
Editorial

Lorna Uden

Faculty of Computing, Engineering and Technology
Staffordshire University
Beaconside, Stafford, ST18 OAD, UK
Fax: +44 1785353497
E-mail: l.uden@staffs.ac.uk

Biographical notes: Professor Lorna Uden teaches computing in the Faculty of Computing, Engineering and Technology at Staffordshire University. Her research interests include technology learning, HCI, activity theory, knowledge management, web engineering, multimedia, e-business and problem-based learning. She has published widely in conferences, journals and chapters of books.

Welcome to this issue of IJWET. I hope you enjoyed reading the special issue. In this issue we have papers from the design of web interfaces to web services. The first paper is 'User interface design practice for the web in New Zealand', by Lal and Overmyer. In their paper, Lal and Overmyer surveyed 62 enterprises in New Zealand, including six major universities with respect to the field of user interface design for the web. The purposes of these studies were:

- to identify the current best practices for user interface design that are being advocated by researchers and practitioners in the user interface design field worldwide for usable websites and web-based applications
- to find out the extent to which these practices are being used to design and develop user interfaces in New Zealand websites and web-based applications
- to find out if the major universities in New Zealand are teaching comparable levels of HCI and usability engineering courses to overseas universities in other developed countries.

Two different surveys were carried out by the authors. The first one was to survey the web design practices that are used by organisations and web development companies to design and develop user interfaces for websites. The second was to survey the major universities in New Zealand to find out what is being taught, and what resources those universities have applied to the HCI subjects that they offer as part of their Information Technology curriculum.

Lal and Overmyer found that the majority of designers and developers do not utilise the best practices covered in the survey, which would enable them to design and develop usable websites and web-based applications. They also found that there is a general lack of awareness in the industry of usability issues among all stakeholders. Another

finding of these authors is that the majority of design and development teams of web development companies and large organisations, including government departments, do not have usability professionals as part of their design teams. The survey of New Zealand universities showed that while the universities offer HCI courses, the majority only offer one or two HCI courses, which are not compulsory for students to study. The survey results are really disappointing in view of the importance of usability in web applications. It is my hope that this paper will draw to the attention other countries that interface design is important and should be taught to students of computer science and information technology, especially where usability is a major industrial concern.

Central to the interface design of an application is the issue of evaluation. There are various methods for evaluating web-based applications. One of the methods is cognitive walkthrough. The second paper, by Ryu, reviews the walkthrough techniques, with the aim of assessing how they can collectively provide average designers with a reasonable suite of usability inspection tools for web-based systems. In his paper, 'Collective web usability analysis: cognitive and activity walkthroughs', Ryu attempts to assess how the walkthrough techniques do this. The paper briefly reviews theoretical foundations of web evaluation and walkthrough methodology and discusses their advantages and disadvantages. He then examines different web usability inspection techniques and the concepts on which each is grounded, combined with case study analyses. Ryu concludes the paper with discussion of the strengths and weaknesses of the methods as they currently exist and suggests improvements for accessing websites. Two walkthrough methods are applied by Ryu to evaluate a website. He exposes the strengths and weaknesses of each walkthrough method. According to Ryu, the walk through approach is to take a hypothetical process, *i.e.*, walking through procedural human information processes, and provide an evaluation at each action stage of how people actually use the system. Although there are many variations of walkthrough-based usability inspection, they typically consist of two phases: a preparatory phase and an analysis phase. In the preparatory phase, the evaluators are given the basic inputs for the walkthrough. In the analysis phase, the evaluators step through each action of every task being analysed. The two methods being used are the Cognitive Walkthrough (CW) and Activity Walkthrough for Web (AWW). Ryu found that AWW can be of value in a very early evaluation process to get extensive insight of contextual usability problems. By comparison, the CW method can relatively explicitly assess how the targeted user population would perform their information seeking task with a detailed design specification. The CW method is more concerned with detecting potential usability problems, moving towards some goal by going through a series of stages or sequence of acts. On the other hand, AWW allows one to assess the intrinsic user's activities under use contexts, so that one can obtain more contextual usability issues, such as contingency that has an impact on a user's covert or overt behaviour. Although the study is useful, more empirical studies would be needed to validate the two methods.

From evaluation we move on to web documents, 'Understanding web documents: finding pagelets transformation using structural patterns' by Ferrydiansyah and Parmanto. These authors claim that dividing web pages into pagelets makes it easier to perform many types of operations on the web page, such as transformation, selective presentation and content retrieval. A pagelet is a small segment of information that is self-contained within a page that has a well-defined topic or functionality. Ferrydiansyah and Parmanto suggested that a pagelet has two attributes, location and functionality. The location of the pagelet is defined by its starting and ending html elements. The functionality of the

pagelet is the type of information contained within the pagelet and what actions are afforded by the pagelet. Ferrydiansyah and Parmanto believe that current work on pagelet detection focuses only on finding the location of the pagelet without regard to its functionality. Under those methods, all pagelets are subjected to the same set of operations regardless of their functionality. In their paper, Ferrydiansyah and Parmanto describe a method to detect the location and functionality of pagelets using html element patterns. These authors believe that the structural patterns of a web page can be used to detect the location and functionality of the pagelet. The method uses structural patterns to find matches to particular pagelet types in a particular web page. A module is created for the detection of each type of pagelet. Modules can be added to give added functionality in detecting additional pagelets. These authors test their technique on multiple popular web pages from the news and e-commerce genres. They found the method adequately recalls various pagelets from the web pages. The biggest problem of this approach is that the correctness of the results depended on the quality of the patterns created. Further research would be useful to identify the most used patterns for a particular pagelet type, as well as more validation for this approach.

So far all the mentioned papers are concerned with desktop computers. The next paper is by Ramakrishnan *et al.* entitled 'Compositional specification and realisation of mixed-initiative web dialogs' and is concerned with mobile computing, such as PDAs, Tablet PCs and 3G phones. Many of these devices have possibilities for multimodal interaction via gestures, speech and handwriting recognition. One such area that is gaining rapid growth in multimodal interaction is web browsing on mobile devices. According to these authors, many factors have contributed to the growing interest in multimodal web interaction. Chief among them is the maturing of commercial speech recognition engines and technologies such as Speech Application Language Tags (SALT) and XHTML plus Voice (X+V). Although technologies such as SALT and X+V enable the amalgamation of speech into browsers, they operate at a lower level of specification than the applications, therefore significantly increasing programming efforts and limiting their impressive power for creating and managing dialogs. To overcome these issues, Ramakrishnan *et al.* have presented Dialog XML – a markup language to specify and reuse expressive web dialogs on mobile devices. By capturing the functional structure of the dialog independent of the modalities used to realise it, Dialog XML facilitates the implementation of web interfaces that integrate hyperlink and speech modes of interaction. It enables the creation of websites that adapt to the needs of users, yet permits the designer fine-grained control over what interactions to support. The framework uses an algorithm based on staging transformation – an approach that represents dialogue by programmes and uses programme transformations to simplify them, based on user input. Further usability studies to evaluate this method would be useful.

From applications of the web we move to services. Paper five of the journal, 'A survey of agent-software engineering for service-oriented computing', is by Leong *et al.* Services play an increasingly important role in software applications today. According to Leong *et al.*, Service-Oriented Computing (SOC) uses services as the fundamental unit to support the development of cost-effective, distributed applications and integrated services in heterogeneous environments. The advantage of SOC is that applications can be easily assembled dynamically from an available pool of cooperative services to enable agile business processes and on-demand computing services. SOC represents an evolution of system development techniques from function-oriented, to data-oriented, to

object-oriented, to interface-oriented, to component-based, to aspect-oriented and to service oriented. The paper surveys and evaluates a number of models and methodologies that attempt to tie in two domains of software engineering; namely agent-oriented analysis and design and service-oriented analysis and design.

These authors believe that present conventional approaches to engineering SOA systems are not able to address the complexities of open and dynamic environments such as those in an extended virtual enterprise or inter-organisational workflows. They then argued for an Agent-Oriented Software Engineering (AOSE) to address the shortcomings. Leong *et al.* have surveyed several AOSE methods for SOC. They suggested that AOSE methodologies such as A4, AOMG, MAGE, Primetheos, Gaia, Tripos, OPEN, MESSAGE, MaSE and Pass can be extended to cater for specific SOC issues. This would enable SOC to handle dynamic and complex service overflow, and provide mechanisms to support automatic services that exhibit self-adaptation, self-configuration, self-healing and self-protection. Although this is a survey paper, we feel it would be of interest to readers who would like to know more about SOA. Since AOSE has the potential to tackle many of the issues facing SOC in complex systems, this calls for new research in the development of agent methodologies for service-oriented software engineering.

Continuing with the subject of services, we come to our last paper on 'Semantic model-driven development of web service architectures', by Pahl. The author argues that building service-based architectures has become a major area of interest. Model-Driven Development (MDD) is an approach for the development of software systems that has gained wide support. It focuses on the notion of abstraction and automation. Pahl proposes an ontology-based engineering methodology for semantic model-driven composition and transformation of web service architecture. He has applied MDD philosophy to web service development. According to Pahl, the overall aim of his approach is to demonstrate the internal coherence of an overarching methodology that can integrate a number of specific techniques and to illustrate the benefits of semantic modelling and MDD. Three frameworks – methodology, ontology and platform – provide the pillars for his model-driven service engineering approach. Pahl argues that ontologies are useful because they provide semantic service descriptions in a sharable format. Ontologies also allow reasoning about service properties. Both are essential to support discovery and composition in service-oriented architecture. Pahl believes that by using this approach he has demonstrated the suitability of ontologies for service architecture. While the approach is useful, further research will be needed to validate the benefits of this method.