Overview of the Articles in This Issue

This issue contains three research papers and an experience report. In the first paper titled “Modeling Variant User Interfaces for Web-Based Software Product Lines,” Lee discusses Software Product Line (SPL) as a software engineering paradigm for modeling variations in user interfaces during the software development life cycle. This paradigm leverages the reuse of core assets, often including a specification or design model having variation points of functionality. Those specific functionalities are termed “variant features” in the product line. A software product within a product line often has specific functionalities that are not common to all other products within the product line. Such a product derived from the core assets has specialized realizations of these variant features (variation points). Therefore, the SPL paradigm involves the modeling of such variant features. The problem is that when developing several such products one of the most effort-intensive modules or function points is the User Interface (UI). However, a barrier to a solution is that little work in SPL has investigated or addressed the modeling of variant features specific to UIs. This lack of work is serious because a great deal of development effort is consumed in that development phase focusing upon customization of user interface.

The research reported in this paper researches and develops a UML extension — Web User Interface Modeling Language (WUIML) — to mitigate barriers to solving the SPL UI development productivity problem. WUIML defines elements for modeling variant features specific to user interfaces for Web-based SPLs. The model elements in WUIML extend from the metaclass and BasicActivity of the UML2.0 metamodel. WUIML integrates the modeling of variant features specific to user interfaces to UML. For example, in a Web-based patient registration software product line, member products targeting British users may use a British date format in the user interface, while member products targeting United States users may use a United States date format in the user interface. Thus, this is a variant feature for this product line. WUIML defines a model element, XOR, to represent such exclusive-or conditions in a product line UI model. The research results report that WUIML can reduce SPL engineers’ efforts needed in UI development.

To validate the WUIML research outcome, case studies are being conducted, and one of these case studies’ results is reported here. The results of this case study...
indicate that modeling UIs for Web-based SPLs using WUML is more effective and efficient than using standard UML.

The next paper “A Subspace Clustering Framework for Research Group Collaboration” by Agarwal, Haque, Liu, and Parsons presents a novice approach for recommending research articles to a small, user population based on their recent querying and browsing search history. The core of the system is a scalable Subspace Clustering Base Algorithm (SCuBA) that operates on the sparse, high-dimensional data collected in this domain. Both synthetic and benchmarked data sets are used to evaluate the recommendation system.

The results of the comparative analysis demonstrated that the proposed algorithm performed better than other domain collaborative filtering approaches when dealing with this specific-domain user community. For further research, the authors suggest a hybrid approach to recommendation systems and ranking schemes to take care of the limitations of specific systems and improve performance. The editors hope that future research will examine the feasibility of developing algorithms for a multi-stage collaborative recommender filtering system. The first two stages of the system are based on binary relationships. The second stage is based on tree structure of the preferences. Furthermore, the editors believe that a useful extension to this study will be to deploy the system on the Web to be used by virtual small research groups all over the world, with an appropriate user interface. As advocated by the IJITWE, researchers supporting a multi-disciplinary integrated approach to research combining members from different academic disciplines and from academia and professional worlds will appreciate the assistance provided by such a system developed in this paper. Also, software design artifacts as generated by the model in Lee’s paper could be used to design different display characteristics for each member in virtual groups.

In the paper “Engineering Wireless Mobile Applications,” Mahmoud and Maamar discuss the technical challenges in engineering wireless mobile applications. These applications are specifically developed for users who are on the move, and hence rely heavily on handheld wireless devices to conduct their day-to-day operations. The paper emphasizes that wireless mobile applications have their own set of obstacles that place an additional burden on application developers who are put on the front line of achieving the promise of businesses and service providers of delivering Internet content to wireless handheld devices. These devices tend to have less memory, less powerful CPUs, different input devices, and smaller displays. Because of these obstacles, methods developed for conventional systems are not optimal for wireless applications. In particular, wireless application development does not always fit into the development model originated to cope with conventional, large software systems. In this paper, the authors present a systematic approach to wireless application development, and discuss practical guidelines for testing wireless applications. This paper plays well with current and future trends in enterprise computing. In a related recent event, the Wireless and Mobile Summit 2005 held in April, Gartner predicted that mobile and wireless systems will control the visage of corporate computing in 2005 and beyond (http://www.gartner.com/2_events/conferences/ra8.jsp). There could be a link between this paper and Lee’s paper on modeling the User Interface for SPL. Software design artifacts generated by the model presented in Lee’s paper could be reused to deploy different displaying requirements for mobile devices.

The final paper titled “Experience Report: A Component-Based Data Manage-
ment and Knowledge Discovery Framework for Aviation Studies” by Blake, Singh, Williams, Norman, and Sliva presents experiences in developing a framework for Component-based Knowledge Discovery from Databases (C-KDD) with a major application to the aviation industry. The following are the major features of this framework: support for heterogeneous sources of data (unstructured and structured), delimited text format, and nonstandard text format. The authors discuss three stages of the C-KDD where the domain experts can manipulate data extraction/loading directives, data transformation instructions, and knowledge discovery instructions. The latter is the most innovative one where domain expert can provide scenario hints through a user interface to control the knowledge discovery mechanism.

The system constructed in this paper was loaded with data from seven different types of sources and tested on two airports under convective weather conditions during the summer months. Two major experiments were conducted to test the framework: checking flight rules against visibility and ceiling (cloud-level), and measuring the correlation between wind speed and airport performance. The system was implemented on Java platform, and the learning components were implemented by integrating the Waikato Environment for Knowledge Analysis (WEKA) data mining application and toolkit into the framework.

Dr. Ghazi Alkhatib is an assistant professor of software engineering, College of Computer Science and Information Technology, Applied Science University, Amman, Jordan. In 1984, he obtained his Doctor of Business Administration from Mississippi State University in information systems and minors in computer science and accounting. Since then, he has been engaged in teaching, consulting, training, and research in the area of computer information systems in the U.S. and Gulf countries. In addition to his research interests in databases and systems analysis and design, he has published several articles and presented many papers in regional and international conferences on software processes, knowledge management, e-business, Web services and agent software, workflow, and portal/grid computing integration with Web services.

Dr. David Rine is a professor and founding chair of computer science with the School of Information Technology and Engineering, George Mason University. He is also part of the group of co-founders of information technology. He received a PhD from The University of Iowa in 1970. He has developed and directed the teaching of several Web-based distance education courses and has directed a variety of MS and PhD students in Web engineering. During his 40-year career in computing, he has published nearly 300 papers in the areas of computer science, engineering, information technology and information systems, computer applications, computational science, science and engineering education, systems engineering and software engineering. Dr. Rine is internationally known for his work in science and engineering education, having accumulated many years of experience in directing curriculum, large scale software, computational science and systems projects. He has received numerous awards from computer science societies and associations, including the IEEE Centennial Award, the
IEEE Computer Society Pioneer Award, the IEEE Computer Society Meritorious Service Award, the IEEE Computer Society Special Award, and the IEEE Computer Society 50th Anniversary Golden Core Award. He has also been a multiple-time recipient of the IEEE Computer Society Honor Roll (and Distinguished Service) Award and the IEEE Computer Society Certificate of Appreciation Award. The Army, Air Force, Navy, NATO, National Science Foundation, IEEE, NASA, IBM, USDA and many other industrial organizations have funded Dr. Rine’s scientific research and development work.