HERA Installation

General Information

First, make sure you have Python3 installed.

Here, Python 3.5, 3.6, 3.7 have been used/tested.
(10.03.2020)

Update:
Newest Python version Python 3.8.2 works as the ones mentioned before.
(28.04.2020)

Installation on Linux

Tested with Ubuntu 18.04 as main operating system and with a virtual machine running on Windows 10.
(10.03.2020)

Update:
For the newest Ubuntu version 20.04 everything works the same way.
(28.04.2020)

Additional to Python3, several other system package dependencies have to be installed. Make sure they are installed (for the respective Python version) before starting the HERA installation or install them using the command *apt install*.

- gcc
- g++
- git
- make
- python3-dev
- python3-pip

Using pip, two other system package dependencies necessary to correctly execute setup.py can be installed, if not yet existing, with the command *pip3 install* respectively:

- setuptools
- requests

The github repository can be cloned using the following command:

git clone https://github.com/existenzquantor/ethics.git

Enter the newly cloned folder before executing the second command, that sets up the project to make it runnable.

python3 setup.py install

If the installation shall happen for the whole system, it may be possible that an additional *sudo* in front of the command is required. (However, when installing the project in an virtual environment, do not use *sudo*, even though at first everything seems to run fine).

To be able to run the test classes, two Python packages are required, as stated in the requirements.txt, using or *pip3 install* respectively.

- pyeda
- PyYAML

It is good to install pyeda before trying to install PyYAML, otherwise, there may occur an error.

In one case, test classes using PyYAML could still not be executed. Updating the Python package using *pip install PyYAML - -upgrade* solved the problem.

The paths used in the test classes indicating which other resources are used vary over the test classes. While some test classes assume the test to be executed from the main project folder (*"./cases/plans/bottle.yaml"*), others assume it to be executed from the test folder (*"../cases/cam/trolley-dilemma.json"*). Of course the paths can be modified as needed.

As this can lead to confusion when running tests using an IDE, in the following it is stated which test class to run from which folder in the directly downloaded recent version:
(10.03.2020)

ethics:

- causality_test.py
- test-plans.py
- test-robot-frank.py

test:

- test-explanations.py
- test-planner.py
- tutorial_plan_explanations.py
- tutorial_plans.py

Installation on Windows

First, it has to be mentioned that it was not possible to run the project only on Windows 10. Although some steps that are more complicated on Windows than on Linux can be bypassed, one major issue could not be resolved using the *setup.py* file as it is.

In the following, this problem is described.

However, it is recommended to use a virtual machine or Docker that uses Linux on Windows instead of trying to install the project directly on Windows.

Besides from Python3, requests (*pip install requests*) and *git* (downloading an .exe file on the website https://git-scm.com/downloads) have to be installed.

Using the git bash shell, the following command can be executed to clone the project repository:

git clone https://github.com/existenzquantor/ethics.git

When now executing the command (with *python* and not *python3* as in Linux)

python setup.py install

the following error occurs:

Configuring CUDD ... error: [WinError 2] Das System kann die angegebene Datei nicht finden

The C library CUDD needed for the Python C extension mhs cannot be compiled directly on Windows, as it cannot execute the command *./configure* listed in the *setup.py* file.

To try to work around that, *Cygwin* can be used to provide functionalities not given in Windows. (https://cygwin.com/install.html)

When running the *Cygwin* installation, the following packages shall be chosen to be included in the installation when being asked for it, as they are needed for the project.

- bash
- make
- gcc core
- g++

Using the *Cygwin* command line, executing the *python setup.py install* command produces another error: arguments in the file cannot be understood by Windows (*-W*).

Trying to compile CUDD manually with the command *./configure* and *make* seems to work, producing the output *CUDD was built*.

However, it is not possible to afterwards correctly link the CUDD library.
One possible explanation for that is that CUDD was not properly configured.
Another one is that CUDD is built using the gcc compiler, but when executing *setup.py* another compiler is used.

To be able to install the project on Windows without complications, an alternative to *setup.py*, using commands understood by Windows would be necessary. As it is, using Docker on Windows (mentioned underneath) is the easiest option.

Another problem, unrelated to the one mentioned before, is that the necessary package *pyeda* cannot be automatically compiled using Windows.
However, the solution for that is to grab its binaries (*https://www.lfd.uci.edu/~gohlke/pythonlibs/*) and to compile it manually before using it for the project.

This takes more time than it does using Linux, but is possible to do.

Use in IDE

To run the project using Pycharm or Eclipse, it first has to be cloned and installed as stated above.
(Here Pycharm 2020.1 and Eclipse 2019-12)
(10.03.2020)

Update:
On Ubuntu 20.04 Eclipse 2019-12 can be installed but not executed.
(28.04.2020)

It is necessary to install *pydev* for Eclipse, to be able to write and run Python code. All other steps are the same for both IDEs.

The easiest way to get *ethics* into the IDE is to create a new project, change the directory of it to the directory of *ethics* and import all existing files in that folder.

Afterwards, trying to run the test classes the following error can occur:

```
ModuleNotFoundError: No module named 'ethics.extensions.mhsModule'
```

In that case, it is necessary to execute the following command in the *ethics* folder:

```
sudo python3 setup.py develop
```

The problem that occurred before should then be solved.

Using an IDE, it is possible to run the test classes that are executable from the *test* folder. The test classes executable from the *ethics* folder however, have to be changed. The paths importing modules and files shall be adjusted.

Use in Docker

Another easy possibility to run the project on Linux, Windows or MacOS is to use Docker.

For that, Docker (https://docs.docker.com/engine/install/) and git (as described above) have to be installed and the repository has to be cloned.

The attached Dockerfile then has to be placed inside of the ethics folder and run using the following command:

docker build .

This builds an environment using Ubuntu 18.04 with all necessary packages installed ready to execute the project classes.