

ITEWS Simulation Studies using Software Simulator/Emulator - Overview
ITEWS Technical Report Series
Document ID: TR-ITEWS-2
Date: Oct 22, 2006

Goal

The design goals for the simulator were to create as much as possible a Domain Specific Language for inter-operating with its various components. For instance, the essence of "Tell the bus with license number KL5-2358 to move from the TVM station to the KLM station at 10 am," is accomplished through the use of a software protocol known as YAML.

Background

A simulation/emulation software has been developed to serve two functions:

1. To exercise and test the telematics system beyond what is feasible in real world testing.
2. To help visualize what is happening in a functional telematics system.

This software is a simulator in that it models real world traffic, i.e. buses coming and going from multiple stations; it is an emulator as it can inter-operate with real world TGate and IVTUs. This software will be referred to as the simulator.

YAML Files

The example of "Tell the bus with license number KL5-2358 to move from the TVM station to the KLM station at 10 am," is expressed in YAML as:

```
-route TVM to KLM
  - vehicle:          KL5-2358
  - time of departure: 10:00
```

Here is a complete scenario in YAML, showing two vehicles in a planned route from Kayankulam station to Allepuzha station:

```
- !arl.amrita.edu,2006/route
  name: KYM to ALP
  nodes:
    -
      seq: 1
      name: KYM
      tt: 0
      code: T003
    -
      seq: 2
      name: ALP
      tt: 50
      code: T004
```

```
vehicles:
  -
    code: V002
    lic: KL9-3041
    tod: 10
  -
    code: V001
    lic: KL8-5692
    tod: 25
```

Methodology

These plain text lines along with many more, combine into a simulation scenario. Each scenario consists of a number of vehicles traversing multiple routes in the same general time frame. Multiple scenarios can be defined and selected to execute in the simulator. A web interface to the simulator has been developed to facilitate the creation and maintenance of scenarios.

Ruby

The simulator has been written in the Ruby programming language. Ruby is a 100% object-oriented scripting language, that is expressive and adept at creating domain specific languages. It lends itself to rapid prototyping and is of great use in the creation of functional and unit tests. There are a significant number of libraries available for Ruby that increase its utility further. YAML file reading and writing is included in core Ruby. Another benefit is the self documenting nature of the code written in Ruby. While some languages encourage obfuscation by design, Ruby encourages clarity and expressiveness by design.