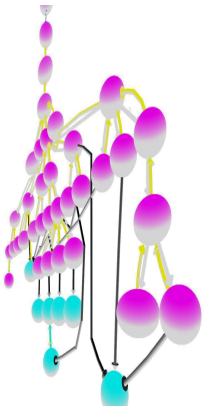
D-TEC – *"DSL Technology for Exascale Computing"*

Dan Quinlan (Lead PI)
Lawrence Livermore National Laboratory (Lead Lab)
Co-PIs and Institutions: Saman Amarasinghe, Armando Solar-Lezama, Adam Chlipala, Srinivas Devadas, Una-May O'Reilly, Nir Shavit, Youssef Marzouk @ MIT; John Mellor-Crummey & Vivek Sarkar @ Rice University; Vijay Saraswat & David Grove @ IBM Watson; P. Sadayappan & Atanas Rountev @ Ohio State University; Ras Bodik @ University of California at Berkeley; Craig Rasmussen @ University of Oregon; Phil Colella @ Lawrence Berkeley National Laboratory; Scott Baden @ University of California at San Diego



D-TEC will define approaches to construct Domain Specific Languages (DSLs). Different sorts of DSLs will be targets for this work; DSLs constructed from existing languages without syntax extension (embedded DSLs) and DSLs with syntax extension to existing base languages. Within each case, we will use common general-purpose languages (e.g. C, C++, Fortran) as base languages from which to build DSLs. A focus of this work is to lower the barriers to the compiler construction and program analysis required to support DSLs in the future. Our work will make future Exascale software development easier by defining how to bury the details (e.g. power management, resiliency, and exotic node architectures) that we anticipate future software to be forced to address behind well-defined abstractions. Our team includes expertise in runtime-systems, compiler research, static and dynamic optimization, tools, and HPC applications; gathered from multiple universities and national laboratories. Our work serves the requirements of the DOE Exascale program and is work in collaboration with other Exascale Co-Design Centers and X-Stack projects.

Domain Specific Languages (DSLs) use automated code transformation to shift the burden of portable performance from the application programmer to the compiler by expressing programmer intent at a high level of abstraction and encapsulating human domain-specific optimization knowledge. DSLs have proven valuable for scientific computing, but widespread adoption is hindered by major implementation challenges. To address them we are extending ROSE with many new features for building DSLs. The Rosebud framework will be a cohesive interface through which a DSL developer uses these features, providing a single comprehensive system that supports all aspects of defining and implementing a DSL. Rosebud encapsulates expert domain knowledge into modular DSL plugins that can be developed independently and then distributed and combined in HPC applications. These plugins implement a form of generalized DSL encompassing "embedded" languages (no custom syntax), custom-syntax extensions to a standard host language, and full standalone custom languages. Rosebud consists of an API defining the form of a textual DSL description plus two main tools: a Generator used to compile DSL descriptions into plugins and a Translator used to develop HPC applications. Given a set of previously developed plugins for the DSLs used in an application, the Translator applies plugins' domain knowledge to transform high level DSL code into optimized host language code; a vendor-supplied compiler then performs low-level optimization and code generation.

* 925-423-2668, dquinlan@llnl.gov

Other supporting work for DSLs within D-TEC will: define a parameterized abstract machine to support analysis and optimization decisions (Rice University), leverage the X-10 runtime system (IBM) to support cross machine optimizations, build on existing polyhedral optimization research for selection of on-node optimizing transformations (Ohio State University), leverage compiler research within the ROSE project (LLNL, Rice University, University of Oregon), define a novel sketching technology to support proof-based correctness checking of DLS lowering transformations (MIT), develop new fault tolerance research for sensitivity analysis and compiler-drive resiliency transformations (MIT and LLNL), define new multi-level tool support for DSL abstractions (Rice University, UCSD, and LLNL), and identify critical abstractions in target Exascale co-design applications (LBL). This research work will address how to make DSLs economical and attractive as a technique to provide higher levels of abstraction for the development of complex HPC application on future exotic Exascale hardware. More information is available at the D-TEC web site: http://dtec-xstack.org.

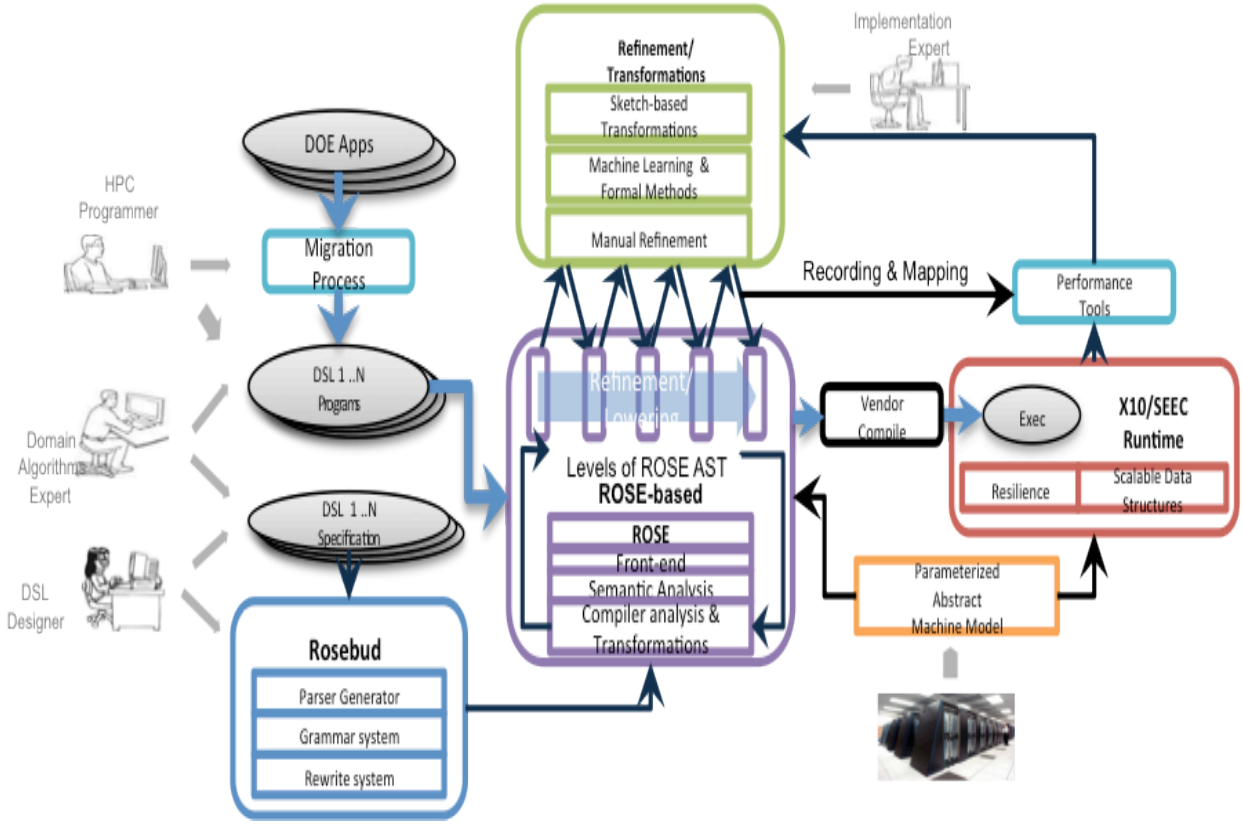


Figure 1: Workflow for development and use of DSLs for Exascale software development.

D-TEC will be releasing software to support the development of future exascale software. D-TEC software will be publically available open source, under a BSD license, with documentation, and mailing list support, and more, at: www.dtec-xstack.org.

For further information on this subject contact:
Name: Dan Quinlan
Org: Lawrence Livermore National Laboratory
Email: dquinlan@llnl.gov
Phone: 925-423-2668