Discussion group:
Concept Project Proposal for ASAM OpenSCENARIO

24-Jan-2019

Gil Amid – Foretellix Ltd
Yoav Hollander - Founder and CTO, Foretellix
blog.foretellix.com

foretellix
verifying autonomous systems

Agenda

1. Introduction – What is this meeting about? Future logistics

2. High-level maneuver descriptions and how they relate to a DSL - Yoav Hollander

3. Using a DSL to unify the treatment of several features - Yoav Hollander

Agenda

1. Introduction – What is this meeting about? Future logistics
2. High-level maneuver descriptions and how they relate to a DSL - Yoav Hollander
3. Using a DSL to unify the treatment of several features - Yoav Hollander

foretellix

Introduction

- **What is this meeting about ?**

- **Allow me to introduce: Speculative execution**

“Speculative execution is an optimization technique where a computer system performs some task that may not be needed. Work is done before it is known whether it is actually needed, so as to prevent a delay that would have to be incurred by doing the work after it is known that it is needed. If it turns out the work was not needed after all, most changes made by the work are reverted and the results are ignored.”

This Discussion group

- "While this discussion group was spawned after ASAM's OpenSCENARIO proposal workshop**, it is not yet the official ASAM workgroup for the concept project.** The way to think about these set of meetings is purely voluntary, no obligation. The assumption is that the conclusions, learning and understandings out of these discussions may flow into the concept project proposal, and later discussions at the concept project. **The discussion group is open to any interested parties."**

- "**The discussion group will in a first step focus on the understanding of the scope of F008 (High-Level Maneuver Descriptions).** Several alternatives have been proposed as potential approaches to support F008. As discussed in the Proposal Workshop, they are not mutually exclusive. **In a first step, we will discuss a DSL approach (Domain Specific Language), and how a DSL could unify the treatment of F008 and some of the other features on the list.**"

foretellix

Future logistics

- Currently suggested frequency – once in two weeks
- If you wish to bring up and present a topic for discussion – please send me a note
- I (Gil) will facilitate the agenda, and schedule the meetings accordingly
  - Presentations will be archived in a TBD mechanism.
- I would like to thank ASAM for the logistics support and teleconferencing faculties
- May we have an interesting discussion group ☺

foretellix

Agenda

1. Introduction – What is this meeting about? Future logistics
2. High-level maneuver descriptions and how they relate to a DSL - Yoav Hollander
3. Using a DSL to unify the treatment of several features - Yoav Hollander

F008: High level maneuver descriptions

- Write at high level, specify subset of params, automatic transform to fully-concrete
  - "The standard shall provide a method for maneuver descriptions on a **higher level of abstraction** … This shall contain only the logical description of scenarios with as **few parameters as possible** … automatically **transformed into the detailed description …"**
- Actual behavior depends on the high-level scenario descriptions *and* on the models
  - "This might include, that one part of the maneuver is undefined and shall be **handled by the driver- or traffic- models** during simulation"
- Use (a) a data model, (b) a DSL or (c) a general-purpose language
  - If (b) or (c), use an OO description for high-level objects (with attributes and methods)
  - Have a UML model for all that
- My proposal
  - Define a *declarative* DSL defining objects with attributes, methods *and constraints*
  - Models and scenarios are objects in this language
  - Actual procedural execution done by (multiple) general-purpose languages
  - Create a UML model from the declarative language

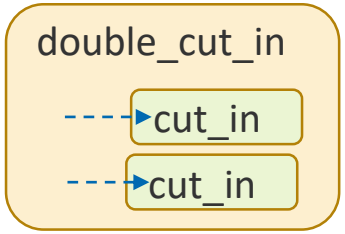
foretellix

Composition

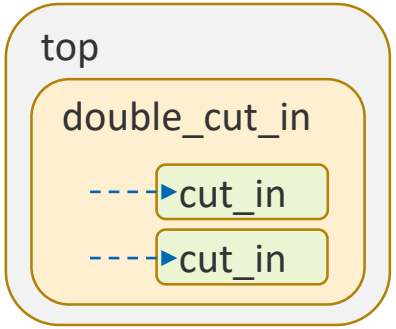
```
// Define a scenario
scenario cut_in {
      side: [left, right];  // Parameter
      do {…};               // Behavior
};
```

cut_in

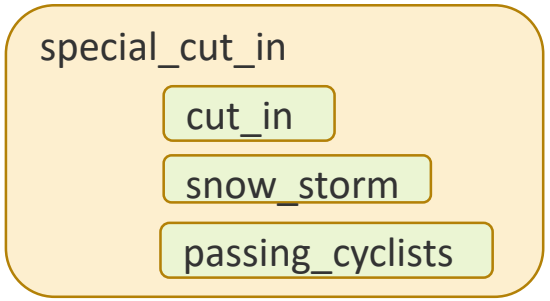
```
// Compose using "serial"
scenario double_cut_in {
    do serial {
        c1: cut_in {.side == left};
        c2: cut_in {.side == right};
    };
};
```



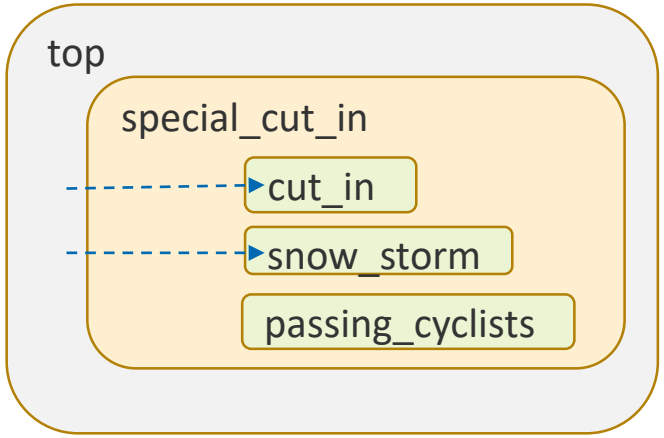

```
// Define a test
scenario top {
    do double_cut_in;
};
```




```
// Compose using "mix"
scenario special_cut_in {
     do mix {
        c: cut_in;
        s: snow_storm;
        p: passing_cyclists;
     };
};
```

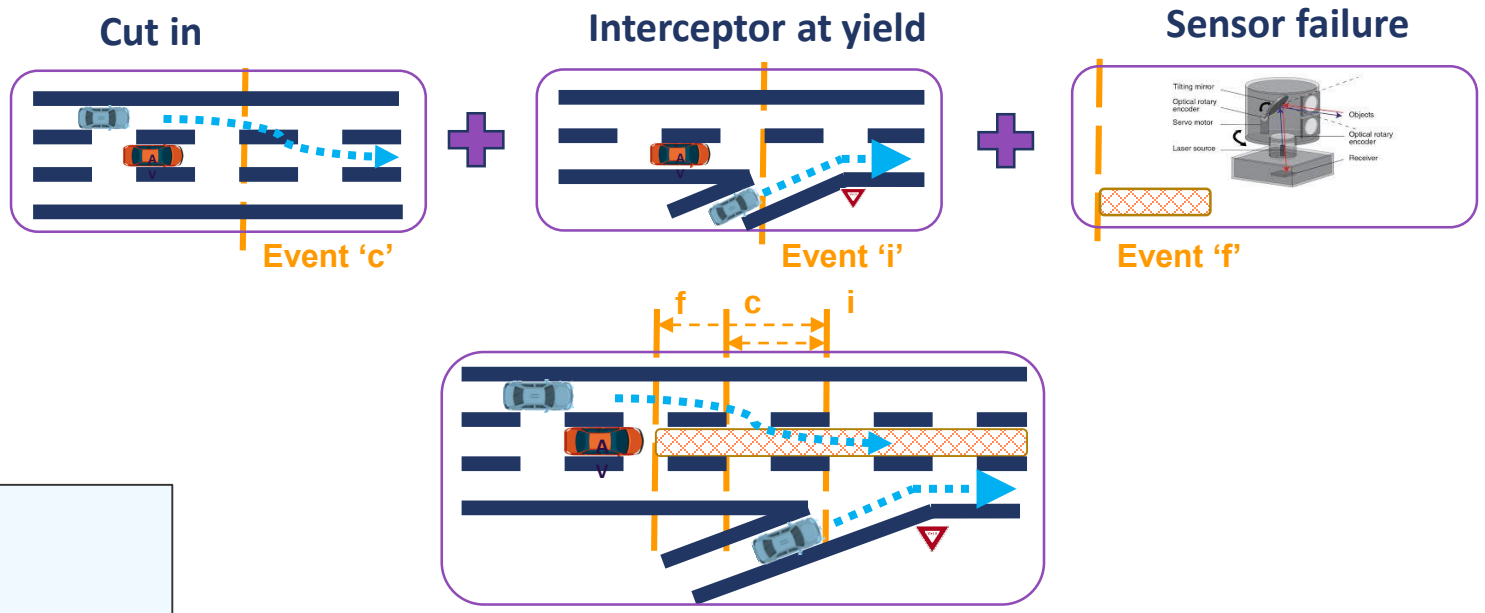



```
// Define a different test
scenario top {
       do special_cut_in {
             .c.side == left;
             .s.strength in [7..10]
       };
};
```



foretellix

Mixing and synchronization

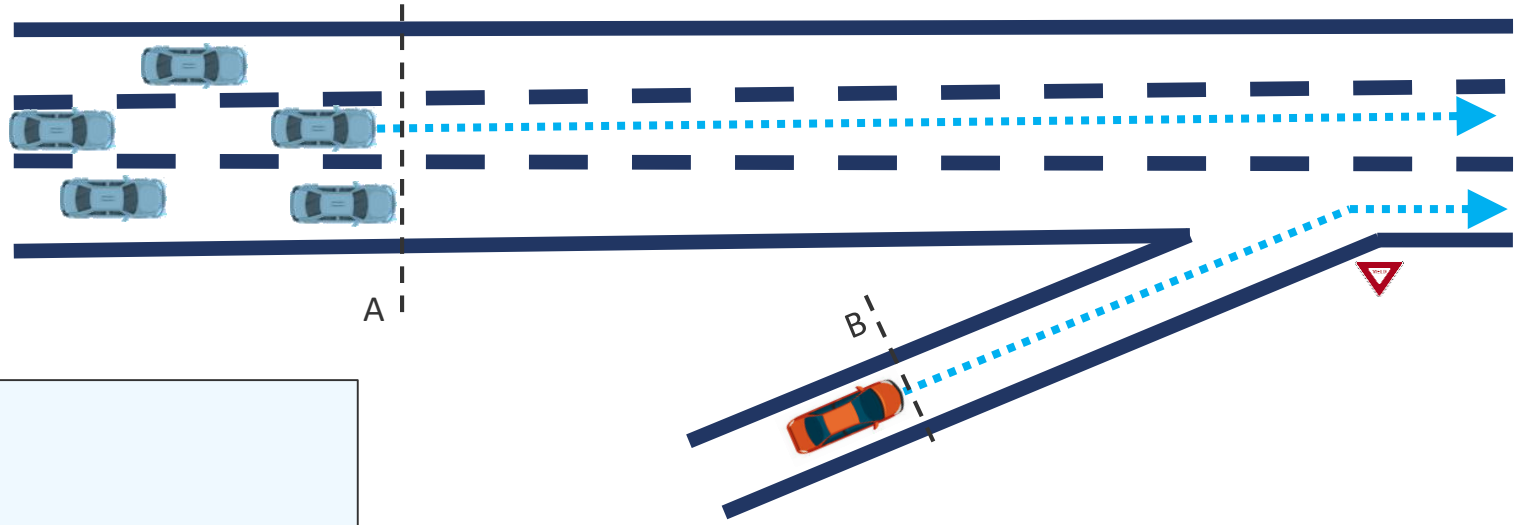



```
scenario mix_multi is {

   do mix {
        c: cut_in;
        i: interceptor_at_yield;
        f: sensor_failure;
   };

    +synchronize(c.change_lane, i.e_traverse.enter,  [-2..2] * second);
    +synchronize(f, i.e_traverse.enter,  [-0..0.5] * second);
};
```


foretellix

Handling synchronization

The car group should reach A when the ego reaches B




```
scenario  group_approaching is {

   do mix {
        g:  car_group.drive(main_road);
        e:  ego.drive(side_road);
   };
    ...

   +synchronize(g.arrive_at_A, e.arrive_at_B, [-1..1] * second);
};
```


foretellix

Agenda

1. Introduction – What is this meeting about? Future logistics

2. High-level maneuver descriptions and how they relate to a DSL - Yoav Hollander

3. Using a DSL to unify the treatment of several features - Yoav Hollander

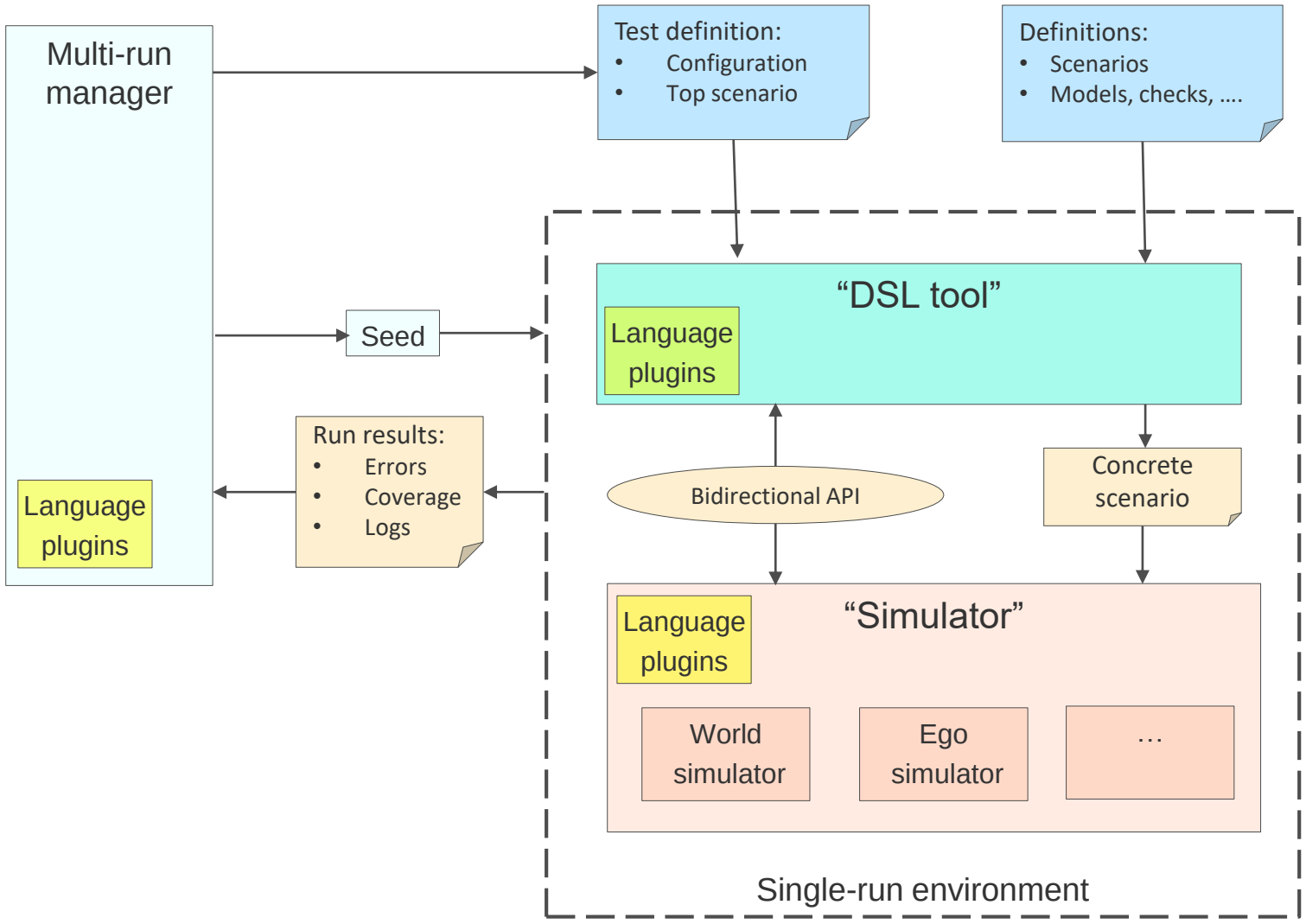
foretellix

How features may relate to DSL constructs

- The table below tries to show how a DSL might help encapsulate / simplify the implementation of the various features.
- Note: This does not make the problems go away – it just makes the features *somewhat* easier to implement, and puts them under a unified framework.
- Hope to explain this better in a subsequent workshop

Feature	Language constructs which may simplify / encapsulate this feature
F001: Maneuver model	Composition, operators (serial, mix etc.), constraints, …
F002: Driver model	Models are agents (with attributes, constraints, hooks to external languages, …)
F003: Traffic model	Models are agents (with attributes, constraints, hooks to external languages, …)
F004: Weather model	Models are agents (with attributes, constraints, hooks to external languages, …)
F005: Environment event model	All the above + event definitions and synchronization
F006: Vehicle dynamics model	Models are agents (with attributes, constraints, hooks to external languages, …)
F007: Parameter stochastics	Constraints (e.g. “keep speed == <some-gaussian-distribution>”)
F008: High-level maneuver descriptions	All of the above + synchronization constraints and generic hooks to external languages
F009: Replay of recorded scenarios	Trajectories as first-class language objects
F010: Automatic parameter calculation	Constraints (e.g. “keep distance == speed * time”)
F011: Automatic metadata for parameters	Ability to extend agents / scenarios in named modules (adding attributes, constraints etc.)
F012: Localization	Ability to extend agents / scenarios in named modules (adding attributes, constraints etc.)

A possible block diagram and suggested terminology



foretellix

Drilling into the DSL tool