Neumann János Számítógép-tudományi Társaság
Megőrizni a múlt értékeit, alkalmazkodni a jelenhez, befolyásolni a jövőt.

Eötvös Loránd Tudományegyetem
Informatikai Kar

NJSZT Újklub (IEEE CS-NJSZT Fórum) presents:

Software Technology Forum

16 November 2016 15:00

ELTE IK EIT Digital Magyar Nemzeti Társult Csomópont, Co-Location Center

1117 Budapest, Bogdánfy utca 10/a.

15:00

Marinus Jacobus Plasmeijer
University of Nijmegen

Task Oriented Programming

The internet has the advantage that it is accessible from almost any platform (Windows, Mac, Android) and any device (PC's, laptops, tablets, smart phones). The collaboration of groups of people over the internet is growing rapidly (Facebook, Uber, WhatsApp) involving the coordination and synchronisation of all kinds of distributed running applications, browsers and apps, located on all kinds of devices. However, it is not so easy to develop such applications, because it becomes technically very complicated to handle all situations, for all people, on all devices involved, in a correct manner. In this talk we present a new style of programming, called Task Oriented Programming (TOP). TOP is specially designed to support the development of distributed multi-user, multi-platform, web-based applications. Special about TOP is that it allows programmers to focus on the tasks the application has to support, without worrying about the technical details needed for the realization.

16:15

Melinda Tóth
Eötvös Loránd University

Application-specific static software analysis

Understanding and maintaining the source code of industrial-scale software product is hard and time-consuming, and it is getting more difficult when the software implements parallel/concurrent/distributed computations and behaviours. Static source code analysis tools support program comprehension through detecting dependencies and relations among small (like functions and processes) or large (modules and components) software elements. For Erlang, which is a dynamically typed language, only an approximation of the real dependencies can be calculated statically. However it is possible to improve these analyses by adding application specific information, such as the most frequently used behaviours. In this talk we present a static source code analysis tool, RefactorErl, and its extensions to support process relation analysis for Erlang including the client-server behaviour implementation.

Organizers:

Registration is required: www.inf.elte.hu