# 3rd International Workshop on Developing Tools as Plug-Ins (TOPI 2013)

Michael Barnett
Microsoft Research
Redmond, WA, USA

Martin Nordio
ETH
Zurich, Switzerland

Judith Bishop
Microsoft Research
Redmond, WA, USA

Karin K. Breitman
PUC-Rio
Brazil

Diego Garbervetsky
Universidad de Buenos Aires
Argentina

***Abstract*—TOPI (http://se.inf.ethz.ch/events/topi2013/) is a workshop started in 2011 to address research questions involving *plug-ins*: software components designed and written to execute within an extensible platform. Most such software components are tools meant to be used within a development environment for constructing software. Other environments are middle-ware platforms and web browsers. Research on plug-ins encompasses the characteristics that differentiate them from other types of software, their interactions with each other, and the platforms they extend.**

## I. Overview

Our knowledge as to how to solve software engineering problems is increasingly being encapsulated in tools. These tools are at their best when they operate in a pre-existing development environment. This approach allows integration with existing elements such as compilers, debuggers, profilers, and visualizers as well as numerous other development and, often, runtime tools.

Building tools as ‘plug-ins’ can be challenging. How do they interact with the core environment? How do they interact with one another, especially since each developer may choose a different set of plug-ins. How can we share tools across different and future core development environments? This workshop is intended for all those interested in developing tools as plug-ins for IDEs, middle-wares and browsers.

## II. Papers

The papers this year included both authors of plug-ins as well as the designers of extensible platforms. They cover the medium and long term challenges of developing tools as plug-ins as well as research contributions identifying recent successful tools as plug-ins, characteristics of good plug-ins and reports of the main difficulties in implementing plug-ins in current platforms.

There were twelve submissions, of which eight were accepted as long papers and two as short papers.

## III. Keynote

The keynote address is by Layla Driscoll, Microsoft Corporation: **Project Roslyn: Exposing the C# and VB compilers code analysis**. Until now, the VB and C# compilers have been used as black boxes. You put text in, and you get out a binary file. In our long-lead project, codename Roslyn, we are changing that dynamic by building an API that exposes our compilers analysis engines. In this information-packed session, we’ll cover the goals of the Roslyn project and go in-depth into the powerful set of APIs for building ”code aware” tools and extensions. We’ll also cover some of the major Roslyn APIs in depth. Armed with this knowledge, we’ll see how easy it is to leverage the APIs to analyze and generate C# and VB source code or add C# scripting support to an application.

Layla Driscoll is the Program Manager Lead for Managed Languages and the PM for Visual F#. Her team is responsible for the existing compilers and the new C# and VB Roslyn compilers. In her time at Microsoft she has also worked on the .NET Common Language Runtime, Office 365 and on tablets. She also had odd jobs like being an electrical technician on a proton accelerator. She received her Baccalaureate in Computer Science from New Mexico Institute of Mining and Technology and her Technology Management MBA from the University of Washington. When building software there are both technical and human challenges. Layla finds both fascinating.

## IV. Acknowledgements

We would especially like to thank Microsoft Research for its generous support.

978-1-4673-3076-3/13/$31.00 © 2013 IEEE