

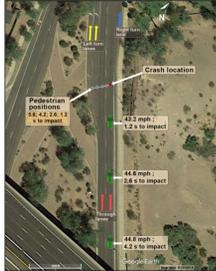
# T2AD: 自动驾驶的测试与测试控制表示语言

董乾, 杨光, 薛云志, 孟令中, 师源, 丁明明, 司宏远

董乾, 13811152430, dongqian@iscas.ac.cn

## 背景介绍

- 自动驾驶广受关注, 包括Tesla、Waymo、Uber和Baidu
- 致命事故表明目前的自动驾驶仍然不安全
- 真实世界中的车辆测试成本巨大, 需要仿真测试
- 固定的测试用例无法充分测试自动驾驶的安全性



## 研究意义

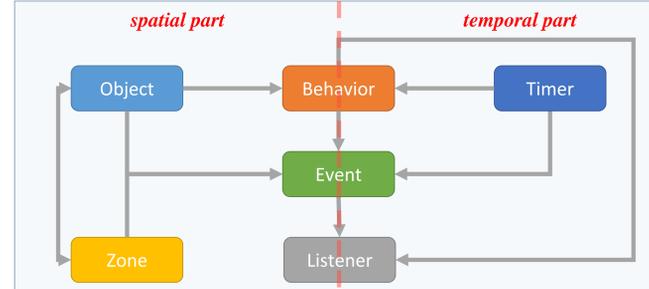
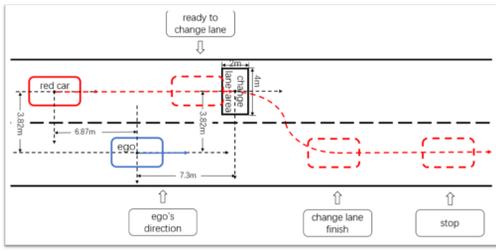
- 自由灵活表示场景中物体的空间和时间信息
- 整合TORCS、Autoware、Carla以及Airsim等模拟器
- 构建自动驾驶测试的顶层测试框架
- 支撑基于模型的测试验证提供, 提供多样性可定制的测试数据集

## 相关工作

- 交通场景模型
- Schuldt[1], 交通场景的五层模型, 道路、设施、施工、物体、环境
- Elrofai等人[2]和Rocklage等人[3], 场景中元素的定义, 行为动作最重要
- Thorn等人[4]和Eilers等人[5], 基本驾驶操作分类
- Allen[6]和Bach等人[7], 不同行为的时序关系
- 语言设计研究
- ISO 26262标准提出功能、逻辑和具体场景三个层次
- Kearney等人[8]、Geoscenario[9]和Parascom[10]介于逻辑层和具体层
- 功能性语言SCENIC[11], 仅限于生成初始静态测试场景

## 我们的贡献

- 通过关键动作分解和行为时间调度构建T2AD
- 支持测试判据注册
- 结合UE4模拟器实现测试用例, 体现出T2AD对于场景控制和测试的能力

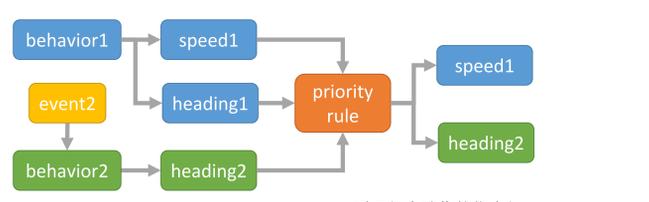


## 动态测试场景模型

- 空间属性
  - 确定物体Object位置
  - 区域Zone描述物体形状, 标记感兴趣的空间
- 时间属性
  - 时钟Timer, 用来记录测试用例运行的时间
  - 行为Behavior, 描述动态场景中物体具有速度进行运动
  - 事件Event, 表达行为的发生与否、物体与区域的位置关系
  - 监听器Listener, 刻画固定的“事件-行为”模式
- 物体和区域在空间上的描述可初始化测试场景
- 时钟、行为、事件、监听器用于构建运动过程

## 动作行为模型

- 速度方面, 需要描述速度的数值以及速度变化的情况
  - 定速巡航、加速、减速、刹车、停车
- 朝向方面, 结合道路信息, 简化描述物体的朝向变化
  - 保持道路、向左换道、向右换道、左转、右转、调头、直行
- 描述与空间中除道路的区域产生关系的变化
  - 进入区域、穿过区域、离开区域
- 原子动作行为优先级: 行为发生的越晚, 优先级越高



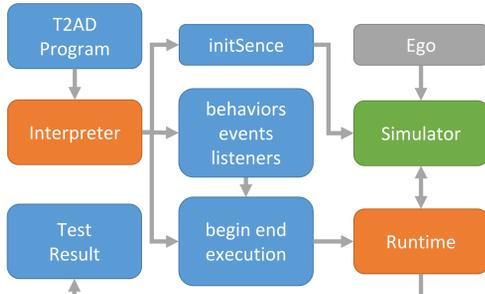
## 重要的操作符

- behaviorOperator, 用于控制物体的方向、速度以及与区域的空间交互, 包括headingOperator、speedOperator和spatialOperator
- sequenceOperator, 用于表示行为动作的时序关系或前提条件事件
- eventOperator, 用于物体有关的事件是否发生
- listenerOperator, 用于声明Listener类的实例和控制实例的监听状态
- oracleOperator, 用于构建测试用例中的判断准则

## T2AD代码结构

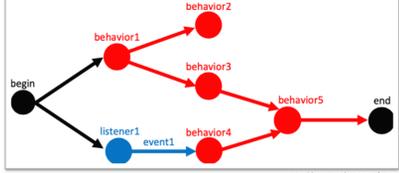
- initSence, 初始化, 创建所需对象
- behaviors, 声明行为, 注册所需测试预言
- events, 声明事件, 注册所需测试预言
- listeners, 创建监听器
- begin和end, 声明开始和结束时的行为
- execution, 声明运行过程中行为的时序关系

## T2AD架构



## T2AD工作流程

- T2AD脚本, 面向对象的语言, 提供数据类型、对象类型、操作符要素
- 解释器interpreter, 规定T2AD程序的结构, 将测试用例解析为三部分的结构化数据
- 执行器runtime, 初始化场景, 实现物体动作行为的控制, 并进行测试评判返回测试结果



## 数据类型

- Boolean, 用于表示逻辑值
- Number, 用于表示整数
- Scalar, 用于表示浮点数, 可用作空间尺度或一维坐标 (m)
- Vector, 用于表示二维坐标 (X, Y), 形式为X@Y, X、Y是Scalar
- Heading, 用于表示物体的方向 (deg)
- Speed, 用于表示物体的速度 (m/s)
- Time, 用于表示时间度量 (s)
- StatusEnum, 用于表示行为状态, 是一种枚举类型, 取值为waiting、running、finished、failed

## 对象类型

对象	属性	类型	含义
Zone	shape	Vector List	边界点集
	heading	Heading	区域方向
	parent	Vector	绑定物体
Object	position	Vector	物体位置
	bounding	Zone	包围盒
	speed	Speed	速度
	heading	Heading	物体方向
Timer	curTime	Time	当前时间
	maxTime	Time	最大时间
Behavior	action	behaviorOperator / Behavior List	动作序列
	status	Boolean	动作状态
Event	criteria	eventOperator	事件标准
	result	Boolean	事件结果
Listener	trigger	Event	触发事件
	action	Behavior	触发行行为
Placeholder	status	Boolean	监听状态
	action	Behavior	替代对象
TestOracle	critierias	Behavior / Event	测试准则
	targets	Boolean List	测试预期
	results	Boolean List	测试结果

- behaviorOperator
  - speedOperator := Object accelerate [to Speed]
  - | Object decelerate [to Speed] | Object reverse
  - | Object cruise [(to Scale) | (during Time)]
  - | Object brake [to Scale] [during Time] | Object stop
- headingOperator := Object changelane (left | right) | Object keepLane | Object goStraight | Object turn (left | right | back)
- spatialOperator := Object enter Zone | Object cross Zone | Object exit Zone

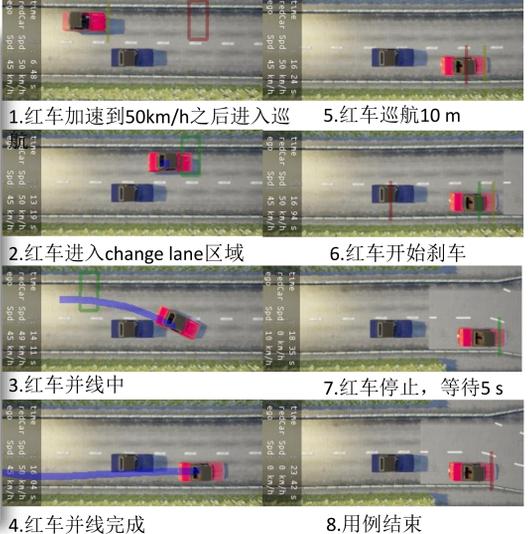
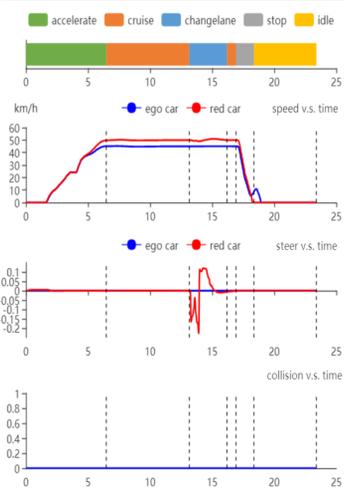


## T2AD测试脚本

```

initSence{
  egoOracle:TestOracle = TestOracle
  ego:EgoCar = EgoCar at 846.90 @ 125.90
  changeLaneArea:RectZone = RectZone with width 4, with height 2, with attach to ego, with offset by -3.82 @ 7.3
  redCar:Car = Car relative to ego, offset by -3.82 @ -6.87
  x:Placeholder = Object
  timer:Timer = Timer with maxTime 300
}
behaviors{
  behavior redCarAcc:
    redCar accelerate to 13.89
    redCar cruise
  behavior redCarStop:
    redCar changeLane right
    redCar cruise to 10
    redCar stop, register to egoOracle
    timer idle 5
  behavior collisionTrigger:
    log "collision " + x
}
events{
  timeout:Event = Event timer isTimeOut, register to egoOracle target False
  changeLaneTrigger:Event = Event redCar isEnter changeLaneArea
  egoCollision:Event = Event ego isCollide x, register to egoOracle target False
}
listeners{
  collisionListener:Listener = listen egoCollision take collisionTrigger
}
begin{
  timer start
  collisionListener open
}
end{
  collisionListener close
  egoOracle analyze
}
execution{
  redCarAcc start begin
  redCarStop waitfor changeLaneTrigger
  end after redCarStop
}
    
```

## 运行结果



- 红车运动过程
  - 0 s--6.47 s, 加速至50 km/h
  - 6.47 s--13.16 s, 定速巡航
  - 13.16 s, 进入并线区域后并线到右侧车道
  - 16.19 s--16.93 s, 定速巡航10 m, 随后刹车
  - 18.36 s, 完全停止, 最后静止5 s

## 实验数据分析

- 红车实现加速、巡航、换道、巡航和停车
- 红车的速度控制达到目标速度50 km/h
- 红车的转向曲线展示红车执行并线操作 (负数表示右转, 正数表示左转)
- 碰撞数据展示主车没有发生与红车的碰撞

## 总结

- 自动驾驶领域的测试与测试控制语言T2AD
- 在语义层面非常简洁地描述测试用例
- 极大丰富现有自动驾驶的测试数据集
- 极大促进基于模型测试验证的发展

主要参考文献  
 1. Schuldt F. Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen[D]. 2017.  
 2. Elrofai H, Paasikallio J, de Geller E, et al. Scenario-based safety validation of connected and automated driving[C]. 2018.  
 3. Rocklage E, Kraft T, Karata A, et al. Automated scenario generation for regression testing of autonomous vehicles[C]. 2017.  
 4. Thorn E, Kimmel S C, Chaka M, et al. A framework for automated driving system testable cases and scenarios[R]. 2018.  
 5. Eilers M, Griffin S, Kähler D, et al. Catalogue of basic driving manoeuvres and associated task distributions[R]. 2017.  
 6. Allen J F. An Interval-Based Representation of Temporal Knowledge[C]. 1981.  
 7. Bach J, Otten S, Sax E. Model based scenario specification for development and test of automated driving functions[C]. 2016.  
 8. Kearney J, Willemsen P, Donkers S, et al. Scenario languages for driving simulation[J]. 1999.  
 9. Quero R, Berger T, Czarnecki K. GeoScenario: An open dsl for autonomous driving scenario representation[C]. 2016.  
 10. Majumdar R, Mather A, Pirron M, et al. Parascom: A Language and Tool for Testing Autonomous Driving Systems[J]. 2019.  
 11. Freeman D J, Drossi I, Ghosh S, et al. Scenic: a language for scenario specification and scene generation[C]. 2019.