Josh Hunter – Railway Simulation – Update and future:

Over the past few months I have been seeking out railway data and simulation software for the purpose of fulfilling SACRED[1] steps 5/6.

The intention is to find a simulated railway environment wherein it would be possible to implement Automation Train Operation (ATO) and Automatic Train Protection (ATP) which would classify the system as GoA-2. From this, modifications of the system will be made in order to do two things:

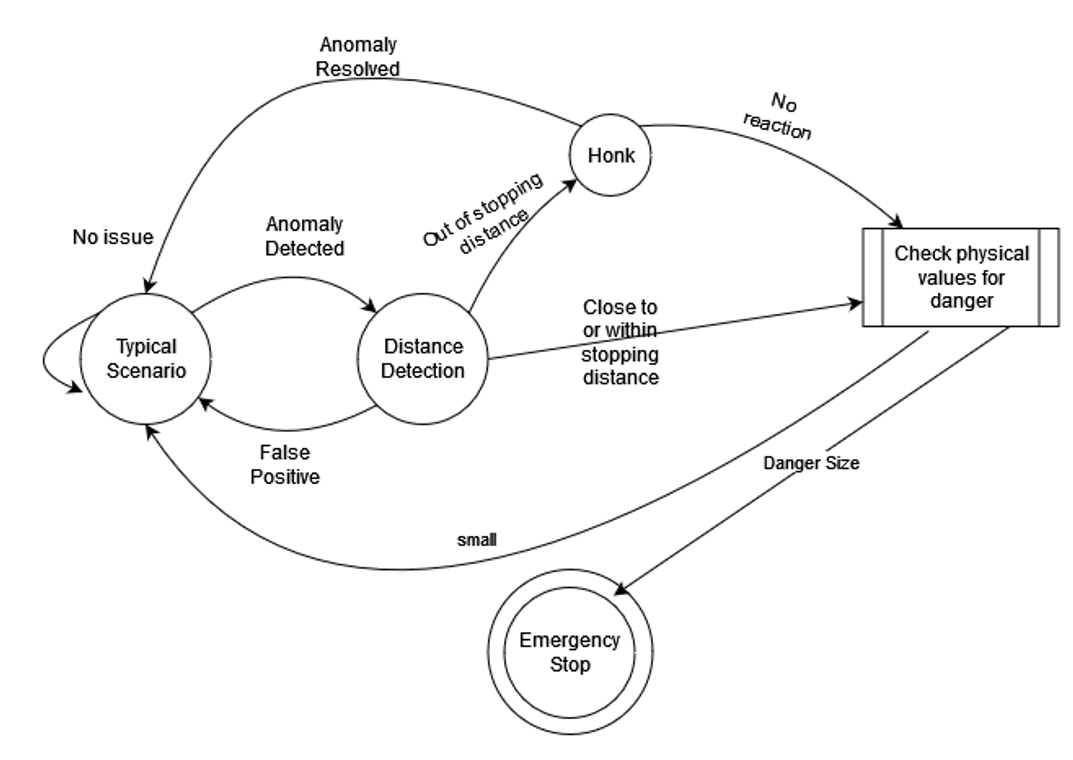
1. Detect track through vision.

2. Detect objects on the track, through vision.

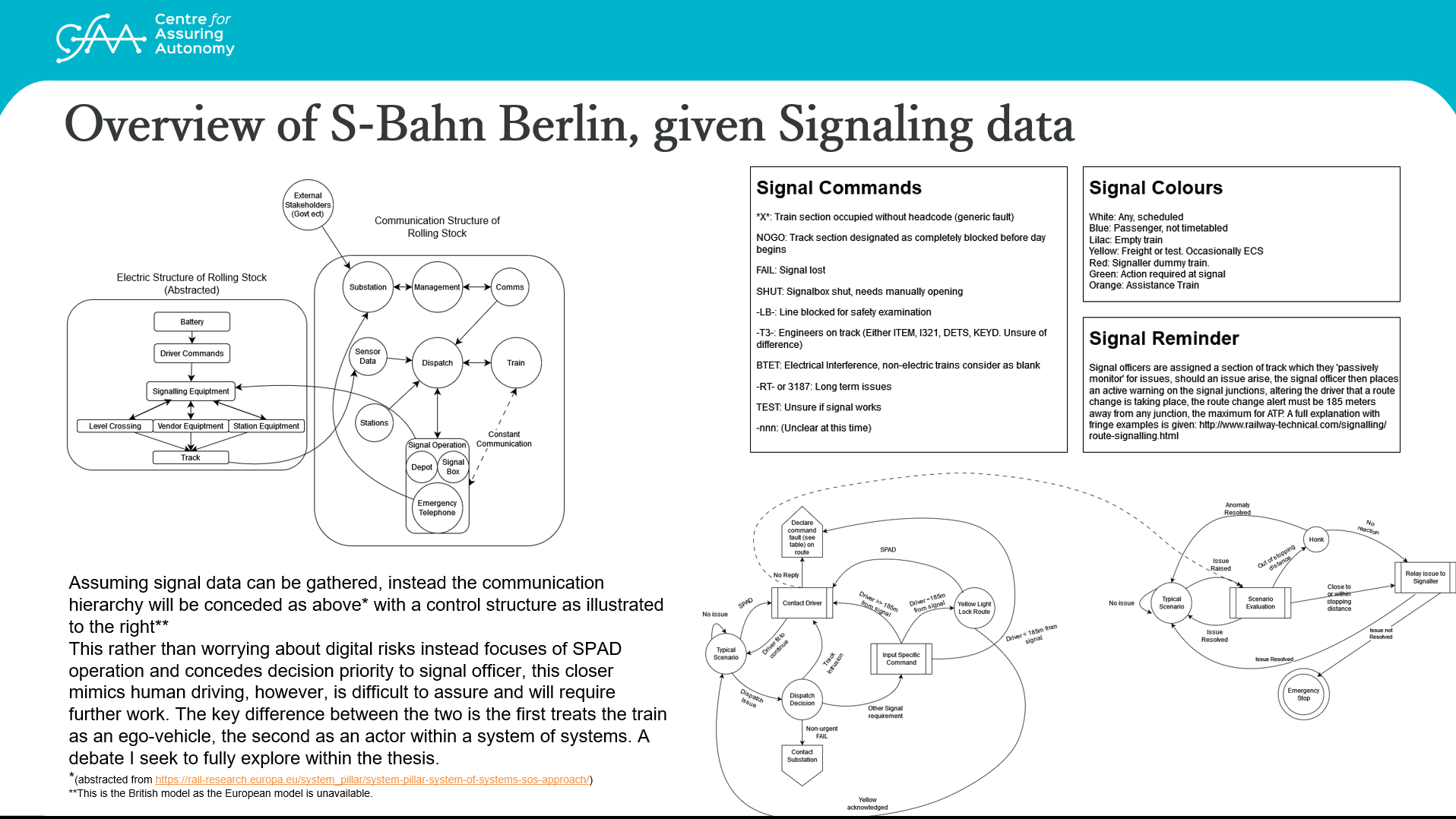
Both of these goals are to be met through collaboration with Fraunhofer IKS.

The simulator used is a modified version of “OpenBVE Railsim.” An open-source railway simulator. For a quick overview of BVE, it exists as a railway specialist program, built as a “more realistic” version of the traditional gamified simulator. The specific version of BVE used is a modified version provided by the OpenBVE team. [2,3] Using UK commercial railways and rolling stock. [4,5] BVE routes for the S-bahn do exist however we were heavily recommended against using them by the OpenBVE team.

Using manual exploration, we created two State Machines for railway operation as part of step 2 of the SACRED methodology, the first existing as an “Ego-Machine” meaning a machine which operates without any external assistance, using only the information it can gather, the second being an “Ecosystem subsystem” meaning it exists within current railway infrastructure of relying upon signal information and processing all information forwards to rail guards.



(State machine Ego version)



(State machine System of Systems version)

These are high level looks at the overall plan of the system, however, sub-diagrams are needed for ‘typical scenario’ (where our simulator currently exists) and “distance detection” (The black-box computer vision system.) The purpose of the simulation is to understand the “Typical Scenario” developed within step 1 of SACREE, the avenue for failure and where exactly any vision system could exist. The specific system which is used depends on the overall application of the system / what the final product should be. Within both examples, processing steps are required, those being steps where a subsystem is required, either for ‘risk classification’ through a fuzzy system (discussed previously, fuzzy system referring to low-power classification such as DeV-iT[15] followed by a high-power verification system, such as YOLO or any proprietary code developed as part of either the collaboration with IKS or as part of the safe tr.ain project. This approach was demonstrated within a presentation given at a previous date[16] overall this approach represents the idea of the “Physical/Digital-Split” an example of which is shown within [18])

However, the ego-approach has problems which need addressing. Current railway metrics of safety (SPAD, SCORE, RPSF among others) all rely heavily upon the infrastructure set up by the ecosystem. Step 4 of SACRED has revealed the necessity of communications between driver and signal operators for regular use of railway systems [9] to remove this communication line is to redesign railway from the ground-up. The SoS diagram showcases two subsystem requirements, with ‘scenario evaluation’ in this case requiring less of a fidelity requirement, information does not need to be at the level wherein a decision can be made and justified, simply it needs to be at a level where information can be related to signal operators, where higher levels of decision can be made off grid. The tradeoff here is that the data would need to be gathered earlier, traditionally by standing cameras, signal sensors or LiDar data. An approach of Track Intrusion detection using machine learning has been published[17] however, further discussion is needed for the purposes of SoS integration.

GoA-2 System:

For the setup of the simulator, we operate on a viewing distance of 1.5km, which we have identified within step 2 of SACRED as the key stopping distance target at ‘reasonable speeds.’ Mode of driving is set to ‘Expert’ which means that all gamified systems have been removed.

The reason behind choosing OpenBVE as a simulator compared to a modified version of CARLA is due to the ground-up dependency on railway rules. While CARLA is a full ego-simulator, OpenBVE considers the entire ecosystem of rail, simulating signal processing, environment change (weather, multiple rolling stock, track quality and air/clamp brakes on track environment.) and other railway-specific requirements. All of which are technically possible within CARLA, but would require a much larger time investment.

The simulator accepts the inputs “Brake, Acceleration, AWS and Honk”

Based on the outputs of “A1, SignalAspect, SignalSpeedLimit, LineSpeedLimit and CurrentSpeed”

And has an internal ‘reward structure’ of “Score.” Score is based on the following tokens:

Negative scoring functions:

Overspeedatsignal -25 -10%

PassedRedSignal -20 -15%

MinorTopple -10 -15%

Derailment -5,000

PassengerDiscomfort -1 -5%

DoorsOpenInTransit -100

PrematureDeparture -20

LateToStop -50

Positive scoring functions:

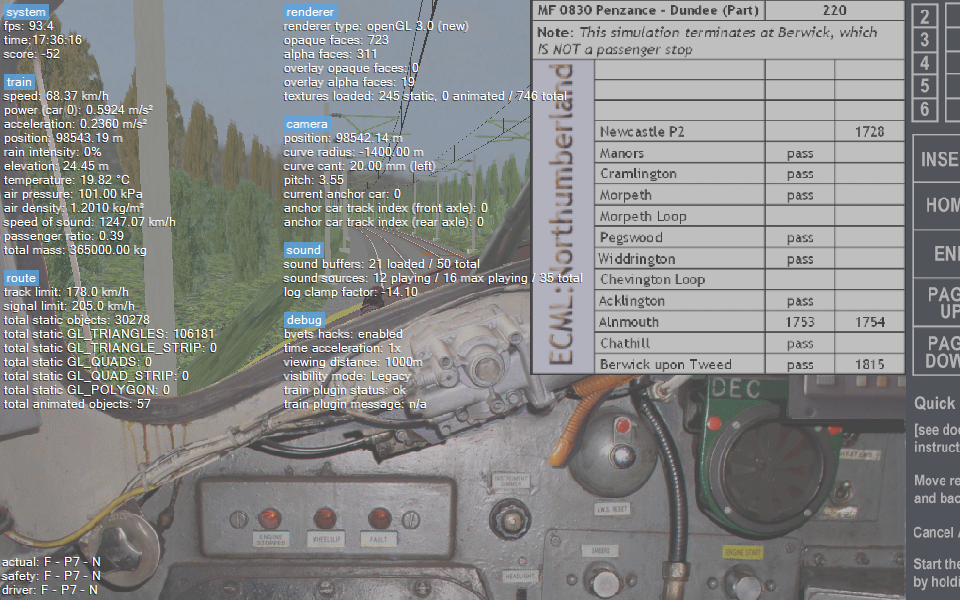
ArrivalAtStop +10

OnTimeBonus +10%

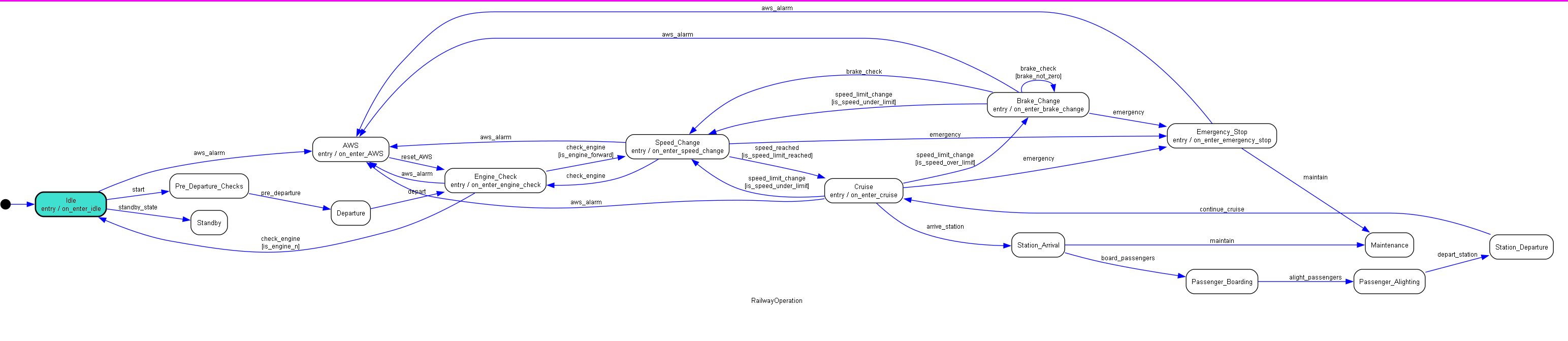
PerfectStopBonus +10%

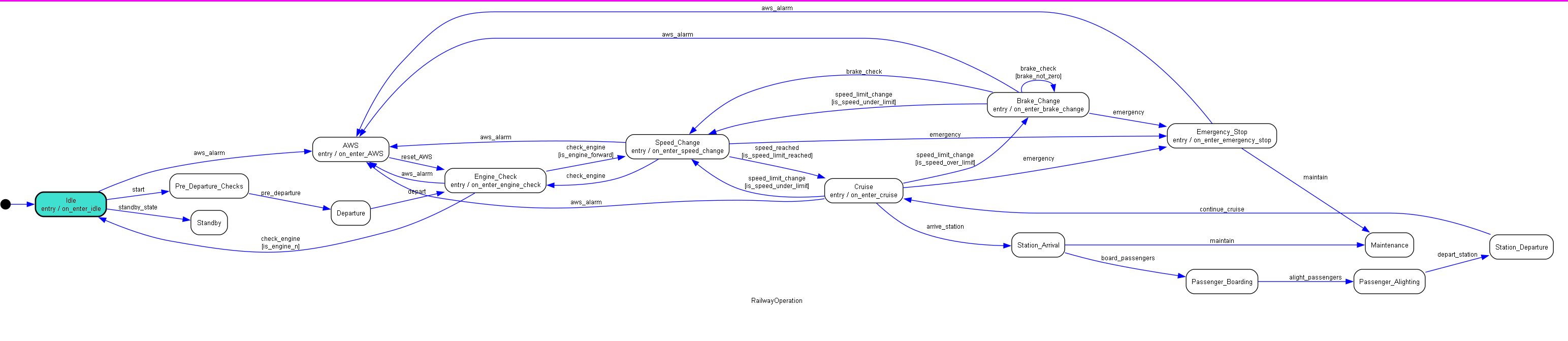
NoDiscomfortAtSignalPass +5 +5%

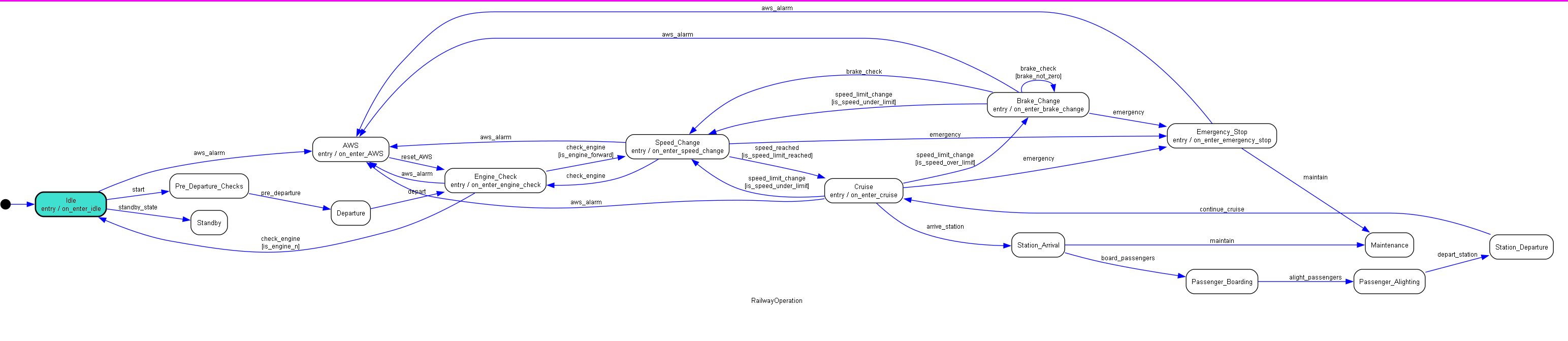
Display of information within simulation:



As the scoring system shows, the simulator is a ‘low reward environment’ which means any ML system involved will need to be built with a sparse reward structure, however, we intend to modify this at runtime. However, using these metrics, we observed how an initial exploratory system would use the simulator, which was recorded within another finite state machine:

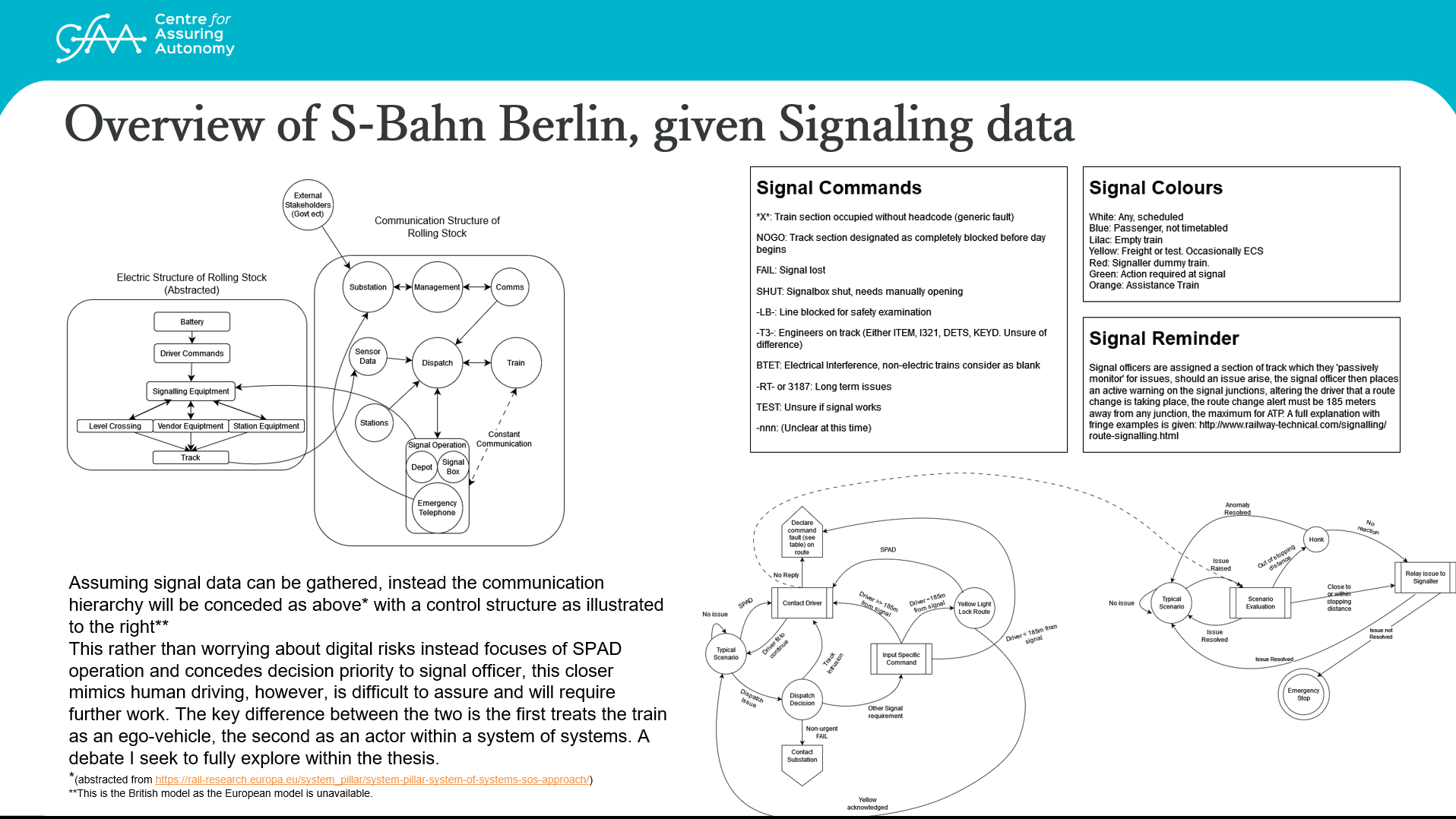
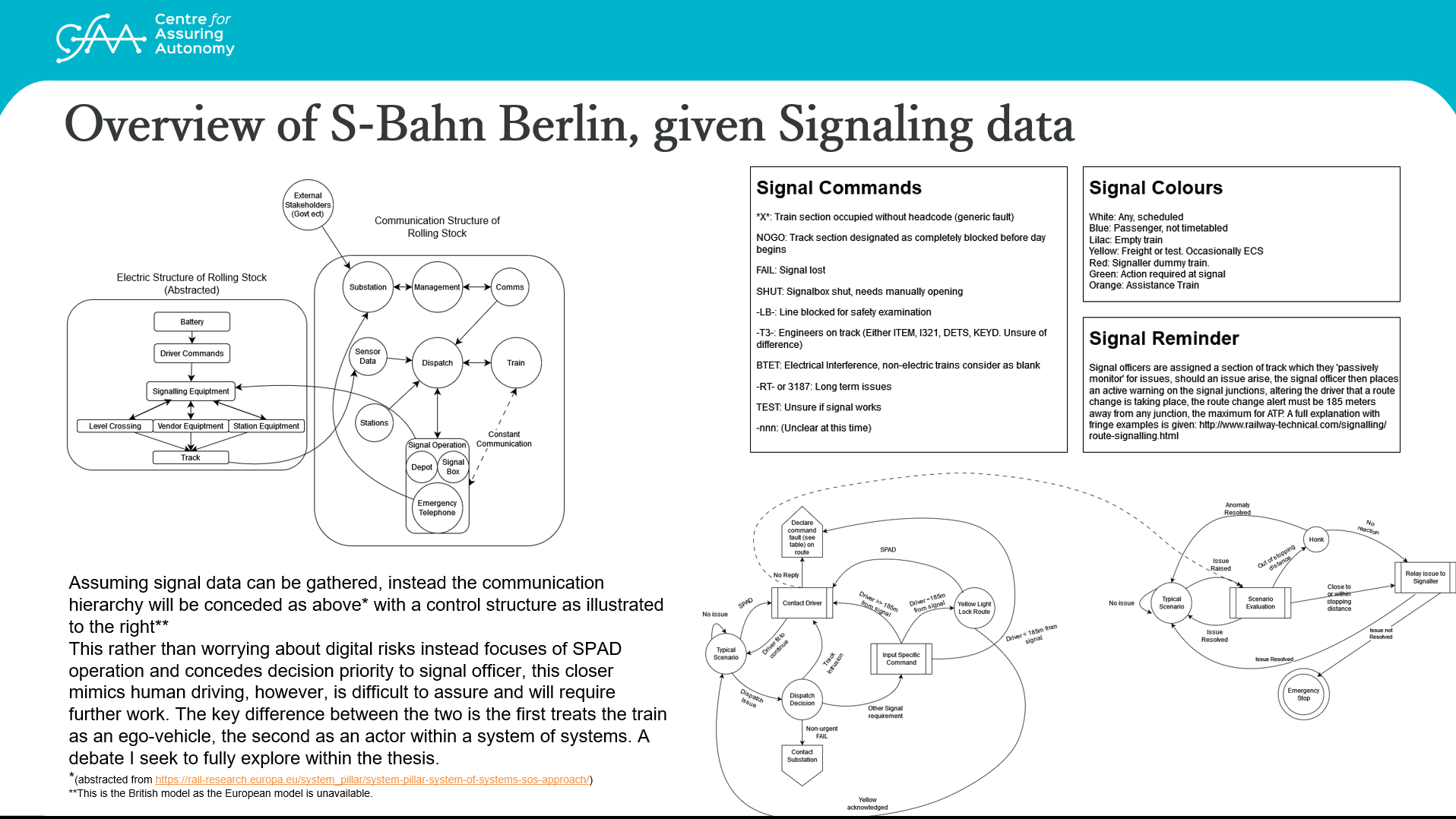


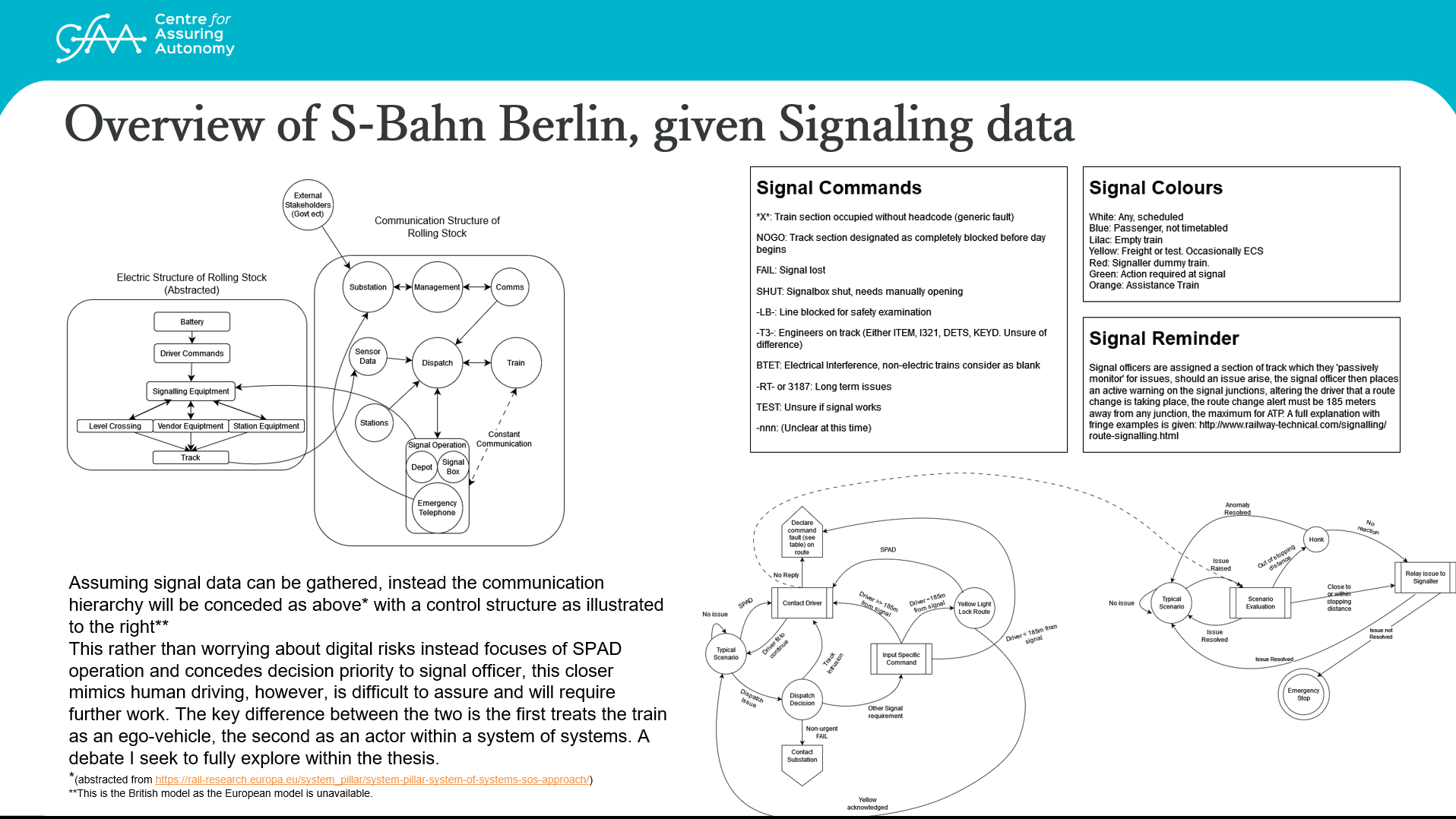




(See full version after citations)

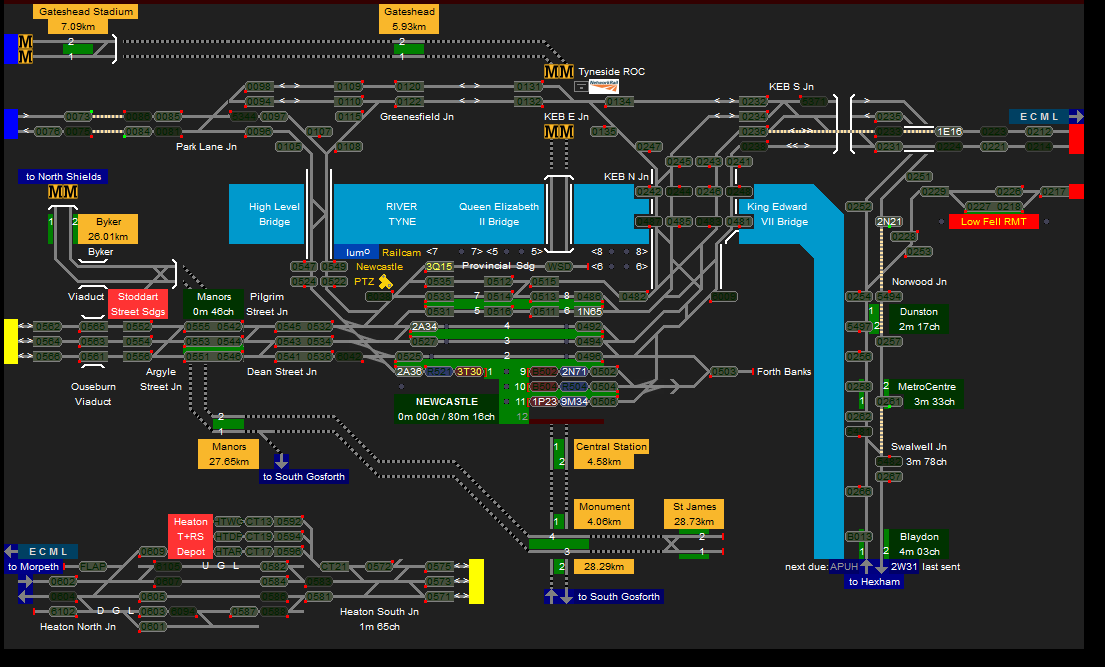
The state machine is based on the simulator, which is itself built upon the UK railway system.[12] The OpenBVE simulator is an open system which is maintained by mainly specialist fans, most of which are from either the UK or Japan, unfortunately I was unable to find much documentation on the S-bahn’s signalling system (other than it uses its own system, separate from ETCS/ERTMS.) However, UK signalling rules are that AWS (The requirement of a response) can happen at three intervals, at the position of a signal pass, at the request of a signal operator, or on a random interval between 30-90 seconds.[7] Signals have multiple states, which are explored within the following document:





[8]

For a summary, signals can either be blank (go) red (occupied) or have a failure state (dictated by the table.) Outside of ETCS, signal processing is done manually, with UK signal information being available live, provided by network rail under paid license and available to public through a RailCam subscription. A plugin has been developed for OpenBVE to grab signal information live (provided you have a subscription) and can be implemented into the simulator. A sample of the signalling information for the discussed route is as follows:

[10]

With the current route (NCL to Swalwell Jn) having the path that follows:

1N65 Red (This is the train currently at station, the simulator works in that you can simulate being 1N65, or can wait for 1N65 to pass) -> 0485 red (the next few signals are red due to the proximity of 2N21. 1N65 will need to wait for 2N21 to reach Norwood Jn before takeoff. 1N65 has a takeoff time in 5 minutes, meaning all surrounding signals will be red for 5 minutes. For simulation purposes then, we would either wait 5 minutes, or override by moving time forward.) With the rest being red. 0257 is green, allowing 2N21 to pass. This information is mostly inconsequential for the purposes of ML processing, however it is important for an argument made within step 4 of SACRED[9]

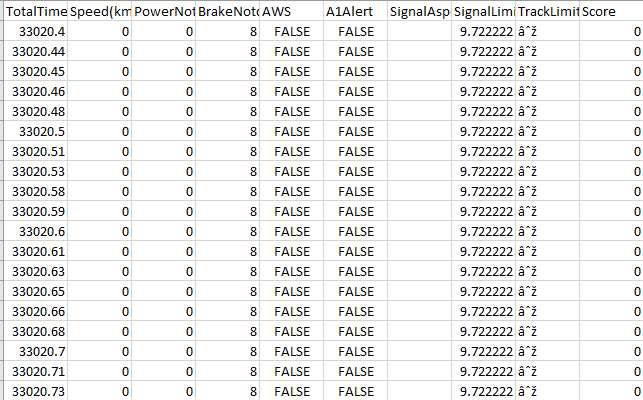
Scoring System Modifications:

The state machine was then redesigned as a GUI system for interpretation of the ML system, wherein the GUI highlights which state the ML is currently in and what outputs are possible, for example, standby, maintenance and emergency stop are end states and are highlighted red upon entry. When the ML is in use, current state is green with possible states are amber. Recording which specific state the ML is in help parse it’s decision making process, this also allows for different training for each state[6] The reward structure was modified in order to reward the system for being within the ‘cruise’ cycle, that being ‘cruise->station arrival->passenger alighting->station\_departure->cruise’ cycle. This is to dissuade the system from changing speed too often and instead promote conservation of speed, rather than fluctuation around the limit. This is to mimic how drivers actually operate systems.

Currently, the plan is to compare three separate approaches to railway operation, the first being a supervised training method, where several run-throughs of the simulator will be collected and stored as a CSV, which will be used for input/output training of a Bayesian model as well as an artificial neural network. The plugin I have developed for the purpose of datascraping is available on Git[11]

An unsupervised approach is described above within Scoring System Modifications where multiple simulators are to be ran simultaneously. This approach has not yet been tested due to the processing power required, however, a low-fidelity version using an abstract version of the simulator is under development.[2]

A snippet of the data is as follows, data is captured on each frame:



However, this is expected to be expanded after more testing. Even with the updated scoring system, the supervised method still results in negative scoring a majority of the time.

Future work:

While this work is all to support claims made within SACRED steps 1-5 and will help evaluate failure steps, allowing for limitation testing and the development of a Safe Operating Context, this also presents an opportunity for collaboration between the PhD and Fraunhofer IKS. After speaking with the team, they have shown interest in the development of a vision system which can assist in the information gathering/data scraping of the simulator. The simulator is an open environment where the track data is fully editable, including the addition of objects on the track through modification of the route CSV[13] Prior to my visit to the Fraunhofer offices, I plan to either develop or find a route with anomalies. Regardless, the simulator will provide sequential vision data needed for both track extraction and object detection.

This would help the development as the SACRED methodology as any additional justification that can be added to step 6 is overall a positive for the project. I also intend to discuss the approach taken within IKS’ work, regarding the System of Systems debate, to use as justification of step 3 of SACRED.

This work is also intended to be an academic paper, which I would appreciate collaboration to write. The work in progress paper is available at the link provided[14] The work currently specifies “the industry state of the art being a wilddash benchmark with a lack of sequential data, which is taken from an ego perspective” referring to the railSem19 data. This section can be removed if required, however I would like to request permission to reference the usage of RailSem19 in industry as I believe this important information for providing context.

References:

[1] <https://arxiv.org/abs/2403.12114>

[2] <https://github.com/leezer3/OpenBVE/tree/ScoreAdder>

[3] Specific modifications revolve around the access of the score for the purposes of training a ML system, modifications were made through email communication.

[4] Rail route: <https://openbve.wixsite.com/openbve/routes-for-the-uk> - Tyne Valley (Chosen 1. Due to the similarities with the route identified within SACRED step 2 and also due to personal familiarity)

[5] Rolling stock provided through personal connection, link to description: <https://www.openbve.net/en/fordon.php?kat=2&nat=40> This specific rolling stock was chosen due to it’s similarities with the S-bahn stock.

[6] Work in progress at this time. Currently the system records all data and rewards are given based on score and which state the system is within, however this is having unsatisfying results

[7] <https://ieeexplore.ieee.org/abstract/document/6316566>

[8] Part of a presentation given at previous alignment meeting, full document available upon request

[9] Insights from Railway Professionals: Rethinking Railway assumptions regarding safety and autonomy. Paper in review state, available upon request.

[10] <https://railcam.uk/diagrams/x_anydiag.php?di=Tyneside> (subscription required)

[11] <https://github.com/TortoiseFather/OpenBVE-ML-Plugins>

[12] <https://railsimroutes.net/libraries/uktrainsys/release_notes.html>

[13] <http://routebuilder.bve-routes.com>

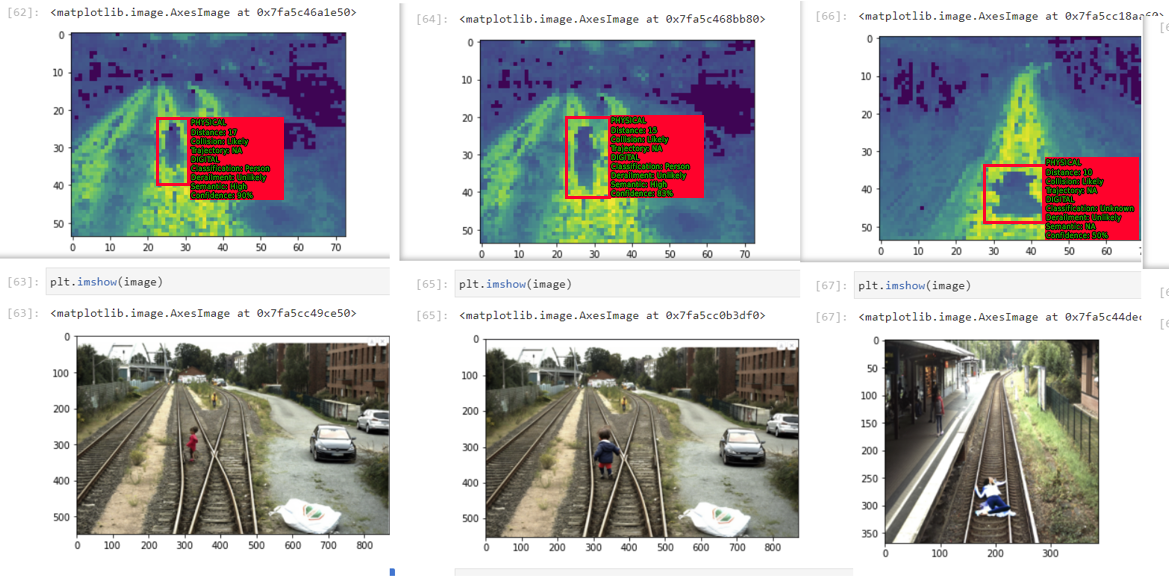
[14] <https://www.overleaf.com/read/jsrhkqfpfznw#78550d>

[15] <https://github.com/mlzxy/devit>

[16] <https://i.imgur.com/lYcJHpQ.png> - Presented at EDCC 2024

[17] <https://www.mdpi.com/1424-8220/19/11/2594>

[18]



Data from RailSem19, image modification provided through communication with Siemens Mobility, physical estimations <https://www.mdpi.com/2076-3417/12/3/1354> <https://github.com/dhananjaymenon/Vehicle-Collision-Detection-System> [https://github.com/gereon-t/trajectopy](https://github.com/gereon-t/trajectopy%20//%20Digital%20all%20from%20YOLOv8)

/ Digital all from YOLOv8

Full version of Finite State Machine

