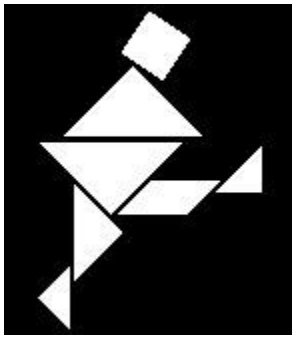

http://www.gpce.org

GPCE 2013

October 27-28, 2013 ● Indianapolis, IN, USA
collocated with SPLASH/OOSPLA and SLE

12th International Conference on Generative Programming: Concepts & Experiences

GPCE is a venue for researchers and practitioners interested in techniques that use program generation, domain-specific languages, and component deployment to increase programmer productivity, improve software quality, and shorten the time-to-market of software products. In addition to exploring cutting-edge techniques of generative software, our goal is to foster further cross-fertilization between the software engineering and the programming languages research communities.

Submissions
Research papers 10 pages (sigplan style)
Tool demos and short papers 4 pages

Submission of papers: June 14, 2013
Author notification: August 22, 2013

Sponsored by
ACM SIGPLAN

Generative and component approaches and domain-specific abstractions are revolutionizing software development just as automation and componentization revolutionized manufacturing. Raising the level of abstraction in software specification has been a fundamental goal of the computing community for several decades. Key technologies for automating program development and lifting the abstraction level closer to the problem domain are **Generative Programming** for program synthesis, **Domain-Specific Languages** (DSLs) for compact problem-oriented programming notations, and corresponding **Implementation Technologies** aiming at modularity, correctness, reuse, and evolution. As the field matures **Applications** and **Empirical Results** are of increasing importance.

GPCE seeks contributions on all topics related to generative software and its properties. Topics of interest include, but are not limited to:

Generative software

- **Domain-specific languages** (language extension, language embedding, language design, language theory, language workbenches, interpreters, compilers)
- **Product lines** (domain engineering, feature-oriented and aspect-oriented programming, preprocessors, feature interactions)
- **Metaprogramming** (reflection, staging, partial evaluation)
- **Program synthesis**
- **Implementation techniques** and tool support (components, plug-ins, libraries, metaprogramming, macros, templates, generic programming, run-time code generation, model-driven development, composition tools)

Properties of generative software

- **Correctness** of generators and generated code (analysis, testing, formal methods, domain-specific error messages, safety, security)
- **Reuse** and **evolution**
- **Modularity**, separation of concerns, understandability, and maintainability
- **Performance** engineering, nonfunctional properties (program optimization and parallelization, GPGPUs, multicore, footprint, metrics)
- **Application** areas and engineering practice (distributed systems, middleware, embedded systems, patterns, development methods)

Empirical evaluations

- **Empirical evaluations** of all topics above (user studies, substantial case studies, controlled experiments, surveys, rigorous measurements)

We particularly welcome papers that address some of the **key challenges** in the field, such as, synthesizing code from declarative specifications ● supporting extensible languages and language embedding ● ensuring correctness and other nonfunctional properties of generated code ● proving generators correct ● improving error reporting with domain-specific error messages ● reasoning about generators ● handling variability-induced complexity in product lines ● providing efficient interpreters and execution languages ● human factors in developing and maintaining generators

This year, GPCE seriously encourages submissions about empirical evaluations of generative software with special considerations during reviewing. See website for details.