The GISMOE Architecture☞

Yue Jia[1], Mark Harman[2], Bill Langdon[3],
1(CREST, University College London, yue.jia@ucl.ac.uk)
2(CREST, University College London, mark.harman@cs.ucl.ac.uk)
3(CREST, University College London, w.langdon@cs.ucl.ac.uk)

Abstract: The GISMOE research agenda is concerned with optimising programs for non-functional properties such as speed, size, throughput, power consumption and bandwidth can be demanding. GISMOE sets out a vision for a new kind of software development environment inspired by recent results from Search Based Software Engineering (SBSE). Details of the GISMOE research agenda are provided in the extended keynote paper for the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE 2012). This talk overview is a brief introduction to the approach and a description of the talk about the GISMOE agenda at the 2nd Chinese SBSE workshop in Dalian, 8th and 9th June 2013.

Key words: SBSE; GISMOE

1 Introduction

In this talk overview (which is extracted from our full paper at ASE 2012 [1]) we provide a glimpse of how a combination of advances in software test data generation, genetic programming and multi objective optimisation can be combined to realise a vision of automated program synthesis. We can think of this approach as a new kind of compiler; one that produces multiple versions of a source program on a conceptual pareto program surface. We briefly describe our proposed architecture and the principal features of this proposed development environment, which we call GISMOE: Genetic Improvement of Software for Multiple Objective Exploration. This is work in progress at UCL CREST, the Centre for Research on Evolution Search and Testing, University College London, UK.

2 GISMOE

The GISMOE approach focuses on pre-existing programs as the source of functional descriptions in this paper. GISMOE uses automated testing to assess the degree to which the functional properties are preserved and to measure the achievement of non-functional requirements. Measurements of both functional and non-functional requirements are used to provide the fitness functions that guide a multi objective optimisation and visualisation process.

☞ A more detailed explanation of the GISMOE approach and architecture can be found in the ASE keynote paper [1].

3 Architecture

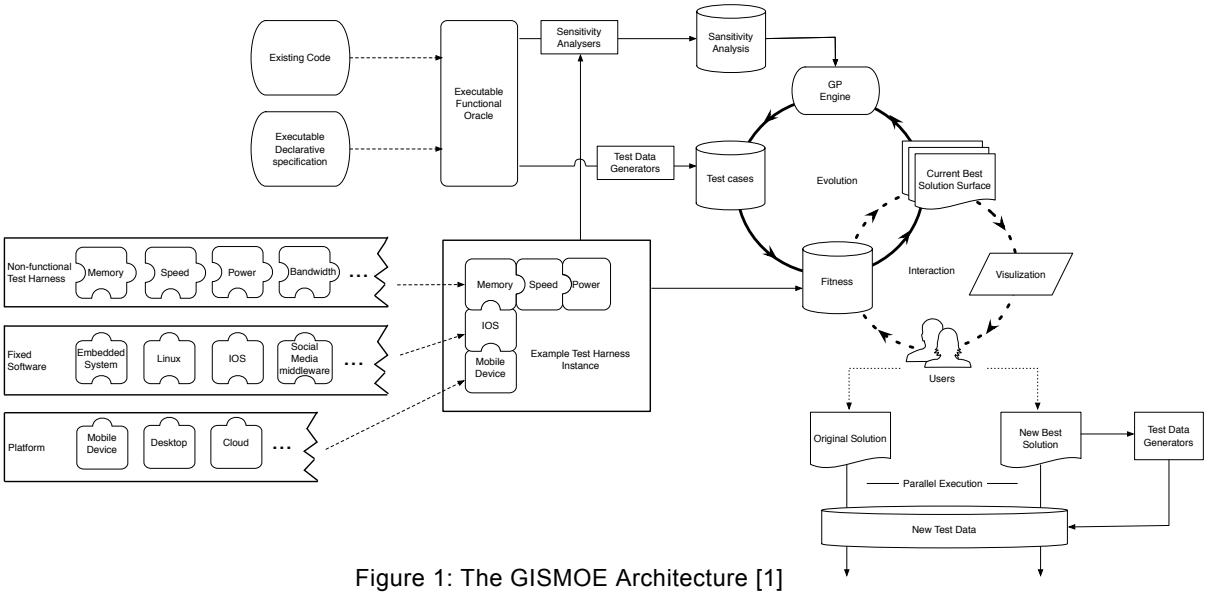

Figure 1: The GISMOE Architecture [1]

Our proposed GISMOE architecture is depicted in Figure 1. Dotted lines indicate choices of components. Solid lines indicate data flows. The 'fixed software' is the code that is not changed by GP. The choice of platform and fixed software constitute the environment in which the non-functional properties are evaluated. Together, these non-functional evaluators constitute the collections of multiple objective fitness values for which GISMOE seeks Pareto optimal solutions (thereby creating the Pareto program surface).

The software developer can choose to combine arbitrary sets of non-functional-property fitness functions with different environments to make a harness that provides fitness data as an input to the GP engine and sensitivity analysis. Test cases can come from any test data generation technique, for which there are many options (already available) that can be used as a component to GISMOE.

Sensitivity information can be pre-computed before the GP improvement process commences. The sensitivity of the program to each non-functional property is computed using the non-functional evaluation harness. This process requires no knowledge of the functional test cases, since it seeks to identify those parts of the program that are non-functionally sensitive, irrespective of functional properties.

The GP engine uses the test cases and the non-functional harness to compute its fitness. In the conformant version of GISMOE, the only solutions considered are those that pass all test cases (these programs are functionally conformant and the approach is conformant to the view that 'correctness is paramount').

4 References

[1] Harman, M.; Langdon, W.B.; Jia, Y.; White, D.R.; Arcuri, A.; Clark, J.A., "The GISMOE challenge: constructing the pareto program surface using genetic programming to find better programs (keynote paper)," Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE 2012), pages 1-14, 3rd-7th September, 2012.