

---

# Open BEAGLE: A New C++ Evolutionary Computation Framework

---

Christian Gagné and Marc Parizeau  
Laboratoire de Vision et Systèmes Numériques (LVSN)  
Université Laval, Québec (QC), Canada, G1K 7P4.  
E-mail: {cgagne,parizeau}@gel.ulaval.ca

## Abstract

This poster introduces a new C++ Evolutionary Computation (EC) framework named *Open BEAGLE*. This framework is freely available on the projet's Web page at <http://www.gel.ulaval.ca/~beagle>.

Open BEAGLE<sup>1</sup> is a C++ framework for doing almost any kind of EC. Its architecture follows the principles of Object Oriented (OO) programming, where some abstractions are represented by loosely coupled objects and where it is common and easy to reuse code. Open BEAGLE has a three level architecture as illustrated by Figure 1. The OO foundations are the basis of this architecture, as an object oriented extension of C++, inspired by *design patterns* (Gamma et al., 1994; Lenaerts and Manderick, 1998). It offers basic functionalities like smart pointers and garbage collection, object allocators, standard containers, and XML I/O streams. The generic EC framework implements basic mechanisms and structures for designing versatile specialized Evolutionary Algorithms (EA). It is summarized in Figure 2. It comprises three main components: a vivarium, an evolution system, and an evolver. The vivarium is a container for demes of generic individuals. The individuals themselves are specified by an abstract genotype. This genotype can be instantiated to any relevant structure (in Figure 2, it is shown as a bit string, but this is just an example). Individuals and demes can also be specialized if needed.

In the evolution system, the *context* contains the state of the genetic engine, such as the current deme and generation number. This concept is similar to the execution context of a computer. The *register* is a central repository for all evolution parameters. The evolving process itself is governed by an *evolver* that defines sequences of operations, contained in operator sets,

---

<sup>1</sup>The recursive acronym BEAGLE means *the Beagle Engine is an Advanced Genetic Learning Environment*.

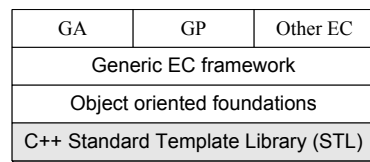


Figure 1: Open BEAGLE Framework Architecture.

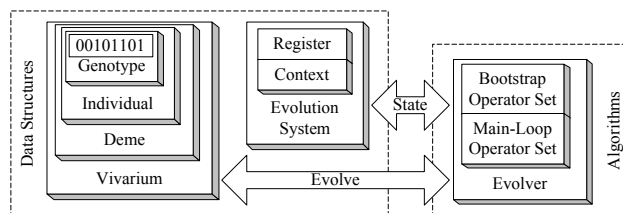


Figure 2: Generic EC Framework Architecture

that are iteratively applied to demes. The evolver applies the *bootstrap operator set* to initialize the first generation, and the *main-loop operator set* to the subsequent generations. For common EA, standard operators have been defined, such as common selection schemes, crossovers, mutations, statistics calculation, and evolution checkpoint backup.

The specialized frameworks are at the top level of the architecture. Currently, only classical genetic algorithms and genetic programming frameworks have been implemented. But we are looking forward to external contributions by interested researchers for other specialized EA. Extensive documentation is available on the Open BEAGLE's Web page.

## References

- Gamma, E., Helm, R., Johnson, R., and Vlissides, J.: 1994, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley
- Lenaerts, T. and Manderick, B.: 1998, Building a genetic programming framework: The added-value of design patterns, in *Proceedings of EuroGP '98*, Vol. 1391 of *LNCS*, pp 196–208, Springer-Verlag