The Dreaming Machine
Fixes Bugs Overnight

Sæmundur Ó Haraldsson*
John R. Woodward*
Alexander E.I. Brownlee*
Kristin Siggeirsdottir°

*University of Stirling
°Janus Rehabilitation
Bug Fixing
What is the problem?

Unbelievably boring
What is the problem?
What is the solution?
What is the solution?
What is the solution?
Why is this human competitive?

- (E) The result is equal to or better than the most recent human-created solution to a long-standing problem for which there has been a succession of increasingly better human-created solutions.

- (G) The result solves a problem of indisputable difficulty in its field.

---

**Fixing Bugs in Your Sleep: How Genetic Improvement Became an Overnight Success**

Sæmundur O. Haraldsson
University of Stirling
Stirling, United Kingdom FKN 4LA
saharald@stir.ac.uk

Alexander J.I. Brownlee
University of Stirling
Stirling, United Kingdom FKN 4LA
debrown@stir.ac.uk

John R. Woolward
University of Stirling
Stirling, United Kingdom FKN 4LA
jronwo@stir.ac.uk

Kristian Sägzerodottir
Janus Rehabilitation Centre
Reykjavík, Iceland
ks@garmin.com

**ABSTRACT**

We present a bespoke live system in commercial use with self-improving capability. During daytime human teams provide an overview and control for many specialists to simultaneously schedule and observe the rehabilitation process for multiple clients. However, in the evening, after the last user logs out, it starts a self-analysis based on the day’s recorded interactions. It generates test data from the recorded interactions for Genetic Improvement to fix any recorded bugs that have raised exceptions. The system has already been tested for over 18 months and has in this time identified, located, and fixed 22 bugs. No other bugs have been identified by other methods during that time. It demonstrates the effectiveness of such live data generation and the utility of Genetic Improvement.

---

**The Use of Automatic Test Data Generation for Genetic Improvement in a Live System**

Sæmundur O. Haraldsson* John R. Woolward and Alexander J.I. Brownlee*

*Department of Computing Science and Mathematics
University of Stirling
Stirling, Scotland
Email: info@cs.stir.ac.uk

**1 INTRODUCTION**

Genetic Improvement (GI) [9] is a growing area within Search Based Software Engineering (SBSE) [11, 12, 13, 31] which uses computational search methods to improve existing software. Despite its growth within academic research the practical usage of GI has not yet followed. Like many SBSE applications, the software industry needs an introduction period for new ideas where they can be tried in outcomes and use these ideas in costeffective solutions. GI is in the ideal position to shorten that period for the latter as it presents a considerable cost decrease for the software life cycle’s often most expensive part: maintenance [13, 34]. There are examples of software improvement (GI being used) and publicly available (GI) which is impressive considering how recent GI is as a technology.

---

**Abstract**—In this paper we present a bespoke live system in commercial use that has been implemented with self-improving properties. During business hours it provides overview and control for many specialists to simultaneously schedule and observe the rehabilitation process for multiple clients. However, in the evening, after the last user logs out, it starts a self-analysis based on the day’s recorded interactions and the self-improving process. It uses Search Based Software Testing (SBST) techniques to generate test data for Genetic Improvement (GI) to try to fix any bugs if exceptions have been recorded. The system has already been under testing for 4 months and demonstrates the effectiveness of simple test data generation and the power of GI for improving live code.

**Keywords**—Search Based Software Engineering; Test data generation; Bug fixing; Real world applications

---

**II. RELATED WORK**

The product is a bespoke program for a vocational rehabilitation centre, developed and maintained by Janus Rehabilitation Centre in Reykjavik, Iceland.

The remainder of the paper is structured as follows. Section II lists some related work and inspirations. Section III details what the system does during business hours and how it keeps records for later generating test data. Section IV explains how the daily data is used to generate and utilise test data and Section V summarises the current data gathered since the launch of JM. Section VI gives an overview of what future direction we are currently contemplating.
Why is this human competitive? (E)
Why is this human competitive? (E)
Why is this human competitive? (E)

22 bugs fixed Overnight!
Why is this human competitive? (G)
Why is this human competitive? (G)
Why is this human competitive? (G)
Why is this human competitive? (G)
Why is this human competitive? (G)
Why is this human competitive? (G)
Why is this best?
Why is this best?

- First ever example
Why is this best?

- First ever example
- All bugs fixed overnight
Why is this best?

- First ever example
- All bugs fixed overnight
- Widespread impact
Why is this best?

- First ever example
- All bugs fixed overnight
- Widespread impact
- Applicable to any software
Why is this best?

- First ever example
- All bugs fixed overnight
- Widespread impact
- Applicable to any software
- Approved by domain experts and practitioners alike