ArgoUML Quick Guide

Get started with ArgoUML 0.34

Kunle Odutola
Anthony Oguntimehin
Linus Tolke
Michiel van der Wulp

ArgoUML Quick Guide : Get started with ArgoUML 0.34

by Kunle Odutola, Anthony Oguntimehin, Linus Tolke, and Michiel van der Wulp

Copyright © 2001 Kunle Odutola

Copyright © 2004-2010 Michiel van der Wulp

Table of Contents

Preface

1 April 2001

In the last decade of the last millenium, object-oriented (OO) technology finally completed it's transformation from a laboratory curiousity in the 1960s to become *the* mainstream software development paradigm. It has been a long, tough journey - that nearly ended in obscurity - primarily because OO technology required a major shift in the thought processes of systems designers, developers and others involved in the software development lifecycle.

It wasn't until the introduction of OO-hybrid languages such as C++, Classic Ada and Object Pascal in the eighties that the mainstream was introduced to and, given the opportunity to experiment with OO technology. These languages combined support for OO programming with support for traditional procedural programming. The lack of support of procedural programming in pure OO languages had hindered the adoption of OO technology for many years. The journey that had begun with the introduction of *objects* in Simula in the 1960s was now complete. OO technology was now accessible to the mainstream...well, OO programming was at least.

Simulation is the basic premise that underpins OO technology. An OO system is designed and implemented essentially as a simulation of the real world using software artifacts. This premise is as powerful as it is simple. By designing and building software systems in this manner, the same language and ideas can be used in the analysis, design and implementation of OO systems. This allowed a system to be designed and tested (or more correctly allowed a system to be simulated) without having to actually build the system first. This feature coupled with the ability to design systems at a very high level empowered experienced OO practitioners to design and successfully implement more complex systems than had previously been possible.

The availabilty of various OO methodologies and the eventual adoption of the Unified Modelling Language (UML) as the standard language for communicating OO concepts continued the advance of OO technology into the mainstream. The popularity of the object-based Visual Basic language and the parallel development and meteoric adoption of the fully object-oriented Java programming language was the final catalyst that fuelled the final stages of the rapid adoption of OO technologies by the mainstream.

ArgoUML was conceived as a tool and environment for use in the analysis and design of object-oriented software systems. In this sense it is similar to many of the commercial CASE tools that are sold as tools for modelling software systems. ArgoUML has a number of very important distictions from many of these tools:

- ArgoUML includes a number of features that support the cognitive needs object-oriented software designers and architects.

- ArgoUML supports open standards extensively - UML, XMI, SVG, OCL and others. In this respect, ArgoUML is still ahead (even almost five years after it's initial introduction) of many commercial tools.

- ArgoUML is a 100% Pure Java application. This allows ArgoUML to run on all platforms for which a reliable port of the Java2 platform is available.

- ArgoUML is an open source product. The availability of the source ensures that a new generation of software designers and researchers will now have a proven framework from which they can drive the development and evolution of CASE tools technologies.

In creating the ArgoUML development tool and environment, Jason Robbins and the rest of his research team at UCL leveraged the benefits of UML as the most prevalent OO modelling language and Java as one of the most productive OO development platforms to produce a solid tool for OO systems design

and further a testbed for the evolution of OO CASE tools development and research.

Kunle Odutola

Chapter 1. Introduction

Kunle Odutola

1.1. Welcome to ArgoUML

ArgoUML is a powerful yet easy-to-use interactive, graphical software design environment that supports the design, development and documentation of object-oriented software applications.

If you are familiar with a family of software applications called Computer Aided Software Engineering (CASE) tools then you should find ArgoUML instantly familiar.

The users of ArgoUML are software designers & architects, software developers, business analysts, systems analysts and other professionals involved in the analysis, design and development of software applications. Main features:

- Open standards: XMI, SVG and PGML

- 100% Platform independent thanks to the exclusive use of Java

- Open Source, which allows extending or customizing.

- Cognitive features like: reflection-in-action, opportunistic design, comprehension and problem solving

1.2. About this guide...

This guide - the ArgoUML Quick Guide - is provided to help you get up and running as quickly as possible with ArgoUML. It provides step-by-step instructions for obtaining, installing and starting ArgoUML, performing common tasks, and learning more about ArgoUML. If you need further details please look in the User Manual [http://argouml-stats.tigris.org/documentation/defaulthtml/manual/].

Chapter 2. Installing ArgoUML

2.1. System requirements

Minimum system requirements:

- Any Operating System that supports Java.

- 15MB of free disk storage space on your hard disk.

- Mouse (or other pointing device) and a Keyboard.

- Java JRE or JDK version 5 or higher.

2.2. Installation options

Now that you have a computer that can install and run ArgoUML effectively, you need to decide which of the installation options available is appropriate for you. You can choose from the following options:

- Java Web Start

  This alternative is good for occasional users and tests. It is the easiest and quickest way to start using ArgoUML. It requires connection to the ArgoUML homepage [http://www.argouml.org/].

  1. Make sure you have Java Web Start [http://java.sun.com/products/javawebstart/] installed.

  2. Follow the Launch ArgoUML-link from the ArgoUML homepage [http://www.argouml.org/].

  3. ArgoUML will be downloaded, cached locally and started.

  4. On subsequent starts, the ArgoUML is available also from the Java Web Start console (even without being connected to the internet)and (if connected) at each start the version is checked against that on the web server and updates are automatically downloaded.

- Use the Windows Installer

  This installation method is the easiest way for most Windows users. It also installs all available modules, which includes support for e.g. C++ and PHP.

  1. Download the installer from the ArgoUML homepage [http://www.argouml.org/]. The filename is of the form `ArgoUML-vvvv-setup.exe` (with vvvv the version number).

  2. Run the downloaded installer program and follow the directions on the screen.

  3. The install wizard gives you the opportunity to install the latest "JRE" (Java Runtime Environment). There is no need to select this if you already have a Sun Java installed, version 5 or better.

- Binary distribution

  This is good for regular users and guarantees that the version of ArgoUML cannot change during the course of your project.

1. Make sure that you have a recent Java JRE installed.

2. Download the ArgoUML binary distribution from the ArgoUML homepage [http://www.argouml.org/].

    This could also include copying it to and from a USB memory stick or CD in order to get it to the computer if no network connection is available.

3. Create an ArgoUML installation directory.

4. Change the current directory to the ArgoUML installation directory and extract the ArgoUML files to the installation directory.

5. Start by running the `argouml.jar` file, either by simply (double-) clicking on the file, or by executing the following command: `java -jar argouml.jar` at the command line, or via a batch-file.

2.3. Installing Auxiliary Modules

The standard installation of ArgoUML already comes with several modules which add e.g. support for C++, php and C# code generation. In some cases, you may want to add even more functionality by adding a third party module. To make this possible, you first need to download the module.

Unpack the downloaded file in the same directory as you unpacked ArgoUML itself.

The result should be that the directory that contains the `argouml.jar` file, now also contains a subdirectory named `ext`, which contains the `.jar` files for the extra languages.

2.4. Command Line Options

When starting ArgoUML from a command line, there are several extra possibilities. Try typing:

java -jar argouml.jar -help

You will see the instructions:

```
Usage: [options] [project-file]
Options include:
  -help           display this information
  -big            use big fonts
  -huge           use huge fonts
  -nosplash       don't display logo at startup
  -norecentfile   don't reload last saved file
  -command <arg>  command to perform on startup
  -batch          don't start GUI
  -locale <arg>   set the locale (e.g. 'en_GB')
  -open <arg>     open given file on startup
  -print <arg>    print given file on startup (and exit)

You can also set java settings which influence the behaviour of ArgoUML:
  -Xms250M -Xmx500M  [makes ArgoUML reserve more memory for large projects]
```

A common problem is that the User Interface is shown in the wrong language. Below is an easy way to switch back to the English UI. Please beware: the language is in lowercase.

java -jar argouml.jar -locale en

ArgoUML may be run without UI, in batch mode. Currently the possibilities are very limited. Below is an example (everything on 1 line!): It reads a "test.zargo" from my working directory, gets the diagram named "A", and writes a PNG graphical file for this diagram. Without the "-batch", ArgoUML would start up the UI after executing the commands.

java -jar argouml.jar -batch -command "org.argouml.uml.ui.ActionOpenProject=c:\Documents and Settings\Michiel\My Documents\test.zargo" -command org.argouml.ui.cmd.ActionGotoDiagram=A -command "org.argouml.uml.ui.ActionSaveGraphics=c:\Documents and Settings\Michiel\My Documents\test.PNG"

2.5. Making .zargo-file clickable (on any Windows)

The Windows installer for ArgoUML assigns the ".zargo" extension to the ArgoUML application, so that double or single clicking on a ArgoUML project file automatically starts ArgoUML.

However, if the installer is not used, but one of the other installation methods is used, then you may want to configure Windows manually for this. In this case, follow the explanation below.

2.5.1. Making .zargo-file clickable (on Windows XP)

Ewan R. Grantham

> **Warning**
>
> We have reports that this isn't working on Windows Vista. Please don't try it there - see the next chapter instead.

This works only if you have installed using the binary distribution.

First, find a Zargo file, and right-click on it. You should get the normal Windows menu, including an Open or Open With option. At this point, give a description for the file like "ArgoUML Model", and tell Windows to use Notepad to open the file. That's not what you want to do ultimately, but it does get Windows to accept the .zargo extension as representing a valid file type.

Now, open Windows Explorer and from the menu select View->Options (or on Windows XP: Tools->Folder Options...). You should get a two (or more) tabbed dialog. Click the "File Types" dialog, and then scroll through the list for the Description you gave - in the example "ArgoUML Model". Click to select that file type, and then click the "Edit" button.

Now, click on the "Open" action and then the "Edit" button there. You will then get a dialog that has a line for entering in the application action which will have a line for using Notepad to open the file. Replace that line with: **"C:\Program Files\Java\j2re1.6.0_05\bin\javaw.exe" -jar "c:\ArgoUML\argouml.jar" "%L"**

Substitute your path to your `javaw.exe` and `argouml.jar` files if they are in a different location. Click OK, then click OK, then click OK (three buttons on three different levels of the dialog).

The %L in the above commandline is needed instead of %1 because that is the way to support filename extensions of more than 3 characters.

2.5.2. Making .zargo-file clickable (on Windows Vista)

Allan Todd

1) Create a batch file "runargo.bat" in your ArgoUML directory which looks something like this: **@echo off cd "C:\ArgoUML" start javaw -cp "argouml.jar" org.argouml.application.Main %1 /B** replacing "C:\ArgoUML" as appropriate.

2) Associate your .zargo files with this batch file (just double-click a .zargo file and follow the instructions).

Chapter 3. ArgoUML Principles

When ArgoUML starts, it shows an empty class diagram on which you can add various objects. ArgoUML works according the following principles:

3.1. Project, Model and Diagram

The file operations save and open one `project` at a time. One `project` is composed of a `model` plus diagram information, i.e. everything you can edit within the ArgoUML window.

The `model` may contain many objects (ModelElements) which form the complete UML description of the system you are describing. All ModelElements might be present on a diagram, but this is not required. Hence, the `model` that is stored in ArgoUML is independent of the contents of the diagrams. This may be explained by the possibility to generate programming code from the `model` - you do not need any diagrams for this.

An ArgoUML `project` also contains all diagram information, i.e. the shapes (presentation) used to represent the various UML ModelElements, their location, color, etc. Some ModelElements appear on multiple diagrams, some on one or none.

Hence, saving and opening projects retains all this information. There is a way to only save the `model` information though, which is by the menu "File" -> "Export XMI...". This may be usefull e.g. when generating programming code with an external tool that understands XMI.

3.2. Objects

Select objects by left-clicking on them. The functionality of ArgoUML can be activated in the menu, in toolbars, or in pop-up menus when right-clicking above an object. Many of these functions work on the selected objects.

All diagrams have toolbars at the top which are used to create objects on the diagram.

Most objects can be added to a diagram and removed from a diagram without deleting it from the `model`! Select an object on a diagram, and then the menu item "Edit" -> "Remove from Diagram" clears the object from the diagram, but the object remains intact in the model, which may be seen in the explorer, i.e. the tree structure at the left hand side. Once removed, and item can be added back to the diagram (or any other diagram for that matter) by selecting it in the explorer and selecting the item "Add to Diagram" in the right-click pop-up menu (then click in the diagram to place the item). Alternatively, you can drag objects from the explorer and drop them onto a diagram.

3.3. Overview of the window

The top of the window contains a menu bar with commands available. In the `File` menu you can store the project or open another project instead.

The upper left part of the ArgoUML window shows a tree model of diagrams and objects. This view can be adapted to your needs by filtering the objects that are shown, and the structure in which they are shown.

The upper right part of ArgoUML shows the current diagram (one at a time). You can drag and drop the objects in the diagrams, and you can use the quick-links that appear when hovering over a selected object to create new objects connected to the already present objects.

The lower right part contains various details of the currently selected object: You select the object in one

of the upper levels and choose what details you want to examine using the tabs.

The lower left part contains a list of all ToDo items for this model.