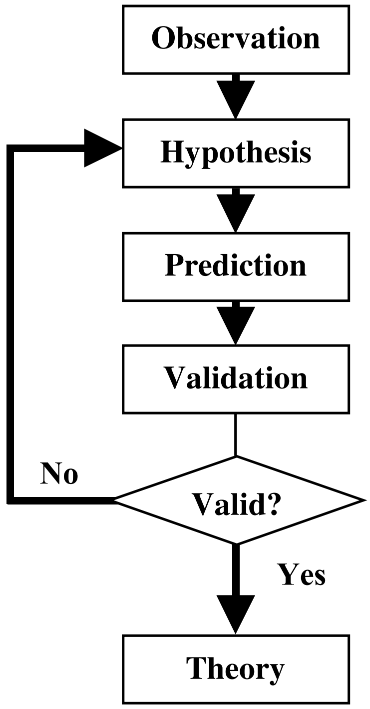
- Prediction

  To predict means to predict results that should be found if the rationale used in the hypothesis formulation is correct (*e.g.*, web applications with a larger number of new web pages will use a larger development effort).

- Validation

  To validate requires experimentation to provide evidence either to support or refute the initial hypothesis. If the evidence refutes the hypothesis then the hypothesis should be revised or replaced. If the evidence is in support of the hypothesis, then many more replications of the experiment need to be carried out in order to build a better understanding of how variables relate to each other and their cause and effect relationships.

Figure 1 The scientific process



The scientific process supports knowledge building, which in turn involves the use of empirical studies to test hypotheses previously proposed, and to assess if current understanding of the discipline is correct. Experimentation in web engineering is therefore essential (Basili, 1996; Basili *et al.*, 1999).

The extent to which scientific principles are applied to developing and maintaining web applications varies among organisations. More mature organisations generally apply these principles to a larger extent than less mature organisations, where maturity reflects an organisation's use of sound development processes and practices (Fenton and Pfleeger, 1997). Some organisations have clearly defined processes that remain unchanged regardless of the people who work on the projects. For such organisations, success is dictated by following a well-defined process, where feedback is constantly obtained using product, process and resource measures. Other organisations have processes that are not so clearly defined (*ad hoc*) and therefore the success of a project is often determined by the expertise of the development team. In such a scenario, product, process, and resource measures are rarely used and each project represents a potential risk that may lead an organisation, if it gets it wrong, to bankruptcy (Pressman, 1998).

The variables used in the formulation of hypotheses represent the attributes of real-world entities that we observe. An entity represents a process, product, or resource. A process is defined as a software-related activity. Examples of processes are web development, web maintenance, web design, web testing, and web project. A product is defined as an artefact, deliverable, or document that results from a process activity. Examples of products are web application, design document, testing scripts, and fault reports. Finally, a resource represents an entity required by a process activity. Examples of resources are web developers, development tools, and programming languages (Fenton and Pfleeger, 1997).

In addition, for each entity's attribute that is to be measured, it is also useful to identify if the attribute is *internal* or *external*. Internal attributes can be measured by examining the product, process, or resource on its own, separate from its behaviour. External attributes can only be measured with respect to how the product, process, or resource relates to its environment (Fenton and Pfleeger, 1997). For example, usability is in general an external attribute since its measurement often depends upon the interaction between user and application. An example of classification of entities is presented in Table 1.

Table 1 Summary characteristics of the three types of empirical investigations

Characteristic	*Survey*	*Case study*	*Formal experiment*
Scale	Research-in-the-large	Research-in-the-typical	Research-in-the-small
Control	No control	Low level of control	High level of control
Replication	No	Low	High
Generalisation	Results representative of sampled population	Only applicable to other projects of similar type and size	Can be generalised within the experimental conditions

The measurement of an entity's attributes generates quantitative descriptions of key processes, products, and resources, enabling us to understand behaviour and result. This understanding lets us select better techniques and tools to control and improve our processes, products, and resources (Pfleeger *et al.*, 1997).

3 Type of empirical studies

Validating a hypothesis or research question encompasses experimentation, which is carried out using an empirical investigation. The three different types of empirical investigation that can be carried out are classified as (Fenton and Pfleeger, 1997):

1 Survey

A retrospective investigation of an activity in order to confirm relationships and outcomes (Fenton and Pfleeger, 1997). It is also known as 'research-in-the-large' as it often samples over large groups of projects. A survey should always be carried out after the activity under focus has occurred (Kitchenham *et al.*, 1995). When performing a survey, a researcher has no control over the situation at hand, *i.e.*, the situation can be documented, compared to other similar situations, but none of the variables being investigated can be manipulated (Fenton and Pfleeger, 1997). Within the scope of software and web engineering, surveys are often used to validate the response of organisations and developers to a new development method, tool, or technique, or to reveal trends or relationships between relevant variables (Fenton and Pfleeger, 1997). For example, a survey can be used to measure the success of changing from Sun's J2EE to Microsoft's ASP.NET throughout an organisation, because it can gather data from numerous projects. The downside of surveys is time. Gathering data can take many months or even years, and the outcome may only be available after several projects have been completed (Kitchenham *et al.*, 1995).

2 Case study

An investigation that examines the trends and relationships using as its basis a typical project within an organisation. It is also known as 'research-in-the-typical' (Kitchenham *et al.*, 1995). A case study can investigate a retrospective event, but this is not the usual trend. A case study is the type of investigation of choice when wishing to examine an event that has not yet occurred and for which there is little or no control over the variables. For example, if an organisation wants to investigate the effect of a programming framework on the quality of the resulting web application but cannot develop the same project using numerous frameworks simultaneously, then the investigative choice is to use a case study. If the quality of the resulting web application is higher than the organisation's quality baseline, it may be due to many different reasons (*e.g.*, chance, or perhaps bias from enthusiastic developers). Even if the programming framework had a legitimate effect on quality, no conclusions outside the boundaries of the case study can be drawn, *i.e.*, the results of a case study cannot be generalised to every possible situation. Had the same application been developed several times, each time using a different programming framework[3] (as in a formal experiment) then it would be possible to have better understanding of the relationship between framework and quality, given that these variables were controlled. A case study samples from the variables, rather than over them. This means that, in relation to the variable programming framework, a value that represents the framework usually used on most projects will be the one chosen (*e.g.*, J2EE). A case study is easier to plan than a formal experiment, but its results are harder to explain and, as previously mentioned, cannot be generalised outside the scope of the study (Kitchenham *et al.*, 1995).

3 Formal experiment

Rigorous and controlled investigation of an event where important variables are identified and manipulated such that their effect on the outcome can be validated (Fenton and Pfleeger, 1997). It is also known as 'research-in-the-small' since it is very difficult to carry out formal experiments in software and web engineering using numerous projects and resources. A formal experiment samples over the variable that is being manipulated, such that all possible variable values are validated, *i.e.*, there is a single case representing each possible situation. If we apply the same example used when explaining case studies above, this means that several projects would be developed, each using a different object-oriented programming language. If one aims to obtain results that are largely applicable across various types of projects and processes, then the choice of investigation is a formal experiment. This type of investigation is most suited to the web engineering research community. However, despite the control that needs to be exerted when planning and running a formal experiment, its results cannot be generalised outside the experimental conditions. For example, if an experiment demonstrates that J2EE improves the quality of e-commerce web applications, one cannot guarantee that J2EE will also improve the quality of educational web applications (Kitchenham *et al.*, 1995).

There are other concrete issues related to using a formal experiment or a case study that may impact the choice of study. It may be feasible to control the variables, but at the expense of a very high cost or high degree of risk. If replication *is* possible, but at a prohibitive cost, then a case study should be used (Fenton and Pfleeger, 1997). A summary of the characteristics of each type of empirical investigation is given in Table 1.

4 Guide to special issue

Empirical studies are an essential component of web engineering research and practice. They provide the means necessary to understand, assess, control, and improve web management and development practices and their use of technologies. The results of empirical studies can be used to inform practitioners and researchers alike, and are the means for building a body of knowledge containing sound theories.

This special issue focused on studies where results are based on qualitative and/or quantitative data, which has been used to describe, investigate, and assess relationships between products, processes, and resources. Six studies are presented, four of which are case studies, and two formal experiments.

In ‘An empirical approach to evaluating web application compliance across diverse client platform configurations’, Eaton and Memon present a case study that assesses the effectiveness of a novel technique to evaluate web application correctness across multiple client configurations. Their technique uses an inductive model that, based on empirical data gathered from fielded applications, predicts how an application is expected to execute. The model evolves automatically based on data from applications that are executing correctly or not, for particular configurations.

In ‘Performance tuning and cost discovery of mobile web-based applications’, Book *et al.* present a case study based on simulation that allow the specification of time and volume predictions for interaction patterns subjected to the frame conditions of a mobile channel. Such predictions can be used to provide project managers with a cost estimate that can be used to decide whether or not it is feasible to add a mobile presentation channel to an existing web application.

Jayalal *et al.* in 'Web site link prediction using a Markov chain model based on multiple time periods' use a case study to investigate the use of a Markov chain model based on an exponentially-smoothed transition probability matrix to estimate links to be selected by users. The model is based on site usage statistics collected over multiple time periods.

Wang *et al.* in 'A user-oriented reliability modelling approach for web systems' describe a series of case studies that compare their user-oriented reliability model to a Markov-based model. The aim of using their model is to prioritise fault removal and therefore help improve reliability by assisting with resource and effort allocation.

In 'Interdependence between technical web accessibility and usability: its influence on web quality models', Arrue *et al.* describe a formal experiment carried out to assess whether more accessible web applications were more usable applications for general users in terms of target found, response time, disorientation and satisfaction. Accessibility was measured using the accessibility metrics proposed in Arrue *et al.* (2005).

In the last paper of the issue, 'Koba4MS: reducing development and maintenance efforts for web-based recommender applications', Felfernig describes a formal experiment that investigated the effects of applying automated testing and debugging techniques to reduce the development and maintenance effort and faults in developing recommender knowledge bases used by recommender applications. Knowledge-based recommender applications aim to providing clients with suggestions (recommendations) that suit very closely their needs.

Acknowledgements

We would like to thank all those who have helped us with this special issue:

Aditya Ghose, Wollongong University

Athula Ginige, University of Western Sydney

Bebo White, Stanford Linear Accelerator Center

David Lowe, University Technology Sydney

Davide Bolchini, University of Lugano

Fabio Vitali, University of Bologna

Geert-Jan Houben, Vrije Universiteit Brussel

Gerald Weber, The University of Auckland

Guilherme H. Travassos, Federal University of Rio de Janeiro

Gustavo Rossi, National University of La Plata

Jose Carlos Maldonado, University of São Paulo

Luciano Baresi, Politecnico de Milano

Luis Antonio Olsina, National University of La Pampa

Marcela Genero, Castilla-La Mancha University

Mario Piattini, Castilla-La Mancha University

Maristella Matera, Politecnico de Milano

Martin Gaedke, University of Karlsruhe

Michalis Vaitis, University of the Aegean

Oscar Pastor, Valencia University of Technology

Piero Fraternali, Politecnico de Milano

San Murugesan, Southern Cross University

Sandro Morasca, Università degli Studi dell'Insubria

Siegfried Reich, Salzburg Research Forschungsgesellschaft

Silvia Abrahao, Valencia University of Technology

Sotiris Christodoulou, University Of Patras

Yogesh Deshpande, University of Western Sydney

References

Arrue, M., Vigo, M. and Abascal, J. (2005) 'Quantitative metrics for web accessibility evaluation', *Proceedings of the First Workshop on Web Measurement and Metrics*.

Basili, V.R. (1996) 'The role of experimentation in software engineering: past, current, and future', *Proceedings of the 18th International Conference on Software Engineering*, 25–30 March, pp.442–449.

Basili, V.R., Shull, F. and Lanubile, F. (1999) 'Building knowledge through families of experiments', *IEEE Transactions on Software Engineering*, July–August, Vol. 25, No. 4, pp.456–473.

Fenton, N.E. and Pfleeger, S.L. (1997) *Software Metrics: A Rigorous and Practical Approach*, 2nd ed., Boston: PWS Publishing Company.

Gellersen, H. and Gaedke, M. (1999) 'Object-oriented web application development', *IEEE Internet Computing*, Vol. 3, No. 1, pp.60–68.

Gellersen, H., Wicke, R. and Gaedke, M. (1997) 'WebComposition: an object-oriented support system for the web engineering lifecycle', *Journal of Computer Networks and ISDN Systems*, Vol. 29, Nos. 8–13, pp.865–1553.

Ginige, A. (2002) 'Workshop on web engineering: web engineering: managing the complexity of web systems development', *Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering*, July, pp.72–729.

Ginige, A. and Murugesan, S. (2001) 'Web engineering: an introduction', *IEEE Multimedia*, January/March, Vol. 8, No. 1, pp.14–18.

Goldstein, M. and Goldstein, I.F. (1978) *How We Know: An Exploration of the Scientific Process*, New York: Plenum Press.

Harper Collins Publishers (2000) *Collins English Dictionary.*

Houghton Mifflin Company (1994) *The American Heritage Concise Dictionary*, 3rd ed.

Kitchenham, B., Pickard, L. and Pfleeger, S.L. (1995) 'Case studies for method and tool evaluation', *IEEE Software*, Vol. 12, No. 4, pp.52–62.

Lee, S.C. and Shirani, A.I. (2004) 'A component based methodology for web application development', *Journal of Systems and Software*, Vol. 71, Nos. 1–2, pp.177–187.

Murugesan, S. and Deshpande, Y. (2001) *Web Engineering, Managing Diversity and Complexity of Web Application Development, Lecture Notes in Computer Science 2016*, Heidelberg: Springer Verlag.

Murugesan, S. and Deshpande, Y. (2002) 'Meeting the challenges of web application development: the web engineering approach', *Proceedings of the 24th International Conference on Software Engineering*, pp.687–688.

Offutt, J. (2002) 'Quality attributes of web software applications', *IEEE Software*, Vol. 19, No. 2, pp.25–32.

Pfleeger, S.L., Jeffery, R., Curtis, B. and Kitchenham, B. (1997) 'Status report on software measurement', *IEEE Software*, March–April, Vol. 14, No. 2, pp.33–43.

Pressman, R.S. (1998) 'Can internet-based applications be engineered?', *IEEE Software*, September–October, Vol. 15, No. 5, pp.104–110.

Ricca, F. and Tonella, P. (2001) 'Analysis and testing of web applications', *Proceedings of the 23rd International Conference on Software Engineering*, pp.25–34.

Taylor, M.J., McWilliam, J., Forsyth, H. and Wade, S. (2002) 'Methodologies and website development: a survey of practice', *Information and Software Technology*, Vol. 44, No. 6, pp.381–391.

Wikipedia (2004) http://en.wikipedia.org/wiki/Main_Page (accessed on 25 October).

Notes

1 http://www.cs.auckland.ac.nz/tukutuku/

2 http://www.zeltser.com/web-history/

3 The values for all other attributes should remain the same (*e.g.*, developers, programming experience, development tools, computing power, and type of application).