Lynne and William Frankel Center for Computer Science
Department of Computer Science, Ben-Gurion University of the Negev
Beer-Sheva, Israel 84105. Tel: (972-8) 6428032
www.cs.bgu.ac.il/~frankel
Saal Auditorium 202, Alon Building for Hi-Tech 37

Seminar Series Supported by Jeffrey and Holly Ullman

Software Modeling Day
in Honor of Prof. Mira Balaban

June 7, 2017

Registration: https://goo.gl/forms/T91mDnW6VqpHKfzy1

9:30 Gathering and Registration

9:45 Greetings
Jihad El-Sana and Arnon Sturm, Ben-Gurion University

10:00 Employing the tool USE (UML-based Specification Environment) for Model-Based Engineering
Martin Gogolla, University of Bremen

MBE (Model-Based Engineering) proposes to develop software by taking advantage of models, in contrast to traditional code-centric development approaches. If models play a central role in development, model properties must be formulated and checked early on the modeling level, not late on the implementation level. We discuss how to validate and verify model properties in the context of modeling languages like the UML (Unified Modeling Language) combined with textual restrictions formulated in the OCL (Object Constraint Language).

10:40 SYNTECH: Synthesis Technologies for Reactive Systems Software Engineers
Shahar Maoz, Tel-Aviv University

Reactive synthesis is an automated procedure to obtain a correct-by-construction reactive system from a given specification. Examples of reactive systems include the software controllers of robotic systems. Despite recent advancements on the theory and algorithms of reactive synthesis, many challenges remain in bringing reactive synthesis technologies to the hands of software engineers. The SYNTECH project is about bridging this gap. In this talk I will present observations, case studies, and initial results from the project.

11:20 Coffee break

11:40 Can Domain Analysis be Automated? Levels of Automation in Domain Modeling
Iris Reinhartz-Berger, Haifa University

Domain models are representations of areas of knowledge that use common concepts for describing phenomena, requirements, capabilities, and solutions. Variability analysis plays an important role in creating comprehensive domain models automatically or semi-automatically.
In this talk I will review the challenges in automating domain modeling and discuss how variability analysis can contribute. Specifically, I will concentrate on an ontological and semantic approach which examines software behaviors and uses different artifacts (including requirements, test cases, and code) to analyze the variability, as well as commonality, in the domain of discourse.

12:20 Languages for Programming: From Punched Cards to Wise Computing
David Harel, Weizmann Institute

After very briefly discussing the main milestones in the development of programming languages, the talk will provide some details about more recent means for programming complex reactive systems. These will include visual formalisms, the use of natural language, and a futuristic "wise computing" approach.

13:00 Lunch break

14:00 Umple: The benefits of text-diagram duality in modeling
Tim Lethbridge, University of Ottawa

The concrete syntax of software models has classically been either diagrammatic (e.g. UML), or textual (e.g. OCL). Different benefits related to understandability, ease of editing, versioning and error detection arise from each form, so both are beneficial. We describe text-diagram duality in Umple, which allows 100% textual modeling, while integrating and generating classic textual programming languages. Umple also generates many types of diagrams with changes reflected as text is edited. Changes made in diagrams also update the textual form.

14:40 Is the programmer's job obsolete?
Dov Dori, Technion

In this talk, I will present Object-Process Programming (OPP) – a visual programming language, which naturally extends ISO 19450 OPM. As part of the ongoing OPCloud Project, we are integrating OPP into the OPM conceptual modeling environment. When completed, a systems or software engineer, like today's mechanical engineer, will ideally be able not just model to a system whose software components must be programmed by programmers (software drafters), but deliver the software components of a functioning system, which can be shown to work properly by simulation.

15:20 Coffee break

15:35 Analysis of UML Class Models - Correctness and Quality
Azzam Maraee, Ben-Gurion University

Model-Based Software Engineering (MBSE) puts models at the heart of the software development process. The Class Model is central within the UML. Analysis of UML class models involves questions like (1) Are the constraints in a model contradictory - consistency; (2) Can the constraints be satisfied by a finite number of objects - Finite satisfiability; (3) Are all constraints in a model necessary - quality and simplification.
In this talk we introduce the Class Model, characterize the above analysis tasks, and summarize known results and our methods for (1) Efficient detection and cause identification of finite satisfiability problems; (2) Removing redundant constraints; and (3) Introducing pattern specification.

16:15 Closing Remarks
Mira Balaban, Ben-Gurion University

16:45 End of Software Modeling Day

Registration: https://goo.gl/forms/T91mDnW6VqpHKfzy1