Table of Contents

The Operations Guide **2**

Jake Good

Hardware and Software Requirements........ 2

Installation of System 2

Backup and Recovery 3

Programming Guide **4**

Ryan Dixon

Data Descriptions for files and records:........ 4

File Format: 4

File Format or Structure Changes: 4

The Operations Guide

Jake Good

Hardware and Software Requirements

Installation and usage of the framework was designed to be simple. Though there are no specific hardware requirements to use our framework, other than enough hard drive space to save your evolved objects onto, there are software requirements that may carry certain other restrictions. For instance, the Java runtime that is necessary for the framework to run requires a Pentium 166MHZ processor with at least 32MB of physical RAM. The other software requirements are that there must be a graphical user interface system installed. The window manager must be compatible with the Java graphics libraries; as the genetic algorithms application requires the use of GUI components. Any actual editing of code for a specific implementation must be done with a text editor and compiled separately.

The framework itself is implemented in Java and requires the Java 1.3 runtime system to be installed. This runtime can be downloaded for free from http://java.sun.com . If the specific implementation of an individual and fitness function requires compilation, then the system must also have the Java SDK installed. This can also be downloaded for free from http://java.sun.com. Installation help may be found on their website, http://servlet.java.sun.com/help/installation.

Installation of System

Installing the framework and application onto your system for use requires a few steps. Once you have received the installation CDROM or a current copy of the framework, it will contain a *.jar file that contains all of the class files necessary to run the framework, an application, and a demo of the system.

The first step in installation would be to create a directory on your hard drive, for example C:\GAFramework or /home/GAFramework. After creating the directory, you can copy the folder structure included on the CDROM or a current copy of the framework into the directory of your choice.

The next step requires a decision to be made on the user's end. If you are not comfortable or do not have access to change the PATH variable in your operating system, then you will need to have a copy of the *.jar file in the root directory of your project that uses the framework. Otherwise, you can leave the *.jar file within the install directory and update the PATH variable to point to that directory. Refer to help files within your current operating system for help on changing the PATH variable.

Backup and Recovery

Backup and recovery of the system will only be necessary when there is a corrupt class file or source code changes. There is a built in mechanism for retrieving object data from the framework. Once there have been generations that have evolved, there is a save function where the objects can be saved off to disk. If the system becomes corrupt, then backing up the system would involve copying the *.jar file from the original install directory into a directory of your choice. Recovery of the system becomes as easy as deleting the current directory and following the installation instructions. One could also recopy the *.jar file back into the install directory.

Programming Guide

Ryan Dixon

Data Descriptions for files and records:

File formats for this framework are currently based on the Java 1.3 default serialization specification. All files saved via `GAObjectMonitor` require two conditions to be met. First, the `GAIndividual`, wrapped by the `IndividualHolder` class, *must* be serializable. Secondly, there must be enough free disk space available for the selected generations to be saved. File sizes will vary (proportionally) depending on the size of each object and the number of objects selected before saving.

File Format:

All objects saved via the `GAObjectMonitor` are saved using an `ObjectStream`. All files are written out as `IndividualHolder` objects, containing both the `GAIndividual` and it's associated `fitness`. Java's default versioning mechanism is used.

NOTE: Exporting and importing with different versions of the same class via the ObjectStream will fail! Be sure to use identical classes when exporting and importing object data! (See "System's Error Messages" section)

For more information regarding Java's object stream format and implementation, please refer to the following website: http://java.sun.com/.

File Format or Structure Changes:

At the time of the first release of this product, no file structure changes have been recorded.

Current structural details are fully documented at Sun Microsystems' website: http://java.sun.com/.