



**Ecole Supérieure des Sciences et Techniques de Tunis**  
Unité de Recherche en Technologies de l'Information et de la  
Communication

## Invitation

Dans le cadre des activités du chapitre tunisien de l'ACM, l'Unité de Recherche [UTIC](http://www.utic.rnu.tn) de l'Ecole Supérieure des Sciences et Techniques de Tunis (ESSTT) a le plaisir de vous inviter à une présentation, qui sera donnée par **Prof Sandeep K. Shukla de l'Université de Virginie (Virginia Polytechnic)**.

La présentation porte sur le **développement orienté modèle de logiciels embarqués**. Cette présentation aura lieu le **mercredi 02 juin 2010 à 10h à l'amphi A2 à l'ESSTT**.

### **Model Driven Embedded Software Generation: A Generative Approach to Safety**



#### **Sandeep K. Shukla**

Associate Professor of Computer Engineering  
Deputy Director of Center for Embedded Systems for Critical Applications  
Director of FERMAT Lab  
Virginia Polytechnic and State University, USA

#### **Abstract**

Avionics, automotive, power plant control, and many other safety-critical embedded systems require safe, predictable, and statically analyzable software. Moreover, as the complexity of these applications mounts, performance and safety both become increasingly important. This increasing performance requirement drives the current market trend of multi-core processors (single chip multiprocessors) in the desktop market. However, recently embedded processors have started to surface with multiple homogeneous or heterogeneous cores. Multi-threaded or concurrent applications seem to be the best way to exploit these available parallel processing resources.

Those with any experience with multi-threaded programming would admit that design and implementation of multi-threaded programs is extremely difficult and prone to subtle concurrency and synchronization bugs, even without the use of advanced techniques such as speculative hreading, or wait-free synchronization etc are. The inherent synchronization and dependency issues and the possible non-determinism are difficult to resolve without

extremely skilled programmers, and possibly with the help of extensive static analysis. Static analysis and/or formal verification of large concurrent applications are again capacity limited by today's state-of-the-art tools and techniques. Nevertheless, given the importance of safety in the target application domains, one has to produce absolutely correct code which is deterministic or predictable, and no non-deterministic execution behavior should lead to disastrous consequences.

Correct-by-construction multi-threaded program generation is therefore our methodology of choice. For this, we need a formal specification language with well defined semantics and proper characterizations as to when it is safe to generate guaranteed deterministic code. We have chosen Polychronous or multi-rate specification language borrowing from the French synchronous programming languages. Synchronous languages have so far proven useful for generating sequential (single-threaded) code for safety-critical applications. A particular characterization of polychronous specifications called the 'endochrony' is a sufficient condition for correct sequential code generation. Therefore, one could generate multiple sequential threads separately from 'endochronous' specification fragments, and compose them by generating the synchronization glue code. Unfortunately, 'endochrony' is not compositional, and therefore, the synchronization code generation becomes non-trivial. We show that a particular generalization of 'endochrony' called the 'weak endochrony' is sufficient for directly generating multi-threaded code from such specifications. Moreover, 'weak endochrony' is compositional and hence provides us with a modular code generation technique from polychronous specification.

In this talk, first, we elaborate on multi-rate specification formalism Polychrony. Then we explain the difficulties of deterministic and semantics preserving code generation from such specifications. Then we discuss endochrony, inadequacy of which leads to the weak endochrony concept, and how this provides a sufficient condition for safe multi-threaded code generation. Finally, we discuss future directions in our work on deterministic multi-threaded code generation for safety-critical applications.

## **Biography**

Sandeep K. Shukla is an associate professor of computer engineering at Virginia Tech. He is also a founder and deputy director of the center for embedded systems for critical applications (CESCA), and director of his research lab FERMAT. Sandeep was awarded the Presidential Early Career Award for Science at Engineering (PECASE) award for his research in design automation for embedded systems design, which in particular focuses on system level design languages, formal methods, formal specification languages, probabilistic modeling and model checking, dynamic power management, application of stochastic models and model analysis tools for defect-tolerant system design, and reliability measurement of defect-tolerant systems. Sandeep has published more than 125 articles in journals, books and conference proceedings. Sandeep co-authored three research monographs, and four edited volumes. Sandeep has been elected as a College of Engineering Faculty Fellow at Virginia Tech. In 2008 Sandeep was awarded the Alexander Humboldt Foundation's Bessel Award. Sandeep also chaired a number of international conferences and workshops, edited a number of special issues for various journals, and are on the editorial board of IEEE Design & Test, IEEE Transactions on Computer, and IEEE Embedded Systems Letters. Sandeep is a senior member of IEEE and ACM. He is also an IEEE Computer Society Distinguished visitor, and an ACM distinguished speaker.