Artifact Submission

haowu@cs.nuim.ie

Computer Science Department, Maynooth University

1 - Description

This artifact submission represents a tool named **QMaxUSE** that is described by the paper:

QMaxUSE: A Query-based Verification Tool for UML Class Diagrams with OCL Invariants

This artifact is designed for verifying UML class diagrams with OCL invariants. This artifact supports two features:

a) A query language that allows users to select parts of a UML class diagram to be verified.
b) A new algorithm that is capable of performing concurrent verification.

2 – Which Badges

We submit our artifact for badges: both **evaluation and availability badges** of EAPLS.

3 – Download Artifact (QMaxUSE)

Our artifact (QMaxUSE v1.0.0) is available at the following web address:

https://doi.org/10.5281/zenodo.5804509

The latest version of QMaxUSE is also available at:
https://github.com/classicwuhao/qmaxuse

There are four binaries and you may try any one of the following packages for a specific operating system:

a) qmaxuse-mac-x64 (for MacOS)
b) qmaxuse-ubuntu-x64 (for Ubuntu)
c) qmaxuse-win-x64 (for Windows)
d) qmaxuse (for MacOS/Ubuntu/Windows OS)
e) z3-libraries (libraries for running maxuse)

QMaxUSE currently supports three operating systems: Windows 10, Ubuntu 15.04+, MacOs. Essentially, QMaxUSE does not require to install any additional libraries. However, if missing libraries are detected please contact us (haowu@cs.nuim.ie). The detailed instructions about how to run and use QMaxUSE is available in the following sections.

3 - Run QMaxUSE.

1. Download one of the zip files (depends on which operating system you are using) mentioned in the section above and extract the zip file.
2. Once the zip file is extracted, run the following command (**from terminal**) in the **lib** directory:
   java -jar qmaxuse.jar
3. This will launch the main panel for QMaxUSE and its command prompt.

NOTE: For Windows, you need to enable ANSI color mode for Windows 10 to avoid seeing color codes in QMaxUSE. How to do it please click here to see the instructions. For Ubuntu, openjdk version:1.8.0+ is required.

4 – Replicate Queries

Once QMaxUSE is launched from terminal, you can see its main panel and also its command prompt **QMaxUSE>**

1. **To replicate the query from our paper (Fig.3. in Section 3)**, you first load our sample model. This is can be done using the menu:

   a. File -> Open Specification
   b. Locate “query_examples/university.use”

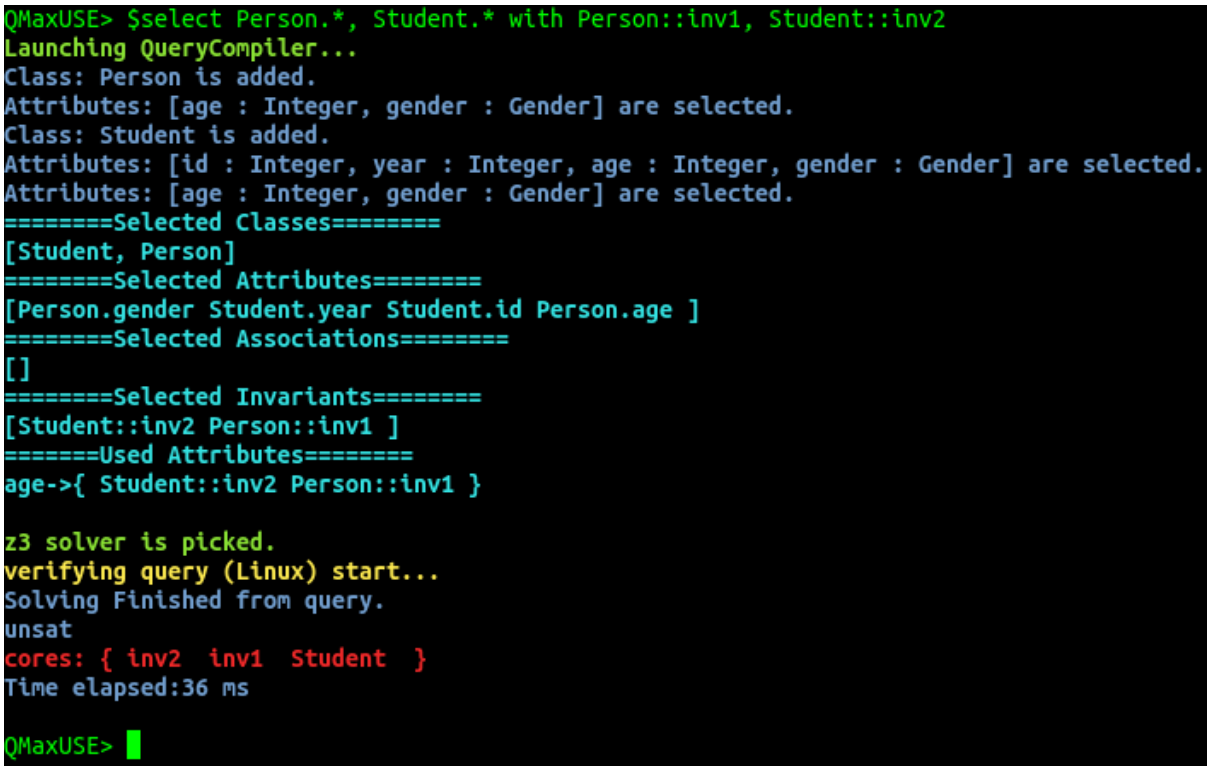
2. Once the model is successfully loaded, you now can issue a query at the command prompt and each query must begin with a $.

3. To replicate the result from Fig.3. in Section 3 (in our paper), you can type the following query at the **QMaxUSE>**

```
$select Person.*, Student.* with Person::inv1, Student::inv2.
```

This should show a query verification result that is similar to the one in our paper. The actual time elapsed may differ depending on your machine spec.

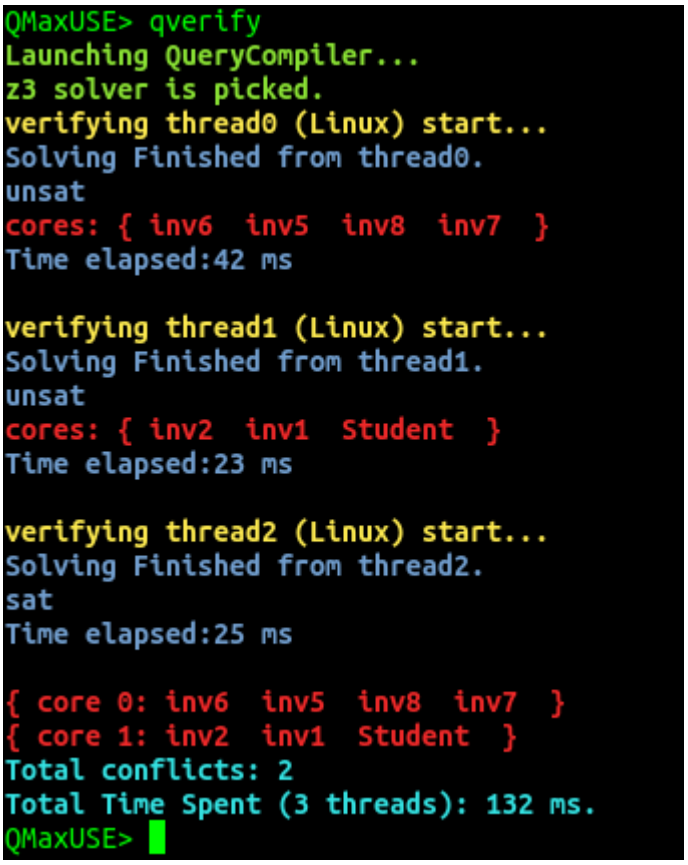
The following screenshot shows a running sample query from our paper.



4. **To replicate concurrent verification,** you can just simply type the following command at QMaxUSE command prompt.

   qverify

This will concurrently verify this sample model using 3 threads and you will see a screenshot as follows:



Again the time spent here may vary. This depends on your machine spec.

5 - Replicate Evaluation

This section introduces how to replicate our evaluation results. In order to replicate our evaluation, one must configure z3 libraries in order to run MaxUSE and compare its performance to QMaxUSE. Here we show how to do it in Ubuntu.

To enable MaxUSE, please download and use the z3 libraries in Section 3. Or you can always also click the following link to download it:

https://github.com/classicwuhao/qmaxuse/releases/download/Latest/z3-libraries.zip

Unzip this file, and it contains 3 files:

com.microsoft.z3.jar, libz3.so, libz3java.so

copy these 3 files into a new folder (under your home directory ~) called "z3lib"

then append the following lines into your **.bashrc** file:

```
Z3_LIB = ~/z3lib

CLASSPATH=:$CLASSPATH:$Z3_LIB

LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$Z3_LIB

export CLASSPATH

export LD_LIBRARY_PATH
```

then do following two things:

1. Copy com.microsoft.z3.jar to the **lib** directory (where maxuse is).
2. Create a directory called **html** in the **lib** directory (for maxuse to log html-based information).

Once you have done this, close everything including QMaxUSE and reopen your bash terminal. In the terminal, launch QMaxUSE again, and load our sample model (described in the above Section).

Now, you can type the following command at QMaxUSE command prompt to use our maxuse technique for verification:

```
maxuse
```

This will verify a model using our previous technique. Now you can compare the time spent by using maxuse and qverify to see the difference. The difference should be significant and qmaxuse boosts verification speed at least 10x-30x.

All the benchmark files are stored in "query_examples/benchmark". It has five groups (**GroupA - GroupE**). **GroupA** represents **Part A** in Table 1 in our paper, and rest of groups are **Part B.** However, B3 in Table 1 (in our paper) means B3.use in **GroupB**.

Now you can load each individual model from our benchmark verify the performance gained. Please note that the speed gained depends on your machine spec. However, you should see a significant performance improvement as our paper claimed.

6 – Checksum

The checksum for our five zip files are:

Name of zip file	**Checksum**
qmaxuse-mac-x64.zip	2108dca5bce8d3d7403782ec51af1cf5da66e1d548f2cbfa0f9331ce11b60542
qmaxuse-ubuntu-x64.zip	c10cc0fa87d8e93897132ac8c5cf29ad556d8c135ac97159b69188e4d3a8baa1
qmaxuse-win-x64.zip	fcfb4b2d05d4f7f50381f5b6a082be43ea6c0760146826f5e57340086e44bdd9
qmaxuse.zip	b22fd20b802d37f1964decc207c5d2262923f86eecf6c86a3ab4b8f871ce92b7
z3-libraries.zip	158825041700a5ee3b88c117db8470b35440fdea9cadb0d01119ab148bdc205b
qmaxuse-all.zip	06dcb396c8c81eee898d0c383cdef2bc861965ef007246ba50bcaa7c08803dfd