

OpenCRG

- ASAM OpenCRG定义了一种描述路面的文件格式
- 采集数据

From time-consuming photogrammetric measurements ...



Möhlenbrink 1997



Breuckmann



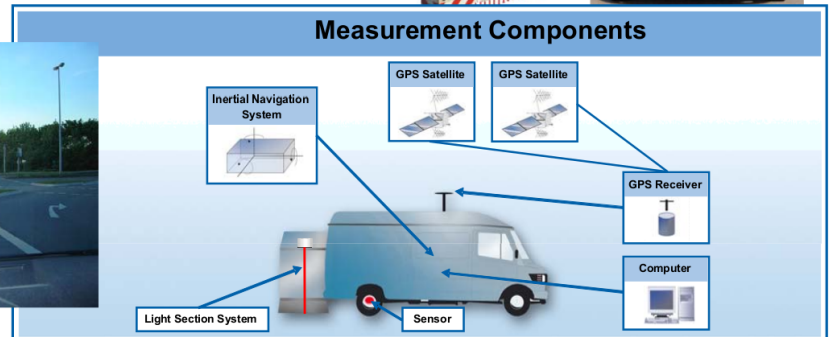
Schniering



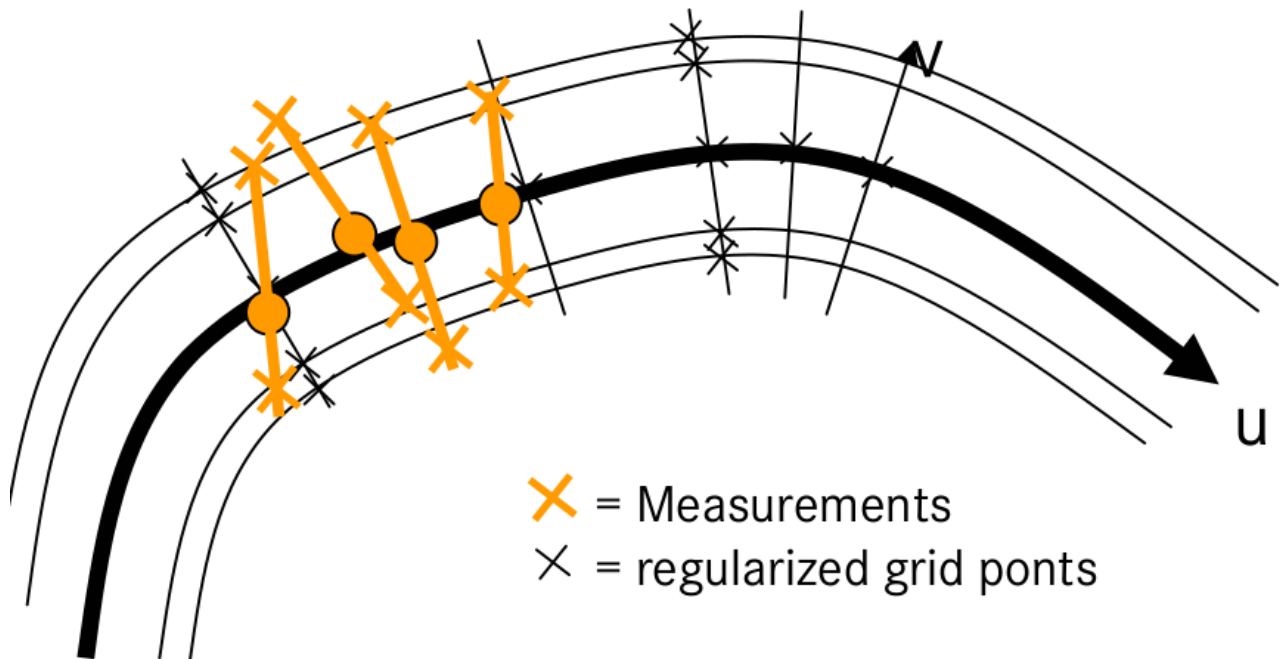
... to fast measurements even on public roads



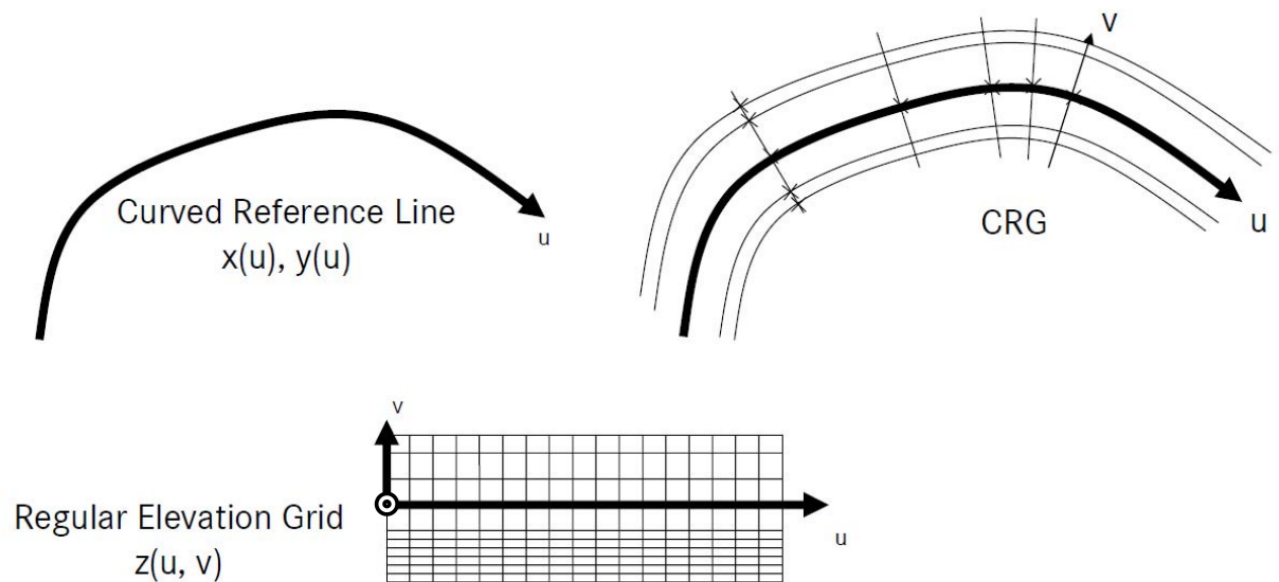
Daimler 2003



- 测量结果是带有微小误差的道路横截面数据和车辆运动朝向角，对数据进行预处理



- 使用一种叫做"curved regular grid"(CRG)的结构存储道路中心线附近的路面高度数据，沿着道路 reference line 进行分割，生成网格状结构，之后把高度数据存放在规则的网格中



- CRG数据主要内容

1. z-array

- $z(u, v)$: 网格点上的高度

2. u-axis(reference line)

- $ubeg$: 起点坐标(默认为0.00m)
- $uend$: 终点坐标

3. v-axis(垂直于reference line)

- $vmin, vmax$: 中心线左/右侧的 v 值 (等间隔或不等间隔)

4. ϕ (朝向角)

- p : reference line是直线只有一个角度值
- $p(u)$: reference line上两点之间的角度

5. $r0$ (reference line起点)

- $(x0, y0)$: UTM坐标

- 坐标系统



Optional: world geodetic system (WGS84, as used by GPS and Google Maps)

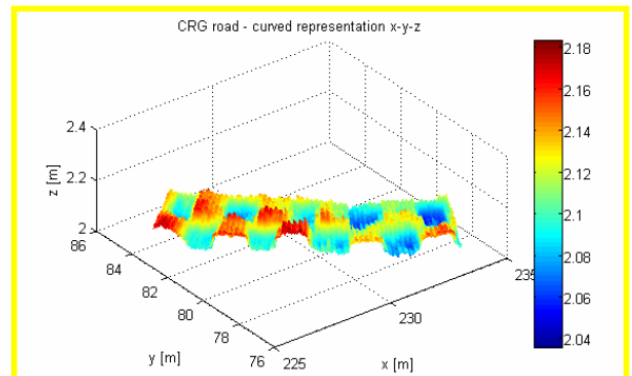
- Header keyword for longitude: reference_line_start_lon
- Header keyword for latitude: reference_line_start_lat

Required: local rectangular system, one of:

- UTM (global grid)
- Local x->east/y->north oriented
- Local rotated

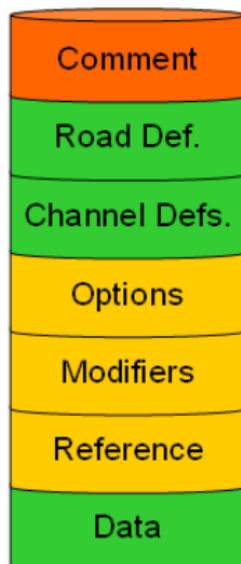
Header keywords:

- reference_line_start_x
- reference_line_start_y



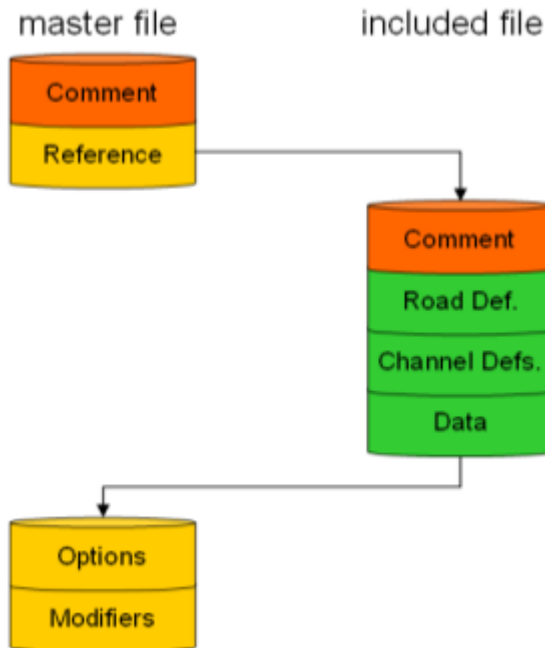
- CRG文件格式

- 文件包含不同的模块



Comment	\$CT:	mandatory comment about file's contents
Road Def.	\$ROAD_CRG:	information about centerline;
Channel Defs.	\$KD_Definition:	mandatory if file contains Data section; definition of individual channel parameters; mandatory if file contains Data section
Options	\$ROAD_CRG_OPTS:	evaluation options
Modifiers	\$ROAD_CRG_MODS:	data set modifiers
Reference	\$ROAD_CRG_FILE:	references to other data files
Data		the actual OpenCRG® data

- 文件必须以注释模块(\$CT)开始，且如果包含数据模块，则数据模块必须放在最后
- 通常将控制文件和数据文件分开



- 道路参数模块(\$ROAD_CRG)定义了reference line的参数和网格参数

```

$CT                                     ! comment text block
crg file example for road surface description (width: 3m, length: 22m)
with curved referenceline and grid of (0.25m...1.0m) x 1.0m
$!*****
$ROAD_CRG                             ! crg road parameters
REFERENCE_LINE_START_U    = 0.0
REFERENCE_LINE_START_X    = 0.0
REFERENCE_LINE_START_Y    = 0.0
REFERENCE_LINE_START_PHI  = 0.0
REFERENCE_LINE_END_U      = 22.0
REFERENCE_LINE_END_X      = 21.953
REFERENCE_LINE_END_Y      = 1.209
REFERENCE_LINE_END_PHI    = 0.0
REFERENCE_LINE_INCREMENT  = 1.0
LONG_SECTION_V_RIGHT      = -1.50      ! with explicit definition below
LONG_SECTION_V_LEFT       = 1.50       ! with explicit definition below
LONG_SECTION_V_INCREMENT  = 0.25       ! with explicit definition below
$!*****
  
```

- 数据定义模块(\$KD_Definition)定义了哪些数据会出现在数据模块中，每一条定义以D:开始，reference line的坡度信息slope和朝向角phi可能会定义在这一模块

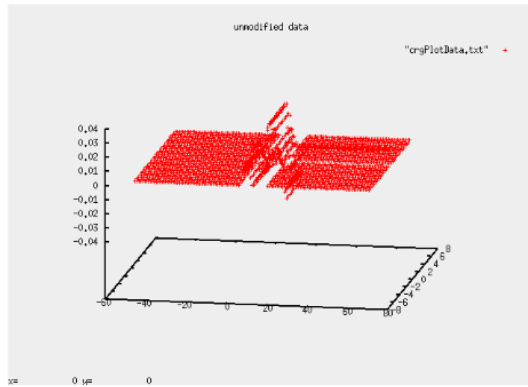
```

$KD_Definition                         ! data definition block
#:LRFI                                ! one of the men readable IPLoS formats
D:reference line slope,m/m             ! longitudinal slope
D:long section at v = -1.500,m         ! 1.50m right of reference line
D:long section at v = -1.250,m         ! 1.25m right of reference line
D:long section at v = -1.000,m         ! 1.00m right of reference line
D:long section at v = 0.000,m          ! on reference line
D:long section at v = 1.000,m          ! 1.00m left of reference line
D:long section at v = 1.250,m          ! 1.25m left of reference line
D:long section at v = 1.500,m          ! 1.50m left of reference line
$!*****
  
```

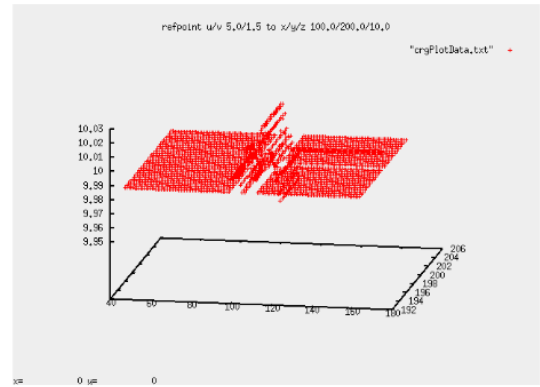
- 在一些比较简单的情况下，直接通过文件读取数据同时进行数据评估，然而，有些时候希望改变评估数据的方式而不改变数据本身

- Modifiers**修改存储在内存中的数据(旋转、缩放、平移等)，并应用于接下来的数据评估过程

original data set

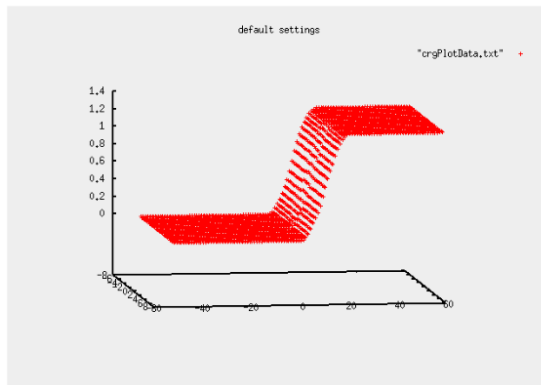


modified data set

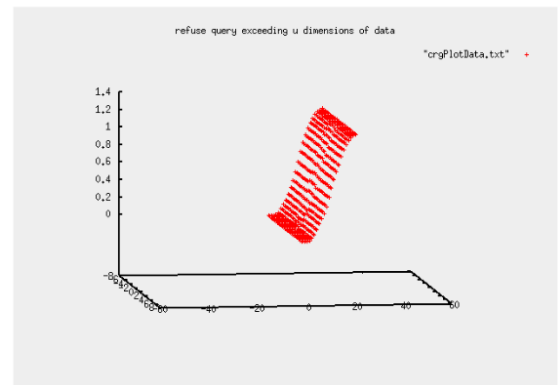


2. Options在不改变数据的情况下影响某些请求的执行方式

default settings



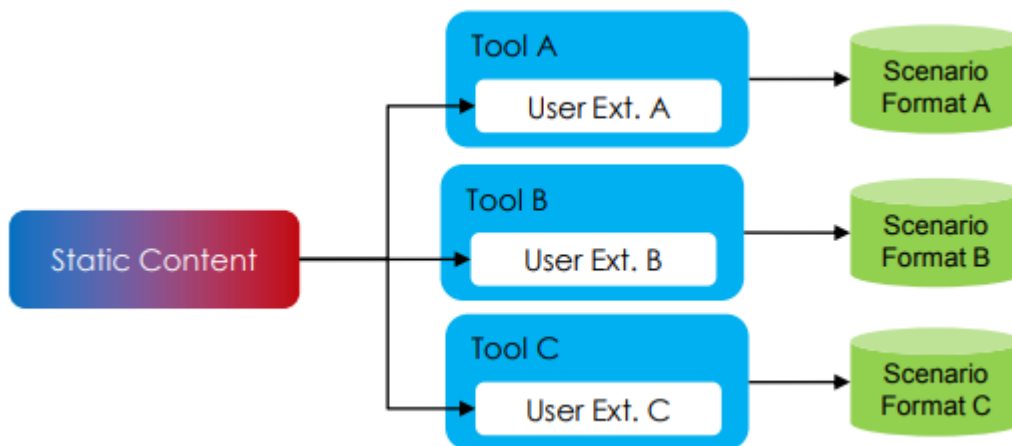
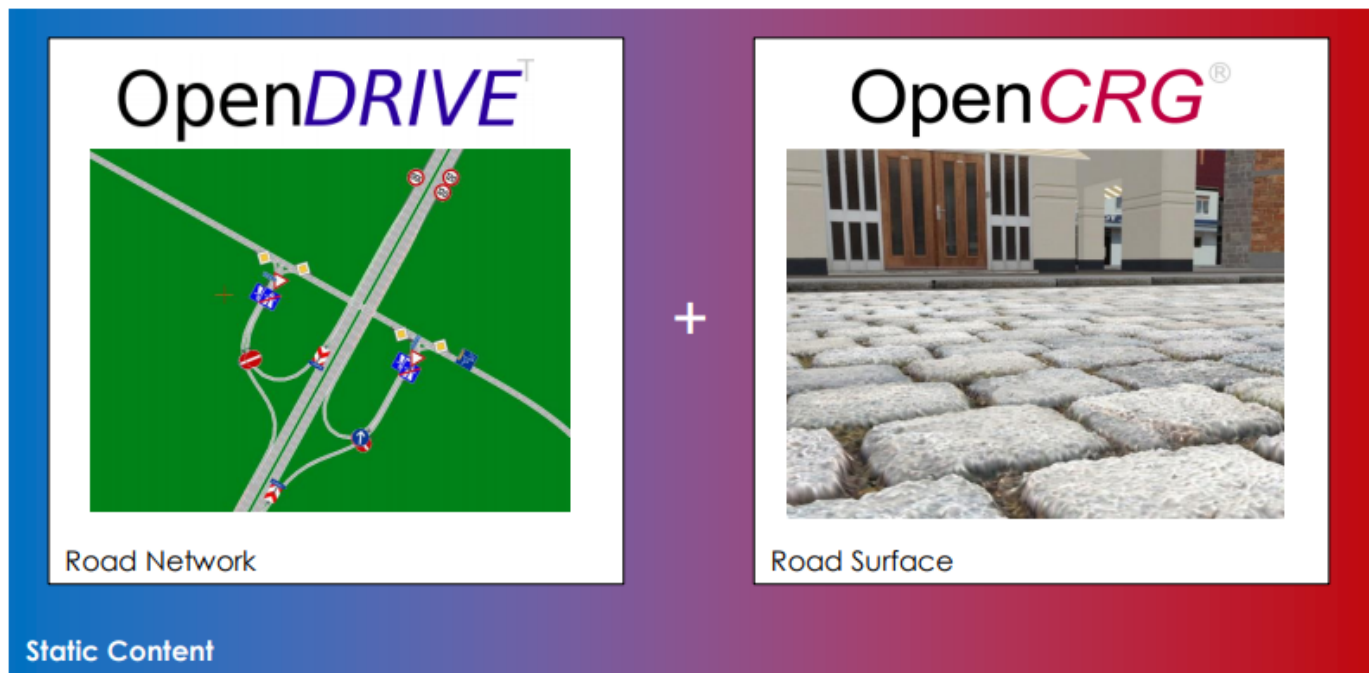
option settings



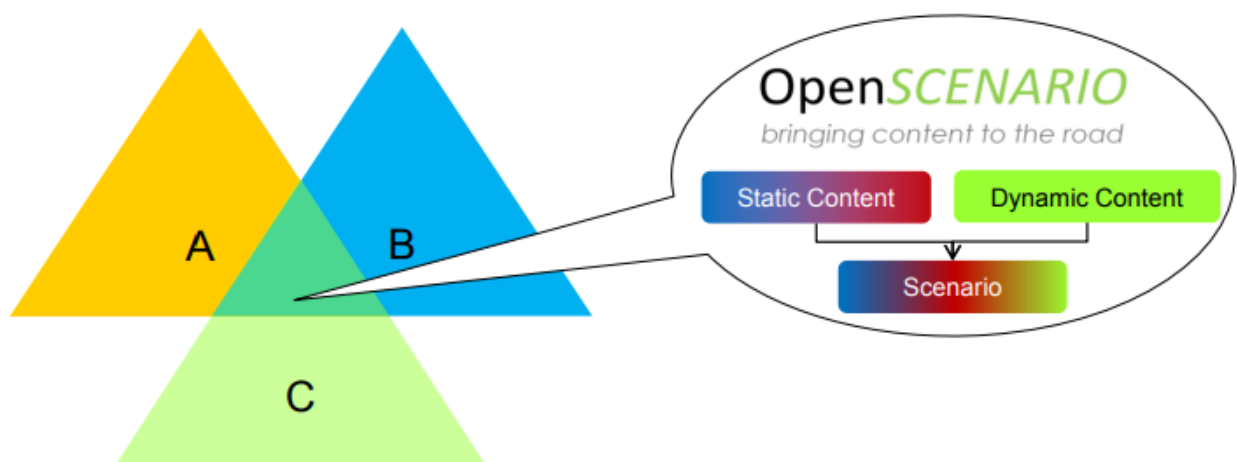
- 数据模块的格式在数据定义模块(\$KD_Definition)中定义

```
$KD_Definition                                     ! data definition block
#:LRFI                                              ! one of the men readable IPLOS formats
U:reference line x,m,0,1.0                         ! undefined data, only for IPLOS viewing
D:reference line phi,rad
D:long section at v = -1.500,m                     ! 1.50m right of reference line
D:long section at v = -1.250,m                     ! 1.25m right of reference line
D:long section at v = -1.000,m                     ! 1.00m right of reference line
D:long section at v = 0.000,m                      ! on reference line
D:long section at v = 1.000,m                      ! 1.00m left of reference line
D:long section at v = 1.250,m                     ! 1.25m left of reference line
D:long section at v = 1.500,m                     ! 1.50m left of reference line
$!*****
* phi      l_sec 1   l_sec 2   l_sec 3   l_sec 4   l_sec 5   l_sec 6   l_sec 7
*          v=-1.50  v=-1.25  v=-1.00  v= 0.00  v= 1.00  v= 1.25  v= 1.50
$$$$$$$$$0$$$$$$$$$0$$$$$$$$$0$$$$$$$$$0$$$$$$$$$0$$$$$$$$$0$$$$$$$$$0
0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000 0.0111111 0.0000000 0.0000000 0.0000000
0.0110000 0.0000000 0.0000000 0.0111111 0.0111111 0.0111111 0.0000000 0.0000000
0.0220000 0.0000000 0.0111111 0.0111111 0.0111111 0.0111111 0.0111111 0.0000000
...
```

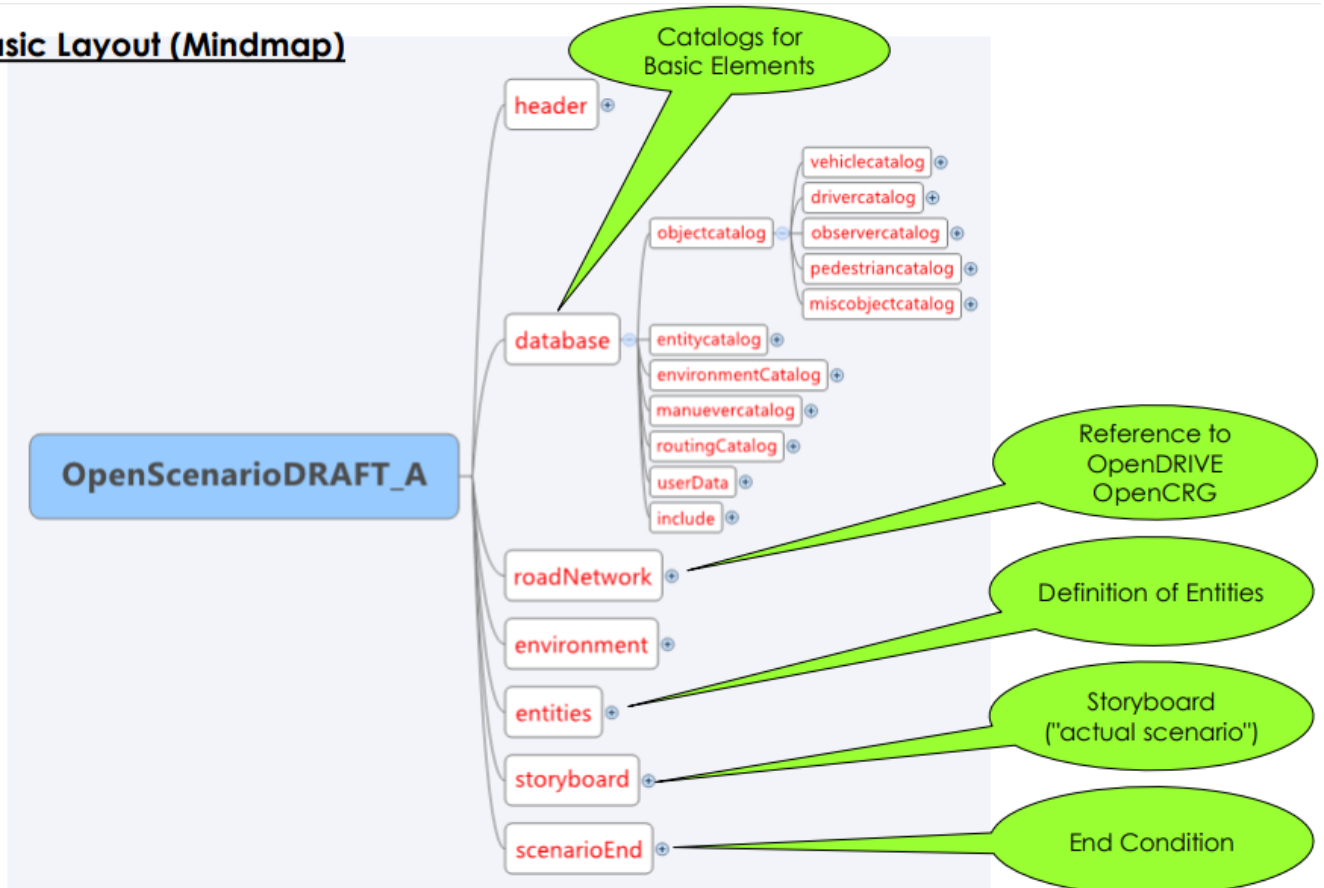
OpenSCENARIO



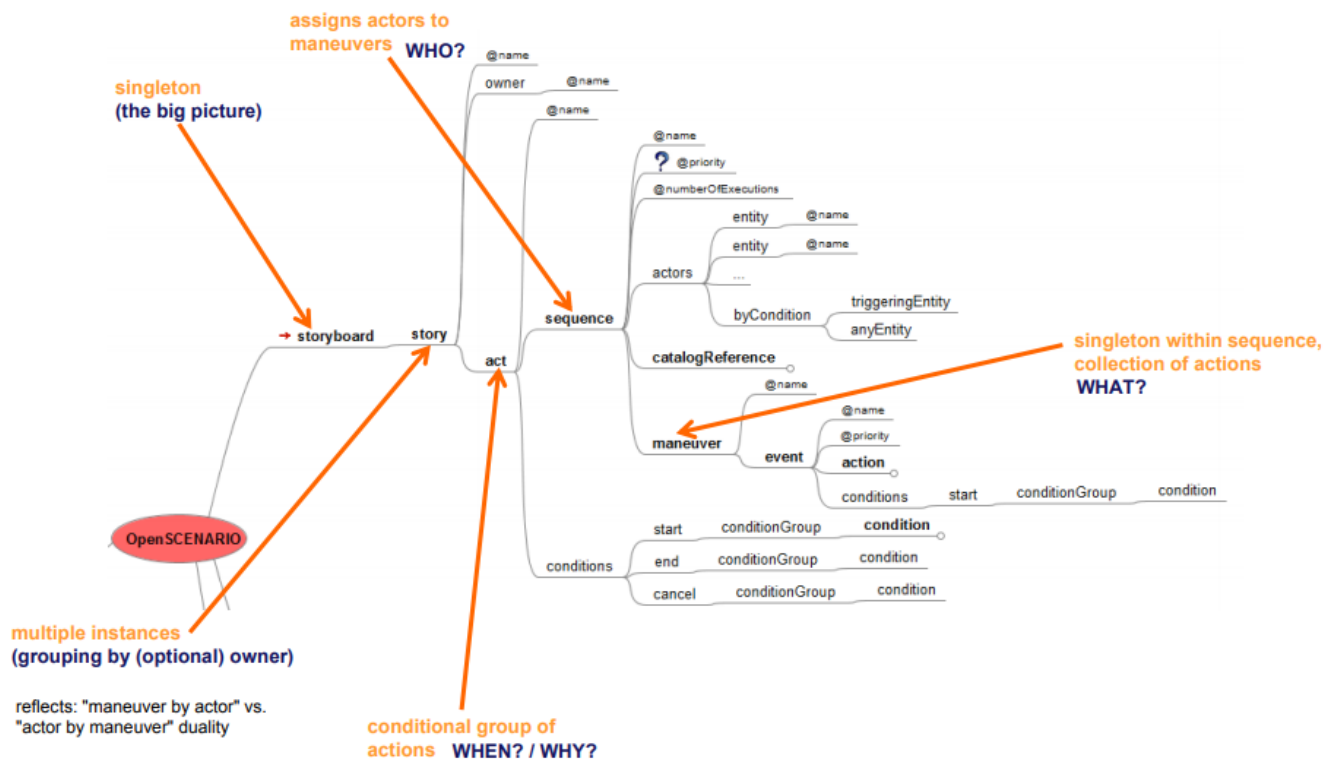
- 动态场景不能以统一标准传输，且严重依赖于所使用工具
- OpenSCENARIO定义了一种用于描述驾驶和交通仿真测试的**动态**场景的文件格式，且独立于软件



Basic Layout (Mindmap)



Storyboard



- Upon the Ego vehicle approaching the vehicle in front, the following things shall happen:
 - player 1 in the front lane shall decelerate and merge to the left
 - with 1s delay player 2 in the front right lane shall merge to the left
 - with 2s delay Ego vehicle accelerates
 - Ego vehicle applies emergency brake

storyboard**story: owner** = Ego**act 1: condition** = TTC to player 1 < 3s**sequence 1.1: actor** = player 1**maneuver 1.1.1: name** = complex lane change**event 1.1.1.1: condition** = upon start of act**action 1.1.1.1.1:** slow down by 5kph**action 1.1.1.1.2:** perform lane change to left within 4s**sequence 1.2: actor** = player 2**maneuver 1.2.1: name** = complex lane change**event 1.2.1.1: condition** = 1s after start of act**action 1.2.1.1.1:** increase speed by 5kph**action 1.2.1.1.2:** perform lane change to left within 5s**sequence 1.3: actor** = \$owner**maneuver 1.3.1: name** = accelerate**event 1.3.1.1: condition** = 1s after end of maneuver 1.2.1**action 1.3.1.1.1:** accelerate by 2m/s^2 to $v_{\text{Target}} = v_{\text{Current}} + 5\text{ m/s}$ **act 2: condition** = TTC to player in front < 1.5s**sequence 2.1: actor** = \$owner**maneuver 2.1.1: name** = emergency brake**event 2.1.1.1: condition** = upon start of act**action 2.1.1.1.1:** apply full brake (or: apply deceleration command for 9m/s^2)