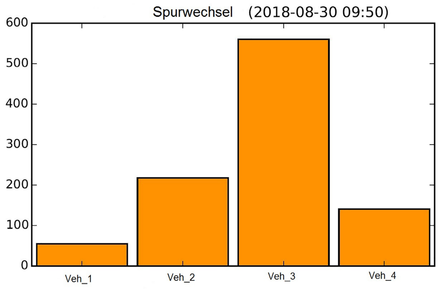
VALIDATION
AND TEST

Ko-HAF
KOOPERATIVES
HOCHAUTOMATISIERTES FAHREN

Logging Tool for Evaluation of Ko-HAF Tests and Validation

INTENTION:
INTERPRETATION OF TEST RESULTS OF KO-HAF DRIVING FUNCTIONS

For example we can count how many lane changes were successfully performed over the project on regular Highways, or on the test track in Dudenhofen or all put together.

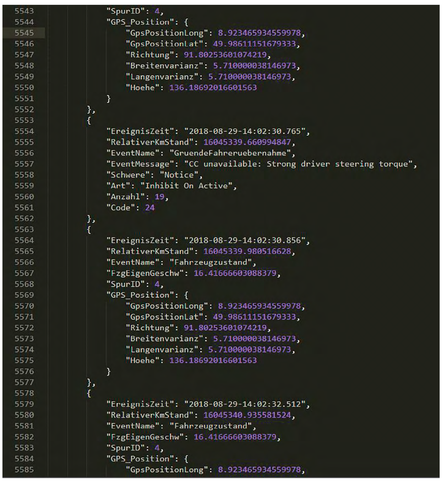


How: While Ko-HAF fleet's vehicles are driving on Ko-HAF test domain, each of them register some **environment data** upon specific **events** on a so called **json file**.

Once the Ko-HAF partner have uploaded their json file a **python script** browses all the json files.

The interesting **events** for the evaluation are for example:

- Start of a test run
- Start of autonomous driving
- Start of a lane change
- Entering a highway and filter into the traffic
- Exit from a highway
- Interruption of automatic driving



Environment data that are being stored in such situations, are test vehicle's speed, its acceleration, and same properties of detected objects in the direct environment of the test vehicle. Lane where the event occured, localisation on the map.

```python
# -*- coding: utf-8 -*-

import os
from database_file import DatabaseFile, LogException, NotStartedException, \
    NotStoppedException


class Database(object):
    def __init__(self, folder):
        self.tracks = []
        self.json_files = []
        # Check whether we got a single folder or a list of folders as
        #   database root.
        if type(folder) == str:
            self.collect_files(folder)
        else:
            for f in folder:
                self.collect_files(f)
        self.read_files()

    def collect_files(self, folder):
        print "Reading folder %s..." % folder
        json_files = []
        for root, dirs, files in os.walk(folder):
            for file in files:
                filename = os.path.join(root, file)
                if filename.endswith(".json") and not filename.endswith(".bak.json"):
                    json_files.append(filename)
        self.json_files = self.json_files + json_files

    def read_files(self):
        files = sorted(self.json_files)
        for file in files:
            try:
                self.read_file(file)
            except NotStartedException:
                pass
            except NotStoppedException as e:
                print "In file %s the end of the drive couldn't be detected (%s)." \
                      % (file, e)
            except LogException, e:
                print "File %s is corrupt (%s)" % (file, e.message)
            except IOError, e:
```

From such information, it is possible to infer how well the **Ko-HAF system performs in real traffic** as well as while deterministic and reproduceable tests, like the ones that can be performed on the Ko-HAF test track.

Continental

Gefördert durch:
Bundesministerium für Wirtschaft und Energie
aufgrund eines Beschlusses des Deutschen Bundestages

www.ko-haf.de