

GUIDE Installation Manual

Luis Da Costa and Marc Schoenauer

March 28, 2008

This is the system's installation manual for GUIDE. GUIDE is a layer that provides a common interface to evolutionary libraries to all users, practitioners or not.

to rewrite

In our work we treat the general problem of bringing the domain of research called Evolutionary Computation (EC) closer to the general user, and therefore making its objects of study (Evolutionary Algorithms, EA) more easily understandable. The ultimate goal of this line of research is to be able to offer EAs as part of a general problem-solver toolkit.

Specifically, we argue for the necessity of a common ground, both in theory and communication, but also in practice; a main unifying idea from EC practitioners must reflect on a single view for newcomers to the field. We argue, in our paper [SC], that one of the main grouping ideas of the field should be the search for EAs that are as independent of the representation as possible. This idea is supported by work published in the literature and by our own theoretical and practical work. On that respect, we present our implementation of a software layer/product called GUIDE, who embodies the ideas of the field that we present here.

We think that GUIDE is a good proposal of a solution for the problem of making EAs widely-known. As a further proof of concept, GUIDE is being used on a personal-research level for some courses on EC (it is available under an Open Source Licence agreement at <http://gforge.inria.fr/projects/guide/>) and also on its first major development, as part of the European Project EvoTest (*Evolutionary Testing for Complex Systems*, <http://evotest.iti.upv.es/>). EvoTest is a multidisciplinary European funded project whose objective is to use evolutionary adaptive techniques in order to find solutions to the problems of testing software systems and deal with its complexity.

Contents

1	How to use this guide	5
1.1	Purpose and Audience	5
1.2	Scope	5
1.3	Suggested Reading Order	5
1.4	Notational Conventions	5
1.5	Associated Documents and Electronic Resources	6
2	Installation Steps	7
A	Description of Settings File	11

Chapter 1

How to use this guide

1.1 Purpose and Audience

This document is targeted to any user, experimented or not, that is looking to explore the functionalities of GUIDE.

1.2 Scope

This manual explains how to install GUIDE in Windows-, UNIX- or MacOS-based machines. It is supposed that a C++ compiler is installed, and a Java compiler is also installed.

1.3 Suggested Reading Order

The main purpose of this Document is fulfilled by the explanations given in Chap. 2. More detail about certain concepts can be found on the the Appendixes; however, the information contained in them is not needed for the successful installation of the product.

1.4 Notational Conventions

The following conventions are used in this guide:

- **Java.** The term "Java" refers to any general version of the Java compiler. When functionality differs between compilers, the specific terms will be used (for example, *Java 1.5* for Java v. 1.5).
- **C++.** The term "C++" refers to any general version of the C++ compiler.
- **Eclipse** refers to the working environment of the same name. IDEs for different languages are found here: <http://www.eclipse.org/home/categories/languages.php>

1.5 Associated Documents and Electronic Resources

GUIDE's Javadoc Documentation is in <http://www.lri.fr/~ldacosta/guideDoc/> and was developed in the TAO team (<http://tao.lri.fr>); now it is hosted <http://gforge.inria.fr/projects/guide/>. There are 2 main papers, to this date, associated with it: the first was published in 2003, under the title *GUIDE: Unifying Evolutionary Engines through a Graphical User Interface*, was written by Pierre Collet and Marc Schoenauer[CS03]. The second is currently being evaluated for publication in the Journal of Artificial Intelligence Tools, and is identified in the bibliography as [SC].

GUIDE generates code for two evolutionary libraries: EO and ECJ. EO's website is <http://eodev.sourceforge.net/> and a main communication describing its functioning was published by Keizer *et al.* and corresponds to reference [KMRS02]. ECJ's website is <http://www.cs.gmu.edu/~eclab/projects/ecj/>

GUIDE is actively used as product in the Evotest European Project <http://evotest.itl.upv.es/>

Chapter 2

Installation Steps

The following steps have to be carried out in order to have GUIDE installed and ready to run in your environment. These steps are to be followed, no matter in what kind of operating system (Windows, UNIX-like or MacOS) the environment is to be used.

1. **Download Evolutionary Libraries.** First you will need to install the evolutionary libraries¹ for which GUIDE generates (compilable and executable) code. You have the choice to install all of them, or just install the one that you think will be the most used by you².
 - (a) **EO.** Install the *Evolving Objects* Library. Go to <http://sourceforge.net/projects/eodev/>. Download it, unpack it and follow the instructions for installation.
 - (b) **ECJ.** Install the *Evolutionary Computation in java* Library. Go to <http://www.cs.gmu.edu/~eclab/projects/ecj/>. Download it, unpack it and follow the instructions for installation.
2. **Download GUIDE Specific Files.** Choose (and create, if needed) a folder in your local machine where you will store all the configuration needs for the execution. Let's identify this folder as <CONFDIR>. Go to GUIDE's webpage (<http://gforge.inria.fr/projects/guide/>), Tab *Fichiers* ("Files" in French). Download:

¹These libraries are used by GUIDE to generate the programs solving a specific problem.

²At this moment (April 2008), the Evotest project is heavily using and testing GUIDE in combination with EO, so the support for this library is stronger than for the others.

- (a) the latest version of the .jar file present (under *Runtime GUIDE* it is called **GUIDE.jar**). Copy it in your local disk.
- (b) **algorithmSchema.xsd** Copy it in <CONFDIR>. **algorithmSchema.xsd** is the grammar file (XSD) for the XML that is the definition of an EA.
- (c) **settingsExample.xml** Copy it in <CONFDIR>. This is an XML file with an example of how to set up GUIDE.
- (d) **typesPlugAndPlayDir.tar** or **typesPlugAndPlayDir.zip** Copy it in <CONFDIR>. This is the packaged tree of pre-defined types used by GUIDE. The detailed structure of this tree of types is described in App. A.
- (e) **extraCode.tar** or **extraCode.zip** Copy it in <CONFDIR>. This is the directory where you are going to indicate the libraries (and any "extra code") needed for the application, along with the location where the main function of the application is.

3. Configure your copy.

- (a) **Unpack the pre-defined types directory.** On <CONFDIR>, do

```
tar -xvf typesPlugAndPlayDir.tar
```

This will create a directory called <CONFDIR> /typesPlugAndPlayDir

- (b) **Unpack the extra code directory.** On <CONFDIR>, do

```
unzip extraCode.zip
```

(if you downloaded the .zip file) or

```
tar -xvf extraCode.tar
```

(if you downloaded the .tar file)

This will create a directory called <CONFDIR> /extraCode, having 3 sub-directories inside: extraCode/includes, extraCode/implement and extraCode/controller

- (c) Copy **settingsExample.xml** into your local, modifiable file (let's call it **mySettings.xml**) Open *mySettings.xml* with your favourite text editor. You will see a file, described in detail in App. A.

- (d) Modify the line identified by the tag **PLUGANDPLAYTYPESDIR**, in order to obtain a line that reads:

```
<PLUGANDPLAYTYPESDIR>
<CONFDIR> /typesPlugAndPlayDir
</PLUGANDPLAYTYPESDIR>
```

- (e) Specify a location for the GUIDE's Log file has to be specified. The Log file is useful in order to track the behaviour of the environment on different times of a specific run. For this, you need to modify the line identified by the tag **LOGFILENAME**, in order to obtain a line that reads:

```
<LOGFILENAME> name of the log file </LOGFILENAME>
```

For example, if you would like the Log file to be at `/tmp/kernelLog.log`, you should write:

```
<LOGFILENAME> /tmp/kernelLog.log </LOGFILENAME>
```

- (f) Extra code:

- i. Look for the line tagged **INCLUDES**. You will see inner tags **DIR**. Modify the first of these in order to obtain a line that reads:

```
<DIR>
<CONFDIR> /includes
</DIR>
```

Delete the others (for now³); you should then obtain this configuration:

```
<INCLUDES> <DIR>
<CONFDIR> /includes
</DIR> </INCLUDES>
```

- ii. Look for the line tagged **EXTRACODE**. You will see inner tags **DIR**. Modify the first of these in order to obtain a line that reads:

```
<DIR>
<CONFDIR> /extraCode
</DIR>
```

³More details about this can be obtained in the Appendix

Delete the others (for now⁴); you should then obtain this configuration:

```
<EXTRACODE> <DIR>  
<CONFDIR> /extraCode  
</DIR> </EXTRACODE>
```

- iii. Modify the line identified by the tag **CONTROLLERDIR**, in order to obtain a line that reads:

```
<CONTROLLERDIR>  
<CONFDIR> /controller  
</CONTROLLERDIR>
```

Leave the rest of the file as it is. But if you want more details, just to App. A

4. Done!! Now GUIDE's GUI will run when writing

```
java -jar GUIDE.jar -settings <CONFDIR> /mySettings.xml
```

in your command line.

⁴More details about this can be obtained in the Appendix

Appendix A

Description of Settings File

The configuration file for GUIDE looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<GUIDEKERNELSETTINGS>
  <PLUGANDPLAYTYPESDIR>/Users/ldacosta/guideTypes</PLUGANDPLAYTYPESDIR>
  <LOGFILENAME>/tmp/kernelLog.log</LOGFILENAME>
  <LOGLEVEL>INFO</LOGLEVEL>
  <CURRENTEXPERIMENT />
  <XSD>
    <ISONLINE>TRUE</ISONLINE>
    <Online>http://www.lri.fr/~ldacosta/evoxml/algorithmSchema.xsd</Online>
    <Offline>/Users/ldacosta/algorithmSchema.xsd</Offline>
  </XSD>
  <CODEGENERATION>
    <TargetLibrary>EO</TargetLibrary>
    <VERBOSE>TRUE</VERBOSE>
    <EXTRACODE>
      <DIR>/tmp/implement/</DIR>
      <DIR>/tmp/implementExtra/</DIR>
    </EXTRACODE>
    <INCLUDES>
      <DIR>/tmp/myinclude/</DIR>
      <DIR>/tmp/anotherinclude/</DIR>
    </INCLUDES>
    <CONTROLLERDIR>/tmp/controler/</CONTROLLERDIR>
```

```
</CODEGENERATION>
</GUIDEKERNELSETTINGS>
```

Different parts are explained here:

1. **PLUGANDPLAYTYPESDIR.** This is the directory where all pre-defined types for GUIDE are kept. The philosophy and practice of types' handling in GUIDE is presented in [SC]; the directory has a particular structure where the types' definitions are present, and also the variation operators that can be applied to them. This is the detail of that structure (Table A.1)

```
typesPlugAndPlay/
    operatorSchema.xml
    bag/
    bool/
    :
    all pre-defined types here
    :
```

Table A.1: Directory Structure for Types Plug and Play

On each type, there is a sequence of sub-directories that completely define it. See Table A.2 , plus Each type has its definitions: **defgene.xml** and **deftype.xml**.

```
bag/
    defgene.xml
    deftype.xml
    cross/
    mut/
    init/
```

Table A.2: Directory Structure for a Type

Each type of variation operators (crossover, mutation and initialization) are represented by a directory with actual definitions, separated by evolutionary libraries. See Table A.3

```
cross/
  EO/
  ECJ/
```

Table A.3: Directory Structure for a Type

For each evolutionary library, there is the actual code for it. Table A.4

```
cross/
  EO/
    cross bag all elements.xml
    cross bag k elements.xml
    cross bag one element.xml
    cross bag uniform plus.xml
    cross bag uniform.xml
  ECJ/
    file 1.xml
    file 2.xml
    :
    file n.xml
```

Table A.4: Directory Structure for a Type

2. **LOGFILENAME.** GUIDE uses a Log to keep a register of the system's behaviour. This is the file name for the XML file where the Log messages are kept.
3. **LOGLEVEL.** Determines what kind of messages are going to be logged. There are 3 levels of Log levels:
 - **SEVERE:** errors that threaten the normal functioning of the Kernel.

- **WARNING:** weird situations that we know how to handle by default.
 - **INFO:** situations that are typically communicated
4. **CURRENTEXPERIMENT.** Last experiment worked with GUIDE.
 5. **XSD.** Handles the lookup of the address where the grammar for the Evolutionary Algorithm files are handled.
 - (a) **ISONLINE.** Holds TRUE if the file has to be looked up online. FALSE otherwise.
 - (b) **ONLINE.** Address online where the XSD file has to be looked up. Currently, that address is `http://www.lri.fr/~ldacosta/evoxml/algorithmSchema.xsd` (so, unless you have a VERY good reason to do so, you should not modify this address).
 - (c) **OFFLINE.** Local address where the XSD file has to be looked up.
 6. **CODEGENERATION.** This section controls the parameters related to the program (code) generation.
 - (a) **TargetLibrary.** This parameter specifies the library for which the code is going to be generated. Currently (March 2008) two options are available: **EO** (for Evolution Objects Library) and **ECJ** (for Evolutionary Computation in Java Library).
 - (b) **VERBOSE.** If this parameter is TRUE, the code generation will be contain hints and comments. If FALSE, it will be "minimalistic".
 - (c) **INCLUDES.** Directories with extra header files needed to compile the application. Each instance of a directory is inside a tag DIR.
 - (d) **EXTRACODE.** Directories with other source files that need to be compiled. For example, if the fitness function uses external libraries, the address should be included here. Each instance of a directory is inside a tag DIR.
 - (e) **CONTROLLERDIR.** The controller of the algorithm must be in this directory.

Bibliography

- [CS03] Pierre Collet and Marc Schoenauer. GUIDE: Unifying evolutionary engines through a graphical user interface. In Pierre Liardet, Pierre Collet, Cyril Fonlupt, Evelyne Lutton, and Marc Schoenauer, editors, *Evolution Artificielle, 6th International Conference*, volume 2936 of *Lecture Notes in Computer Science*, pages 203–215, Marseille, France, 27-30 October 2003. Springer. Revised Selected Papers.
- [KMRS02] M. Keijzer, J. J. Merelo, G. Romero, and M. Schoenauer. Evolving Objects: a general purpose evolutionary computation library. In P. Collet et al., editor, *Evolution Artificielle'01*, pages 229–241. LNCS 2310, Springer Verlag, 2002. URL: <http://eodev.sourceforge.net/>.
- [SC] Marc Schoenauer and Luis Da Costa. Representation-independent evolutionary algorithms: Principles and guide. Submitted to the Journal of Artificial Intelligence Tools.