

GISMO: Genetic Improvement of Software for Multiple Objectives

Principal Investigator

Prof. Mark Harman, CREST centre, UCL

Named Research Fellow

Dr. William B. Langdon, CREST centre, UCL

Academic Collaborator

Marc Shoenauer, INRIA, France
Andrea Arcuri, Simula Labs, Norway

Industrial Collaborators

Motorola Labs, Basingstoke, UK.
IBM
NVidea

Keywords: Software Engineering, Software Testing, Optimizing Non-Functional Requirements, Genetic Programming.

What is Genetic Improvement Programming (GIP)?

GIP evolves replacement software components that maximize achievement of multiple objectives, while retaining the interfaces between the components so-evolved and the surrounding system. The GISMO project will develop theory, algorithms and techniques for GIP as a way to automatically optimize multiple Software Engineering objectives such as maximal throughput, fastest response time and most reliable performance, while minimizing power consumption, faults, memory use, compiled code size, peak disk usage and disk transfers.

The term ‘component’ should be interpreted in its widest context. In this proposal, we use the term to refer to any piece of code that can be identified as a sub part of the overall program or system with well-defined interface to the encasing system. For example, we include functions, files, modules and procedures, for which interfaces are defined by parameters and shared global variables. We also include smaller segments of contiguous code that perform a coherent well-defined task or set of tasks, for which the interface is captured by the defined and referenced variables of the segment of code. What is important is that these pieces of code can be replaced by an evolved component that preserves their functionality and interface, while maximizing achievement of challenging new multiple objectives.

The Problem Addressed by GISMO?

The emergent computing application paradigms require systems that are not only correct, but also which optimize for many different competing non-functional requirements. Increasingly, we need to adapt existing systems to cater for operating environments with challenging non-functional properties. For instance, the migration from stand-alone systems to large-scale distributed systems brings with it a need for optimization of non-functional properties such as response time and throughput. The increasing prevalence of smaller hand-held systems such as communications devices, raises the importance of non-functional properties such as power consumption and memory use.

Managing any one of these non-functional objectives is a challenge, but managing several at once is a daunting prospect, even for the most skilled and experience developer. The multiple objectives that have to be optimized are often conflicting. For instance, one can often trade speed of execution for compiled code size. Humans cannot be expected to optimally balance such competing constraints and may miss potentially valuable solutions.

The GISMO Solution?

The GISMO solution rests on two core observations:

1. There exists a wealth of relatively well-tested code upon which organisations already rely.
2. Evolutionary computation has proved able to balance many different competing and potentially conflicting criteria.

We therefore seek to use evolutionary computation, not to evolve entire systems, but to replace components within existing systems with evolved replacements. The goal of the evolution will be to optimise for a new set of non-functional properties. These non-functional properties will be mapped into fitness functions that will guide the evolution. The research challenge is to develop techniques that evolve components that balance these objectives, in a scalable way, while producing code that is useful and acceptable to the developer. The GISMO project will address the scalability issue using parallel computation. It will address the human acceptability issue using interactive evolution.

Why the Project Will be Highly Transformative

Hitherto, genetic programming has proved to be good at evolving small code fragments for a single objective, while evolutionary computation has proved effective at solving multiple objective problems. The GISMO approach to scalability, human acceptance and multiple objectives are all entirely novel for genetic programming. If the proposal is even partly successful in its goal of automatically finding GIP-evolved component replacements, this would be a major breakthrough. It would significantly increase our ability to migrate systems to challenging new operating environments and, simultaneously, dramatically reduce the cost of doing so.

Is it feasible?

The PI and Named Research Fellow, Bill Langdon, have demonstrated the feasibility of the GIP approach. In our initial work we were able to port a critical component of Unix `gzip` utility to a CUDA platform. The GISMO project will employ Dr. Langdon as a Named Research Fellow for four years, in order to develop this new approach to software development.

What are the GISMO Objectives?

To achieve its aims, the GISMO project will:

1. Develop a theory of Genetic Interface Programming (GIP).
2. Develop techniques for parallel GIP computation for scalability and interactive GIP evolution for human involvement.
3. Develop new algorithms for achieving single and multiple objective GIP.
4. Evaluate qualitatively and quantitatively using benchmarks and real world systems from the industrial partners.