Antitrust Policy Notice

- Eclipse Foundation meetings involve participation by industry competitors, and it is the intention of the Eclipse Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore important that attendees not participate in any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at Eclipse Foundation meetings and in connection with Eclipse Foundation activities are described in the Eclpse iFoundation Antitrust Policy available at https://www.eclipse.org/org/documents/Eclipse_Antitrust_Policy.pdf.

- If you have questions about these matters, please contact your company counsel, or if you are a member of the Eclipse Foundation, feel free to contact legal@eclipse.org.

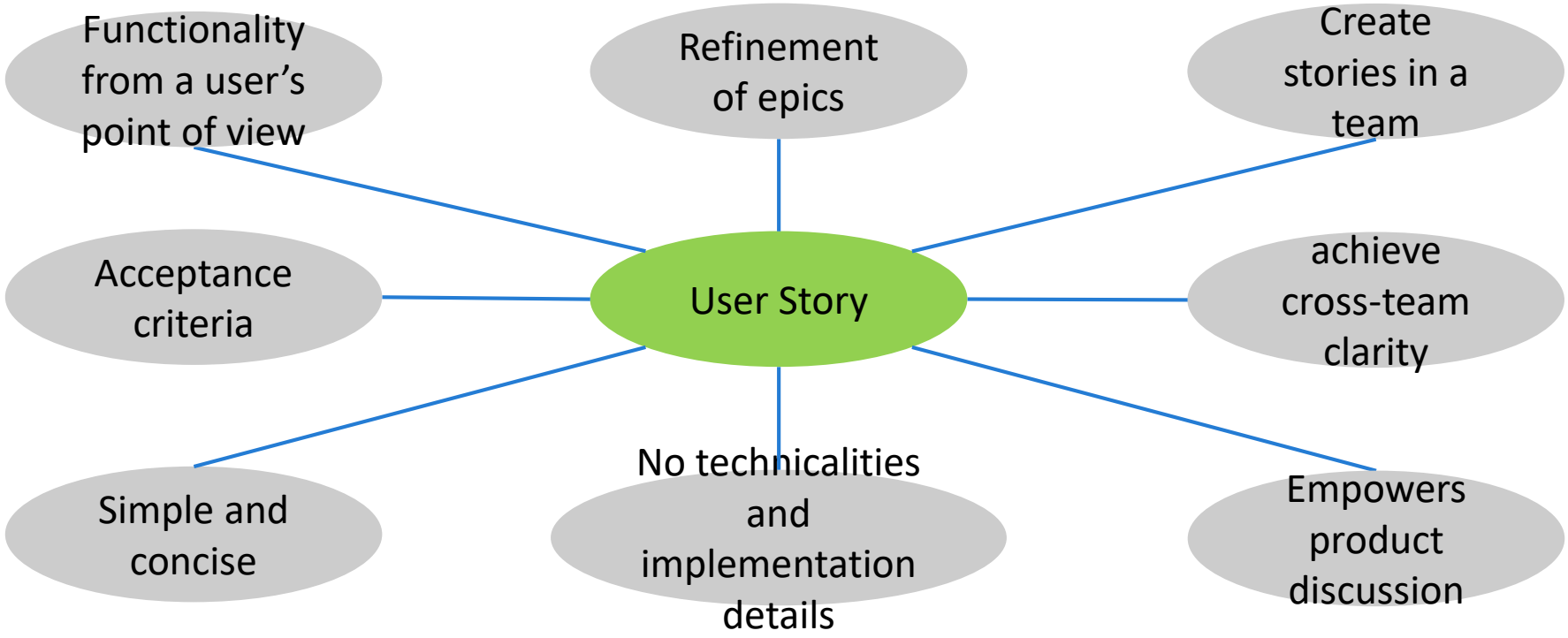
OPENPASS

STORY WRITING WORKSHOP
12.02.2020

AGENDA

Start	End	Requirement Team	High-Level Target	Requirement Team	High-Level Target
08:00	08:30	All	Get together		
8:30	9:15	Tuan	Goals of the Workshop, how to write User Stories and next steps in Tuleap		
9:15	10:15	All	examplary discussion with all: Continuous Integration		
10:15	10:30	All	Break		
10:30	12:00	Bosch Daimler TÜV SÜD VW?	crash configuration description & post crash calculation result csv table & folder structure	BMW ITK? Daimler	Code Quality Project directory
12:00	12:30	All	Presentation and Discussion of User Stories		
12:30	13:30	All	Lunch		
13:30	15:00	Bosch BMW Daimler ITK? VW?	PCM integration in OSI: - refactor GUI plugins wrt results - OpenSCENARIO extension for PCM - PCM spawn point - GUI Plugins for PCM simulation	ITK BMW Daimler? VW?	Observer
15:00	15:30	All	Presentation and Discussion of User Stories		
15:30	15:45	All	Break		
15:45	17:00	All	Detailed Release Planning V0.7		

HOW TO WRITE GOOD USER STORIES



Good user story:

As a regular user, I want to see a global top view visualization of the simulation run such that I have a better imagination of the simulated scenario. Acceptance criteria: Visualization which shows the moving agents in a road environment.

OVERVIEW HIGH-LEVEL TARGETS

Slide 6: Setup Continuous Integration

Slide 15: Setup Project Structure

Slide 22: Observer

Slide 28: Crash Configuration

Slide 35: PCM Integration

SETUP CONTINUOUS INTEGRATION

Overview:

- Setup cmake buildsystem
- Setup Conan
- Migration to GitLab
- Setup GitLab Wiki
- Setup Linux buildjob in GitLab CI
- Provide official releases to users
- Documentation
- Setup Windows buildjob in GitLab CI

SETUP CMAKE BUILDSYSTEM

As a developer, I want to have a Cmake build system set up. Therefore, the build system has to be set up for all components (Core + GUI).

Note: Qmake is already marked as deprecated and will be discontinued in future (Date?).

Acceptance criteria:

- Cmake build system is set up
- Qmake build system stays available in parallel
- Mark all Qmake / QT project files as deprecated
- Both build systems are working und Linux and Windows

SETUP CONAN

As a developer, I want to have the option to use Conan to collect and build all third party dependencies. This should be an alternative to manually collect and build the dependencies.

Acceptance criteria:

- Conan is set up to collect and build all third party dependencies
- The third party dependencies are available for the openPASS build

MIGRATION TO GITLAB

We as Architecture Committee want to move the project infrastructure to the Eclipse GitLab. Therefore, a project has to be created and the source code and all issues have to be moved to the GitLab project.

Questions that need to be clarified:

- Who will move the content?
- Will there be a Jenkins behind the Gitlab?
- When are we able to move?

Acceptance criteria:

- Source code is managed in the GitLab project
- Issues are all transferred and accessible in the GitLab project
- Links on the website are adapted to the GitLab links

SETUP GITLAB WIKI

As a user, I want to find information according to the project setup and development in the GitLab wiki. I.e. the setup guide and tutorial, which can currently be found as pdf in the repository shall be included in the wiki.

Note: later the coding guidelines shall also be described in the wiki.

Acceptance criteria:

- GitLab wiki is set up
- Setup guide for openPASS is included in the wiki
- Tutorial on openPASS simulation is included in the wiki

SETUP LINUX BUILDJOB IN GITLAB CI

As a developer, I want to have a build job that automatically builds the source code for the simulator and GUI and runs all tests under Linux. The build artifacts shall be saved and accessible for the committer(last 10 builds).

Known challenges:

- How do we include the third party dependencies (e.g. QT Libraries, protobuf, OSI, Boost)

Acceptance criteria:

- Build jobs are automatically triggered by new commits
- Build jobs can be manually triggered
- Simulator and GUI are built and tests are run under Linux
- The artifacts from the last 10 builds are stored and can be accessed

PROVIDE OFFICIAL RELEASES TO USERS

As a user, I want to get official releases delivered as binaries.

ToDo: What is an official release? Are there any requirements by the Eclipse foundation?

Acceptance criteria:

- Official releases are available for all users in a central location

DOCU

As a user, I always want to have the newest documentation built and available on the website (e.g. wiki). Therefore, a CI job should be triggered automatically and build the documentation. For official releases, the documentation shall automatically be made available on the website.

Acceptance criteria:

- CI job for building the docu is set up
- Automatic deployment of docu works for official releases

SETUP WINDOWS BUILDJOB IN GITLAB CI

As a developer, I want to have a build job that automatically builds the source code for the simulator and GUI and runs all tests under Windows. The build artifacts shall be saved and accessible for the committer (last 10 builds).

Known challenges:

- How do we include the third party dependencies (e.g. QT Libraries, protobuf, OSI, Boost)

Acceptance criteria:

- Build jobs are automatically triggered by new commits
- Build jobs can be manually triggered
- Simulator and GUI are built and tests are run under Windows
- The artifacts from the last 10 builds are stored and can be accessed

SETUP PROJECT DIRECTORY STRUCTURE

Overview:

- Concept for project directory structure
- Restructure the project directory

CONCEPT FOR PROJECT DIRECTORY STRUCTURE

As a developer, I want to have a new project directory structure.

Requirements:

- Follow best practices of open source
- Apply DRY principle on paths
  - or keep paths short - the old windows issue
  - no more openpass_this/openpass_that/algorithm_this/algorithm_this_and_that/…
- Reflect architectural changes
  - or simply moving stuff around
- Consider the extension of the source code with custom (non open source) components
- Concept also plans a space for contributions beyond source code (templates, python scripts, simulation configs)
- Discuss a clear seperation of documentation based on user perspective (Users versus Dev)
    - Compare and match to the different layers of documentation (see attachement)

A proposal from intech is availabe for further discussion (see attachment).

Acceptance criteria:

- A concept for the project directory structure is available

RESTRUCTURE THE PROJECT DIRECTORY

As a AC, we want to have the previously developed concept concerning project directory structure implemented. This story only restructures the folders and not files.

Todo: Refine story based on concept. This story might have to be split into several smaller stories, depending on the cencept.

Acceptance criteria:

- Directory structure is adjusted according to the concept
- Project can be built and simulations work as before.

CODE QUALITY

Overview:

- Definition of naming conventions
- Make source code compliant to code quality rules
- Proposal for code formatting

DEFINITION OF NAMING CONVENTIONS

As a developer, I want to have naming conventions defined for the whole project.

Requirements:

– Common definitions for file names, libraries
– Common definitions for naming of classes and variables
– Common definitions for usage of namespaces
– Further general rules
– Common rules for inline documentation

A proposal from intech is available for further discussion (see attachment).

Acceptance criteria:

- Naming conventions are defined
- An according template for new components is available
- The AC has accepted the naming convention
- The naming convetnions are documented in the wiki

MAKE SOURCE CODE COMPLIANT TO CODE QUALITY RULES

As an AC, we require all source code to comply with the code quality rules (i.e. naming conventions, coding rules).

Acceptance criteria:

- Source code complies with the established code quality rules.

PROPOSAL FOR CODE FORMATTING

As the AC, we want to have a clang format file for automated code formatting. This formatting will not be required right away but serve as a template for discussion.

Acceptance criteria:

- Clang format file is available for the project

OBSERVER

Overview:

- Requirement analysis and framework selection
- Implementation publish/subscribe mechanism
- Utilization of publishing mechanism
- Implementation of new observer
- Remove old observation framework

OBSERVER – REQUIREMENT ANALYSIS AND FRAMEWORK SELECTION

As AC, we want to have an analysis of requirements concerning the observer. Based on the requirements, also a framework shall be selected for future implementation (e.g. MQTT / Apache Kafka).

Exemplary requirements:

- Publishing / Logging of OSI messages
- License compatibility with Eclipse
- Broker can handle any topic without prior knowledge about it (generic!)
- Do past timestamps need to be stored? (e.g. log out timestamps before a critical scenario was detected)

Acceptance criteria:

- Requirements are identified and listed
- Frameworks for the publish subscribe mechanism are compared
- Selection has been made, fulfilling the requirements

IMPLEMENTATION PUBLSIH/SUBSCRIBE MECHANSIM

As a user, I want to have a publish subscribe mechanism implemented for later use for observations. Components do not need to be connected yet.

Acceptance criteria:

- Publish Subscribe mechanism is implemented
- Data can be published
- Data can be read through subscription
- Exemplary demonstration of publishing and subscribing is given

UTILIZATION OF PUBLISHING MECHANISM

As a developer, I want all components to be connected to the new observation framework. Therefore, all currently observed data shall in parallel be published to the broker.

Acceptance criteria:

- Data, which is written to the observation instance, is also published to the broker

IMPLEMENTATION OF NEW OBSERVER

As a user, I want to have an observer, which subscribes to the data in the broker and writes out the simulationOutput.xml. This should be written in parallel to the existing output. A comparison has to be made for validation of the functionality.

Acceptance criteria:

- A new observer is implemented
- The new observer subscribes to the relevant topics in the broker
- The output of the observer is the same as from the existing observer

REMOVE OLD OBSERVATION FRAMEWORK

As a developer, I want to have the old observation framework (including interface, network and components) to be removed.

Acceptance criteria:

- All content related to the old observation framework is removed

CONCEPT FOR CRASH CONFIGURATION / RESULT CSV

Overview:

- Extend CollisionDetector (calc crash config)
- Log crash configuration à la Wagstrom et al./OSCCAR
- Implement a collision factor/ coefficient of restitution k
- Extend dynamics collision for post-crash behaviour
- Export simulation results (cyclics) sorted by agents
- Enable folder structure for variations of PCM cases

EXTEND COLLISION DETECTOR (CALC CRASH CONFIG)

- As a user I want the CollisionDetector to calculate the crash configuration (three collision angles & two velocities)

- Acceptance criteria:
  - The extact time point of first contact is detected as collision.
  - The core module „EventDetector“ is extended with this functionality.

LOG CRASH CONFIGURATION À LA WAGSTROM ET AL/OSCCAR

- As a user I want to obtain the crash configuration from CollisionDetector, written to a result file. As published in the ESV paper by Wagström et. al. three angles + two velocities are sufficient to determine any crash conifguration.

- Acceptance criteria
  - The geometric relative positions of the collision partners are evaluated in order to obtain the angles and the velocities at the time of collision are logged.
  - The resulting collision parameter („crash configuration“) are written to the „event section“ of the simOutput.xml.

IMPLEMENT A COLLISION FACTOR / COEFFICIENT OF RESTITUTION K

– As a user, I want to have a impact calculation based on energy and momentun which allows to determin a „delta-v“ and an according post crash state of the collision opponents. This impact can be parametrized with a „k factor“ describing the type of crash, i.e. continuously shifting from plastic to elastic collisions.

– Acceptance criteria
  – collision factor can be configured in the GUI, similar to the penetration time.
  – Core can import and transfer impact parameter to applicable module
  – Calculate crash including collision factor (up to dv)

EXTEND DYNAMICS COLLISION FOR POST-CRASH BEHAVIOUR

– As a user, I want the dynamics after a collision (eg from impact calculation) transferred to dynamics module to calculate the post-crash trajectory of the collision partners.

– Acceptance criteria:
  – The post crash behaviour is determined by the impact calculation.
  – After the collision, the control over the dynamics is given back to the vehicle dynamics model.

EXPORT SIMULATION RESULTS (CYCLICS) SORTED BY AGENTS

- As a user, As a user I want to have the ability to write the cyclics as a csv-table sorted by agents (e.g. pcm-/highd-format).

- Acceptance criteria
  - The cyclics are available as a csv-table sorted by agents (e.g. pcm-/highd-format).

ENABLE FOLDER STRUCTURE FOR VARIATIONS OF PCM CASES

– As a user I want a folder structure for results files which allows me to store the different PCM runs (single experiments) as well as variations of a single experiment in an appropriate way.

– Acceptance criteria

- The csv data is split in runs and/or experiments and stored in a logic folder structure.
- The GUI plugins may process the csv data conatined in such folder structure.

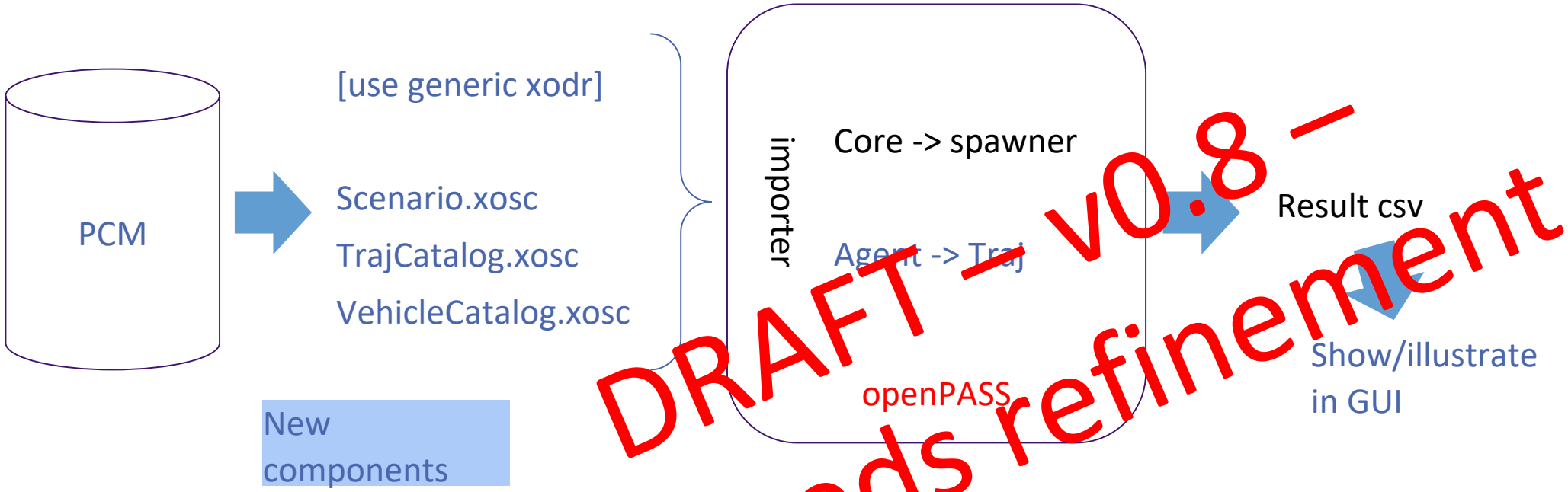
PCM INTEGRATION

Overview:

- Show results in GUI
- Fill scenario.xosc from PCM input
- Fill TrajectoryCatalog.xocs from PCM input
- Fill VehicleCatalog.xosc from PCM input
- Add PCM Agent TrajectoryFollower

DRAFT – v0.8 – needs refinement

PCM INTEGRATION



SHOW RESULTS IN GUI

– As a user I want to have a GUI (eg as additional plugin) which shows/illustrates the [csv] result output

– Acceptance criteria
  – Histogram & timeplots

– (see PCM evaluation plugin)

DRAFT – v0.8 – needs refinement

FILL SCENARIO.XOSC FROM PCM INPUT

– As a user I want to have a scenario.xosc file from the PCM input.

– Acceptance criteria
  – If value not given by PCM, use default values

DRAFT – v0.8 – needs refinement

FILL TRAJECTORYCATALOG.XOSC FROM PCM INPUT

– As a user I want to convert the pcm trajectory into the TrajectoryCatalog.xosc

– Acceptance criteria
  – If value not given by PCM, use default values

– NB: vehicle will use „input-trajectory“ as a path to follow (incl vehicle dynamics)

DRAFT – v0.8 – needs refinement

FILL VEHICLECATALOG.XOSC FROM PCM INPUT

- As a user I want to have (at least) sizes & weight of involved vehicles from PCM input

- Acceptance criteria
  - If value not given by PCM, use default values

DRAFT – v0.8 – needs refinement

ADD PCM AGENT TRAJECTORYFOLLOWER

- As a user, I want a trajectory follower that follows the PCM Input as good as possible imposing vehicle dynamics.

- Acceptance criteria
  - The agent has a PID controller to follow the trajectory.

DRAFT – v0.8 – needs refinement

MODULAR DRIVER - ARCHITECTURE

As a user, I want to

- …have a driver with detailed, explicit submodels for cognitive driver behavior model;
- …use static system config to describe agent with modular driver;
- …use stochastics to determine initial driver model parameters

Acceptance criteria:

- The driver architecture consists of detailed, explicit submodels.
- The driver can be configured via the static system config.
- The driver parameters are stochastically drawn.

MODULAR DRIVER - SENSOR

As a user, I want to

– …have a driver sensor with a container structure for driver sensor data as well as driver internal signals, so this submodel obtains ground truth wrt other agents and world.

Acceptance criteria:

- The sensor obtains the ground truth needed for the subsequent submodels of the driver model.

MODULAR DRIVER - INFORMATION ACQUISITION

As a user, I want to

– …have a driver that includes a model for human perception in cognitive process; see parameter: “Avert view time”; this should be a simple model, i.e. first implementation.

Acceptance criteria:

- This submodel filters the ground truth, modeling human perception.
- The degree of information acquisition can be set by the parameter „avert time“

MODULAR DRIVER - MENTAL MODEL

As a user, I want to

– …have a driver that stores his information and extrapolates road evolution and positions/behavior of other agents (e. g. if not visible anymore); this should be a simple model, i.e. first implementation.

Acceptance criteria:

- The driver model stores the information available to him in a mental model.

MODULAR DRIVER - SITUATION ASSESSMENT

As a user, I want to

– …have a driver that determines relevant agents and calculates relative indicators such as THW, THW etc. based on the information available in the mental model; this is a driver internal criticality assessment.

Acceptance criteria:

- The driver has a submodel which calculates relevant metrics/indicators.

MODULAR DRIVER - ACTION DEDUCTION

As a user, I want to

– …have a the driver that is capable of following the road, of car following behavior from standing still up until breaking in critical situations, lane changes based on SUMO model. This incorporates reactions to traffic infrastructure, e.g. speed limit, traffic lights.

Acceptance criteria:

- The driver follows the road or other agents in front of him.
- The driver conducts lane changes like the accoridng SUMO model.
- The driver reacts to traffic infrastructure and traffic rules.

MODULAR DRIVER - ACTION EXECUTION

As a user, I want to

– …have a driver which decisions are provided to the dynamics in terms of longitudinal and lateral signals by the submodel “action execution”.

Acceptance criteria:

- The driver provides his decisions in terms of brake and accelerator pedal positions and steering wheel angle to the dynamics module.

CRITICALITY

As a user, I want

– …a component that is able to evaluate the criticality of a situation, by means of, at least, TTC (time to collision), THW (time headway), and speeding (velocity above speed limit). Flexible controllability with logging group criticality.

Acceptance criteria:

– The criticality is evaluated for each participant within the simulation
– Logging group criticality is available
– The component allows for both
  - global, retrospective analysis of the criticality (after a simulation, e.g., as part of a post-processing)
  - criticality assessment for a specific participant (e.g. ego vehicle) in each time step of the simulation
– The component is ready to communicate with other components (e.g. modular driver – action deduction)

IMPLEMENTATIONS FOR 0.7

- Collision calculation (Daimler)
- Modular driver (Daimler & Bosch)
- Spawner + Urban environment (BMW)
- Observer (BMW)
- Criticality (open)(Daimler & Bosch)
- Project directory structure (Wunsch 0.7)(ITK)
  - Cmake review pull request, eventually first adjustments (BMW)
- Concept for refactoring GUI plugins (Daimler)
- Concept for PCM integration (Daimler)