

---

## Editorial

---

### Lorna Uden

FCET,  
Staffordshire University,  
The Octagon, Beaconside,  
Stafford, ST18 0AD, UK  
Email: L.uden@staffs.ac.uk

**Biographical notes:** Lorna Uden is Professor Emeritus of IT Systems in the Faculty of Computing, Engineering and Technology at Staffordshire University. Her research interests include technology learning, HCI, big data, mobile learning, activity theory, knowledge management, web engineering, multimedia, e-business, service science and innovation, semantic web, software as a service (SaaS), internet of things and problem-based learning.

---

Welcome to Vol. 10, No. 2 of the *IJWET*. The first paper is, ‘Improving the modularity of web-application models with aspects’ by Gefei Zhang and Matthias Hölzl. Zhang and Hölzl argue that modern web applications often contain features, such as landmarks, access control, adaptation, or rich UI, that are difficult to model modularly with existing model-driven web engineering (MDWE) approaches. These authors show how high-level aspects (HiLA), an aspect-oriented extension for UML state machines, can represent these features as aspects.

HiLA is an aspect-oriented extension of UML state machines. It provides a new language construct to separately model parts of the system behaviour, and thus enhances the modularity of the models. HiLA can be used to separate different concerns in modelling web applications as follows: first, the modeller starts with a very simple state machine (the base machine) to model the basic navigation structure. Typical hard-to-modularise features of web applications, such as landmarks, access control and adaptation, are then modelled separately in aspects. This way, the basic navigation structure, as well as other, business-process features, is kept simple and easy to read, hence the model is simpler and each concern is ‘packaged’ as a single unit that can either be understood in isolation, or at least with a small set of possible interactions.

HiLA models achieve a high grade of separation of concerns and satisfy the ‘Don’t Repeat Yourself’ (DRY) guideline. Particularly interesting features of HiLA are:

- 1 its capability to serve as a substrate for domain-specific modelling languages
- 2 declarative combination of features to precisely control the desired overall behaviour of an application and ensure that different features do not interfere with each other
- 3 support for automatic and formal verification of (existence or absence of) interference between features.

HiLA can be used on top of a number of state-machine-based MDWE approaches to enhance the modularity of models.

The authors of this paper showed some modularity problems exhibited by UML state machines when they are used in the context of MDWE, and how to use HiLA to mitigate them. One of the highlights of the approach is the automatic and formal verification of existence and absence of interferences between the aspects, and the simple, declarative definition of feature combination. In future work, they plan to improve the scalability and usability of the validation process, better support the definition of conflict resolution strategies based on verification results, and implement code generation out of HiLA aspects.

The second paper is, 'Efficient methodologies to overcome the effects of hanging pages in search engine optimisation' by P. Ravi Kumar, Ashutosh Kumar Singh and Anand Mohan. According to these authors, 'hanging page' is one of the important challenges in search engine optimisation (SEO). Hanging pages can affect the SEO process especially for the link structure-based search engine ranking algorithms. In this paper, they first explore different types of hanging pages in a website and their problems using a mathematical model. Second, they provide information about SEO and its challenges. Third, they explore how hanging pages can affect SEO through examples and experiments. Finally, they propose methods to handle hanging pages (naturally forming hanging pages) and suggest methods to improve websites rank through experiments.

This paper has introduced hanging pages in a mathematical way and exposed the problem of hanging pages in SEO. It also proposed methods to handle hanging pages in SEO, discussed the challenges of SEO and explored the different on-site and off-site ranking factors for SEO. The authors have also listed out a couple of black hat techniques which can potentially reduce the ranking of a website and even remove the entry from the index of a search engine. They conducted experiments to show the effect of hanging pages on SEO. Further experiments are necessary to validate the effectiveness of the research.

The third paper is, 'Predicting web page performance level based on web page characteristics' by Junzan Zhou, Yun Zhang, Bo Zhou and Shanping Li. The authors of this paper argue that large-scale web applications, based on complex architecture, spanning multiple data centres and content distribution networks, may face the performance issue of high page-load time. Web service providers can improve the performance at maintenance phase, but have few tools to predict the performance at early phases. In this paper, they present an approach to predict web page performance based on classification methods. Classification models are trained using known labels and they can predict user experience for a new page. The authors applied various classification techniques as predictors.

They present a web page performance prediction framework that can be used at early phases of software development life cycle to avoid the negative impact caused by web page performance problems. They evaluated their solution on over 2,200 websites' landing pages. Experiments show that their framework can provide helpful predictions. Among the classification algorithms investigated, random forest achieves the best performance in terms of accuracy. More data are needed to verify the approach.

The final paper is, 'Efficient run-time verification of web service composition' by Yik-Shiung Yau and Fang-Fang Chua. These authors argue that current web service composition (WSC) is a technology which realises SOA and has proven to be a successful approach to govern the continuous, dynamically and unbounded business requirements of the open world business domain. Although WSC offers significant benefits throughout the software development life-cycle (SDLC), WSC has its drawbacks

and limitations. The major drawback is where the behaviour and the quality of web services consumed could deviate from what is anticipated, particularly during run-time. Although the current WSC verification approaches are being developed to detect violation, there are no underlying efforts mentioned to optimise the verification process and no guidelines are provided to implement those approaches in an optimised way.

According to these authors, WSC-RTVSystem is a highly responsive verification system that allows users to specify and verify the requirements of the designed WSC system during run time. WSC-RTVSystem is flexible enough to allow users to specify their own verification properties according to the system's needs. WSC-RTVSystem is centred on the service intermediary. The current design of WSC-RTVSystem is tightly coupled to the self-developed service intermediary, which is not compliant to service intermediary standard. Being able to support only the generic type parameters is also one of its limitations. Instead of binding directly to service providers, the service requestors are required to bind with the service intermediary with service providers' service descriptions in order to utilise WSC-RTVSystem, which introduces additional effort.

To further improve the response time of WSC-RTVSystem, it would be useful to assign a dedicated local service for verification purposes (asynchronous processing) with handshaking with the system where the forwarded messages are generated and held until the verification is completed. By performing this, the verification process and message generation will be done in parallel, hence improving the response time which is the time taken for both processes to finish. Further improvement is to extend the verification properties in WSC-RTVSystem to support non-functional properties such as timing and reliability. Non-functionality properties allow the quality attribute of a system to be measured and analysed. This would make WSC-RTVSystem to be more comprehensive in verification of composite web services.