



Workshop

Generating detailed 3D Streetspace Models in CityGML 3.0 using the free OpenDRIVE data conversion tool r:trân

115th OGC Technical Committee (Virtual) – September 15 2021

Christof Beil, Benedikt Schwab, Thomas H. Kolbe

(christof.beil, benedikt.schwab, thomas.kolbe)@tum.de

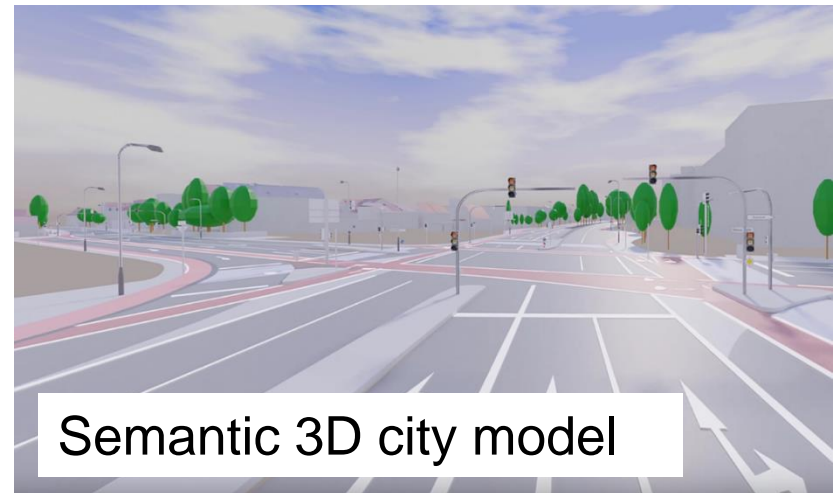
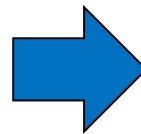
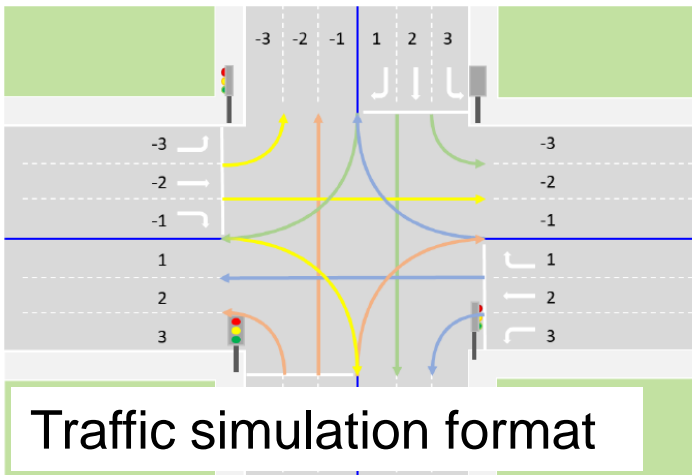
Technical University of Munich

Agenda

- ▶ Motivation
- ▶ Comparison of OpenDRIVE 1.6 and CityGML 3.0
- ▶ **r:trân** – Converting OpenDRIVE data to CityGML
 - Introduction to the software
 - Hands-on tutorial
- ▶ Visualizations using FME / 3DCityDB
- ▶ Try it yourself!

Motivation

- ▶ **OpenDRIVE** is a commonly used standard in the automotive industry
- ▶ Mostly commercial software for **driving simulations**
- ▶ Limited (free) software for visualizing the data
- ▶ **Difficult** to get 3D **visualization** from OpenDRIVE data (parametric representation)



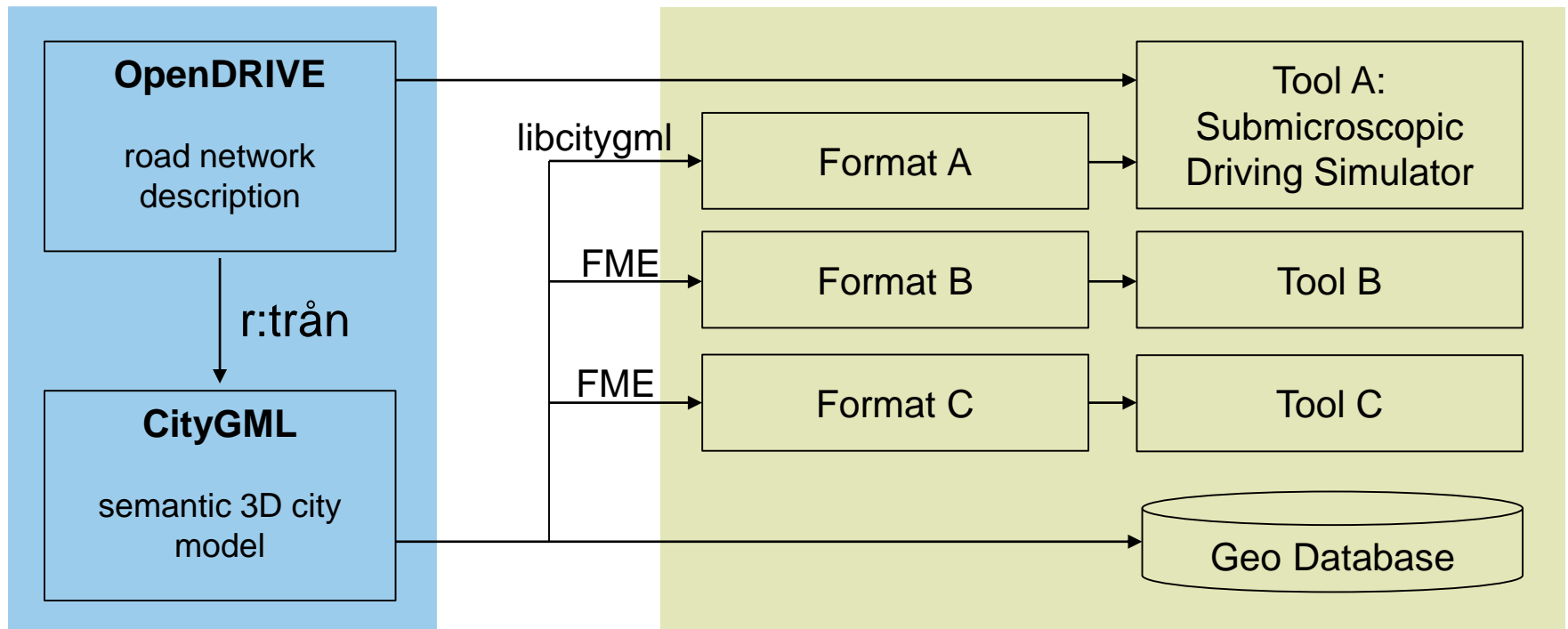
Motivation

- ▶ **Few options** to combine with other data (point clouds, BIM, city models etc.)
- ▶ Few possibilities for sub-setting data (**no WFS, no database storage**, file-based)
- ▶ Other applications need explicit geometrical representation
 - E.g. asset / facility management, spatial analyses / simulation
- ▶ **CityGML suitable** for these applications
- ▶ **Open tools** for storage, management and visualization of CityGML data available (e.g. **3DCityDB**)

Motivation

Spatio-Semantic Road Space Model

Application

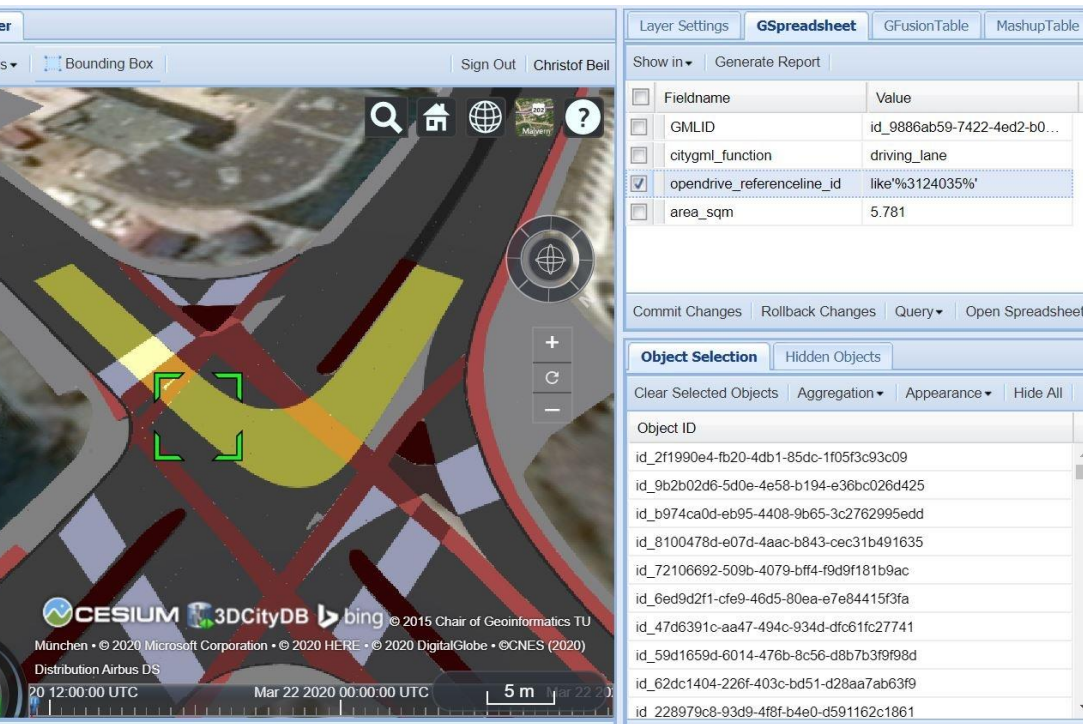


(Schwab, Beil, Kolbe, 2020)

Application of the OpenDRIVE data converted to CityGML

Spatial / semantic query

Map for pedestrian simulation tool “momenTUM”

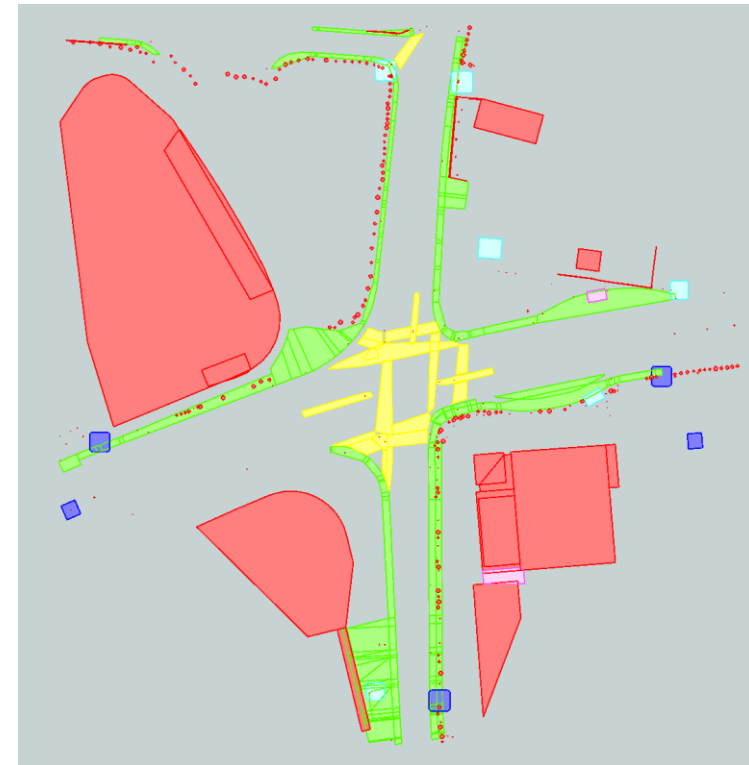


The screenshot shows the Cesium 3D CityDB web interface. On the left, a map view displays a street intersection with overlaid OpenDRIVE data in yellow and green. On the right, the 'Layer Settings' panel is active, showing a table of object selection results.

Fieldname	Value
GMLID	id_9886ab59-7422-4ed2-b0...
citygml_function	driving_lane
<input checked="" type="checkbox"/> opendrive_referenceline_id	like'%3124035%'
<input type="checkbox"/> area_sqm	5.781

Below the table, the 'Object Selection' panel shows a list of object IDs:

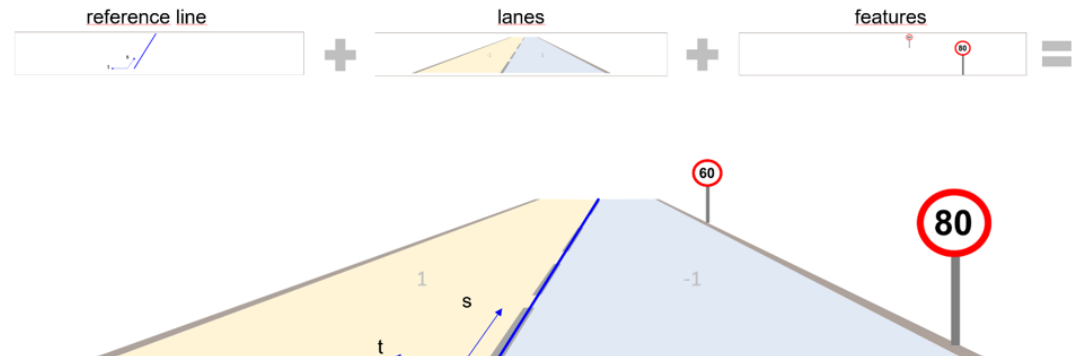
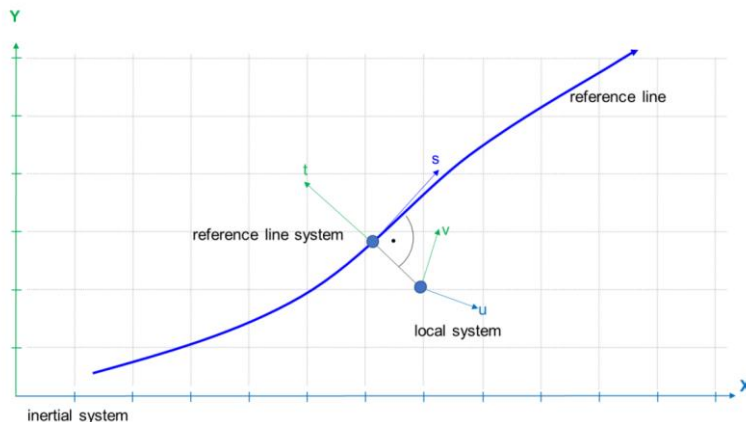
- id_2f1990e4-fb20-4db1-85dc-1f05f3c93c09
- id_9b2b02d6-5d0e-4e58-b194-e36bc026d425
- id_b974ca0d-eb95-4408-9b65-3c2762995edd
- id_8100478d-e07d-4aac-b843-cec31b491635
- id_72106692-509b-4079-bff4-f9d9f181b9ac
- id_6ed9d2f1-cfe9-46d5-80ea-e7e84415f3fa
- id_47d6391c-aa47-494c-934d-dfc61fc27741
- id_59d1659d-6014-476b-8c56-d8b7b3f9f98d
- id_62dc1404-226f-403c-bd51-d28aa7ab63f9
- id_228979c8-93d9-4f8f-b4e0-d591162c1861



(Schwab, Beil, Kolbe, 2020)

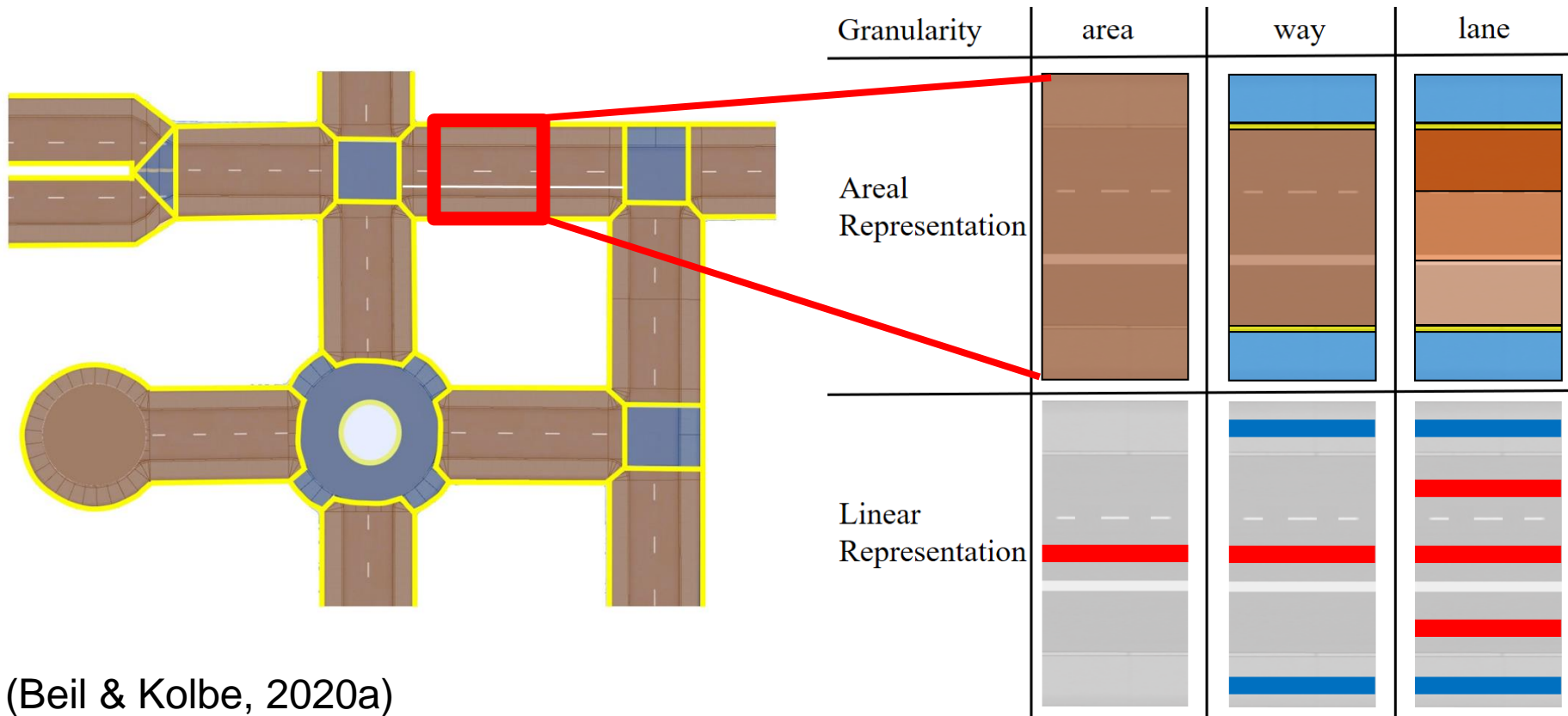
OpenDRIVE

- ▶ Established standard in the automotive industry
- ▶ Analytical description of road networks for
 - driving simulation
 - georeferencing with PROJ string
- ▶ Modelling concept
 - Reference line based (parametric representations)
 - Multiple affine transformations
 - Automotive-specific



CityGML 3.0 Transportation Module

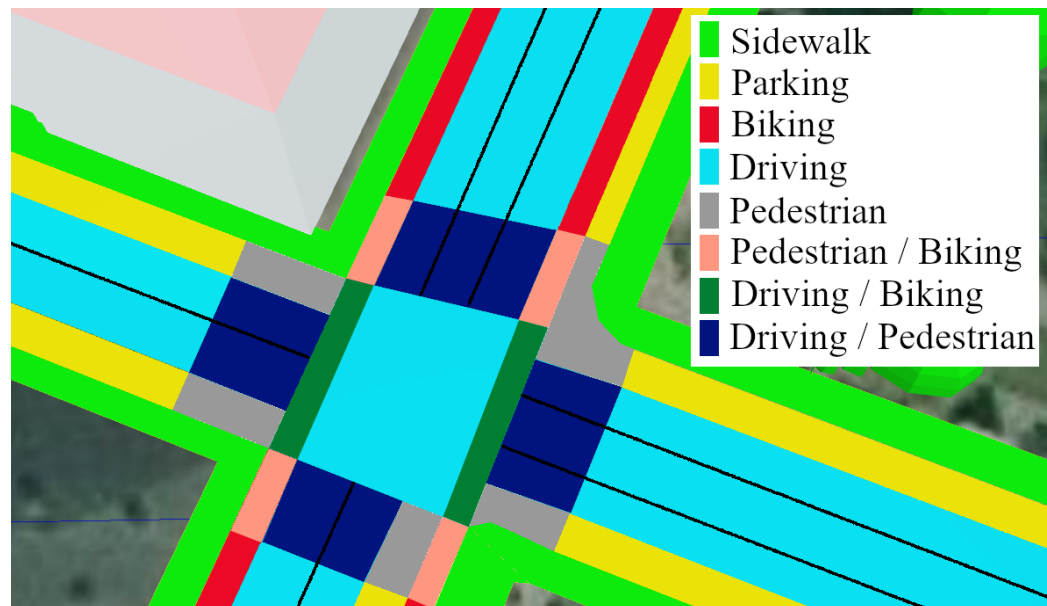
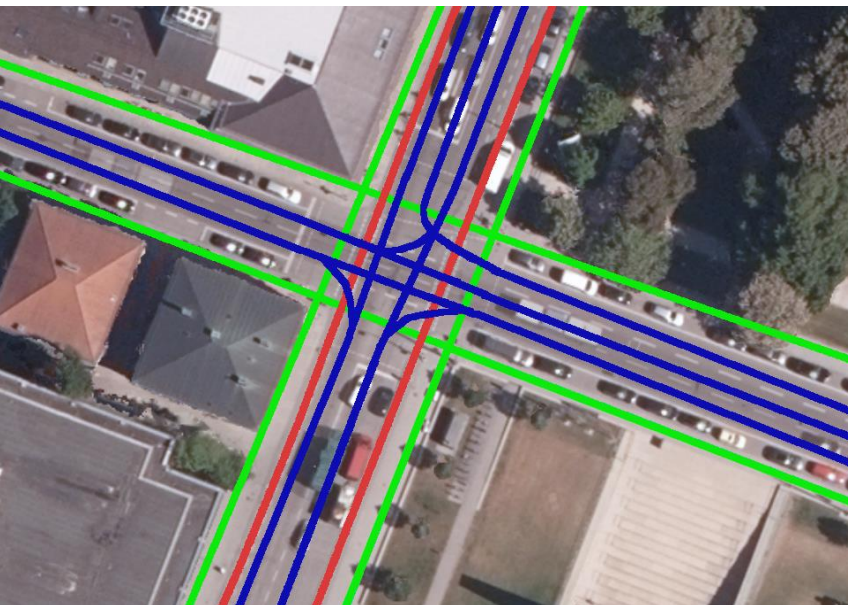
- **Sections:** Segments belonging to one Road
- **Intersections:** Belonging to multiple Roads



CityGML 3.0 Transportation Module

- ▶ **Combined Road, Track, Railway, Waterway representations**
- ▶ **CityGML** module supports representation of multiple transportation types in a **consistent, standardized and integrated** way

(Beil & Kolbe, 2020b)



Comparison of OpenDRIVE 1.6 and CityGML 3.0

► Application Domain

CityGML3.0

Topographic model w.r.t.
geometry, topology, semantics
and appearance

- Core element: *“city object as geographic feature with volumetric extent*
- Application domains:
 - Geospatial simulations (flood, energy, ...)
 - Urban planning
 - ...

OpenDRIVE1.6

Model to describe static road space
networks and objects along roads

- Central element: *“road”*
 - Main application domain:
Driving simulation

Comparison of OpenDRIVE 1.6 and CityGML 3.0

► Geometries

CityGML3.0

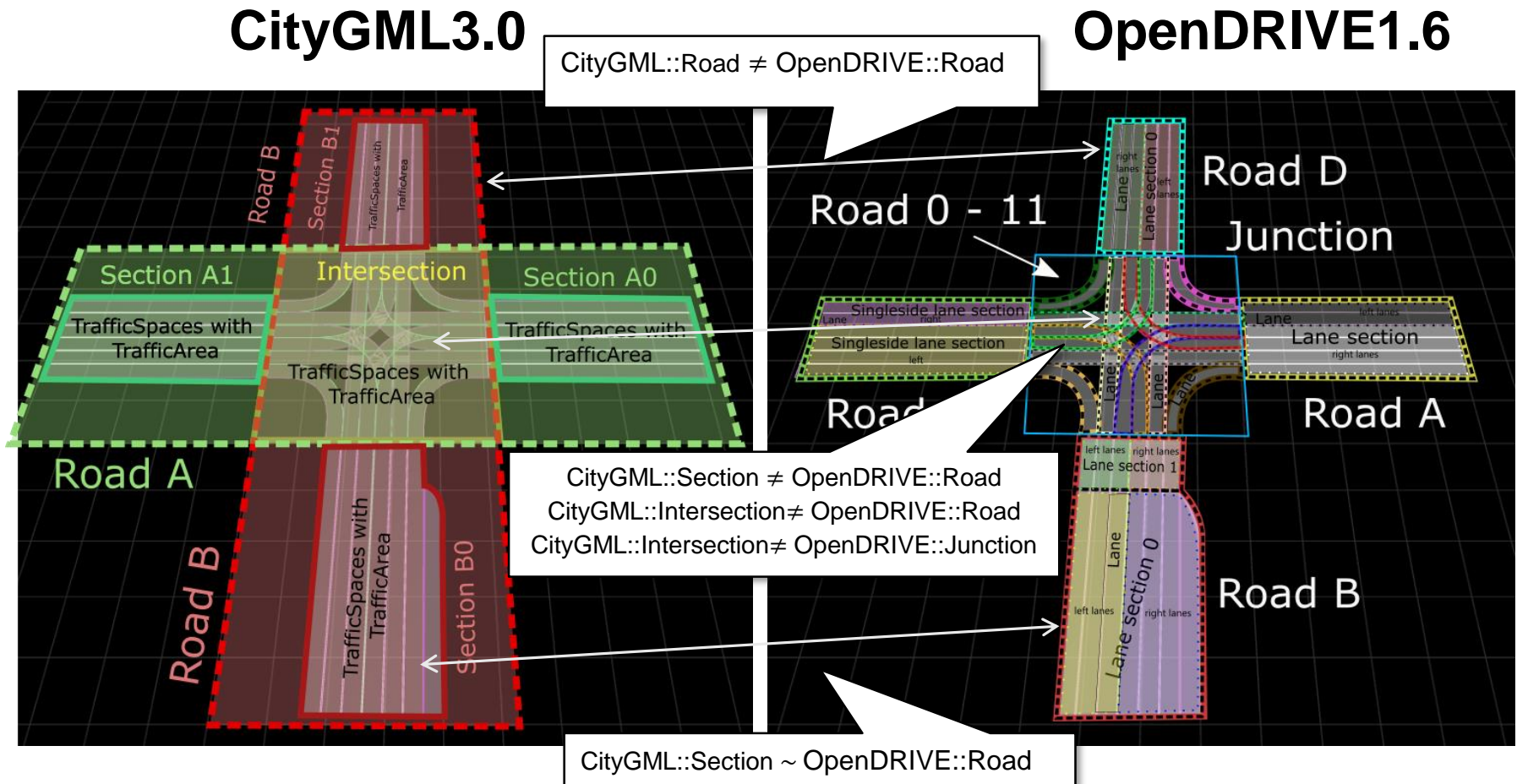
- Application schema of GML (ISO Standard)
- Conceptual model uses the spatial schema of ISO 19107
- Mainly explicit geometry (derivation of datasets from real-world measurement data)

OpenDRIVE1.6

- Standard-specific geometry
- Parametric geometry for road modeling
- Based on a reference line

Comparison of OpenDRIVE 1.6 and CityGML 3.0

► Semantic concepts





- ▶ r:trân is an
 - open library
 - for road space model transformations
 - steered by parameterizable recipes

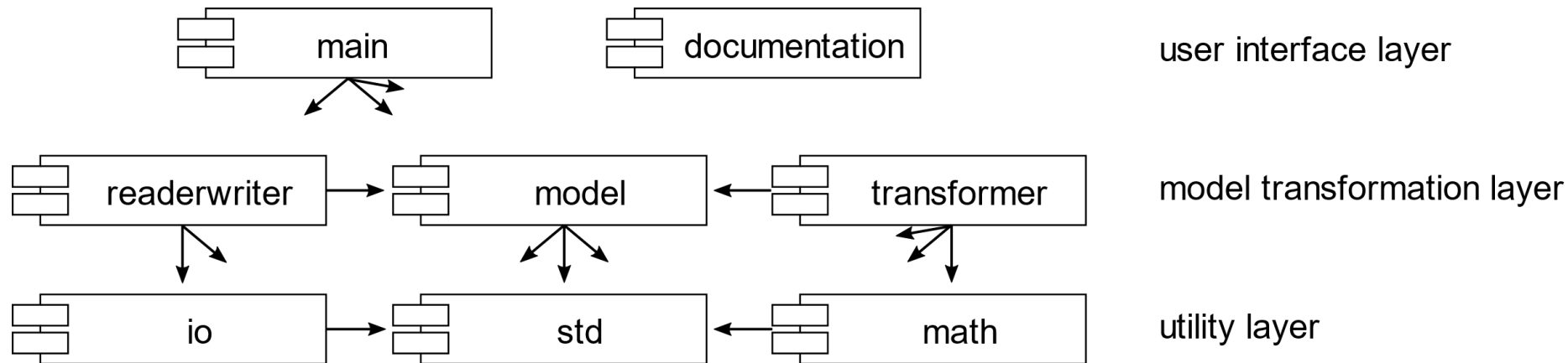
- ▶ Initial use cases
 - Conversion of OpenDRIVE → CityGML 2.0
 - Conversion of OpenDRIVE → CityGML 3.0

<https://rtron.io>

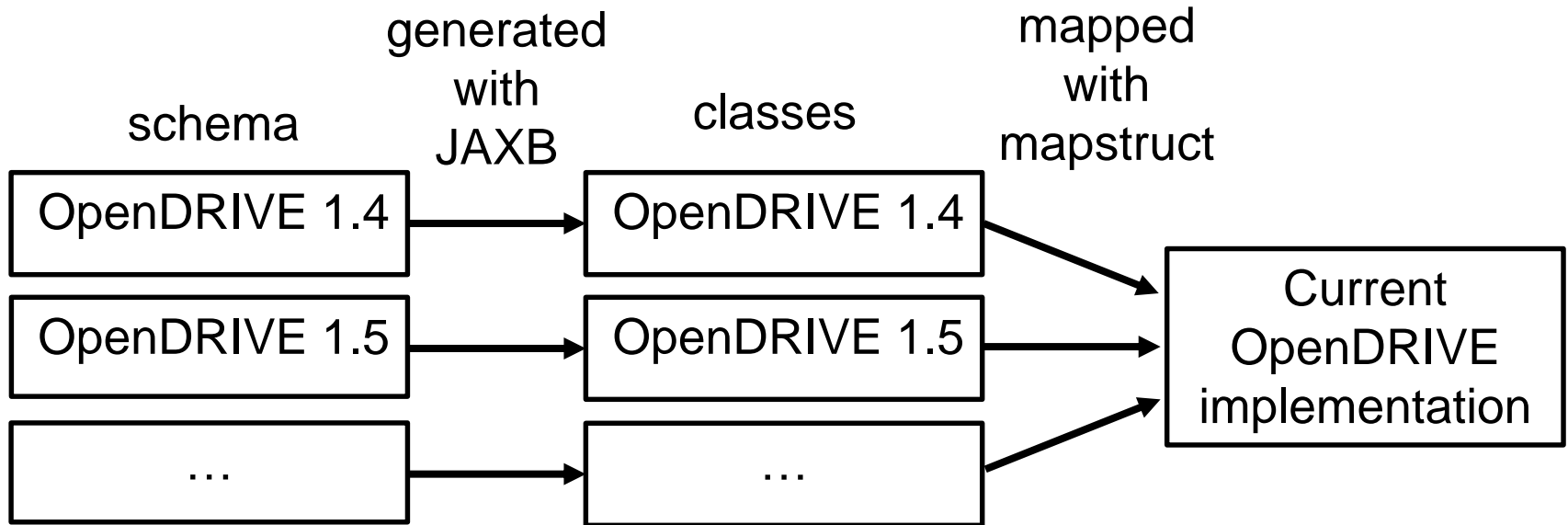
<https://github.com/tum-gis/rtron>

Architecture of r:trân

- ▶ Runs on the Java Virtual Machine (JVM)
- ▶ Implemented in Kotlin: Concise, safe, interoperable
- ▶ Separated into components



OpenDRIVE Reader

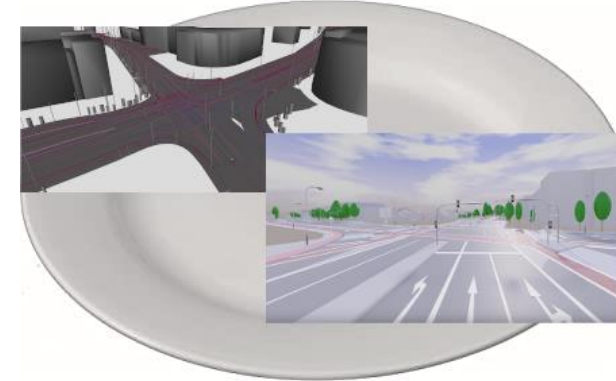
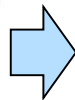
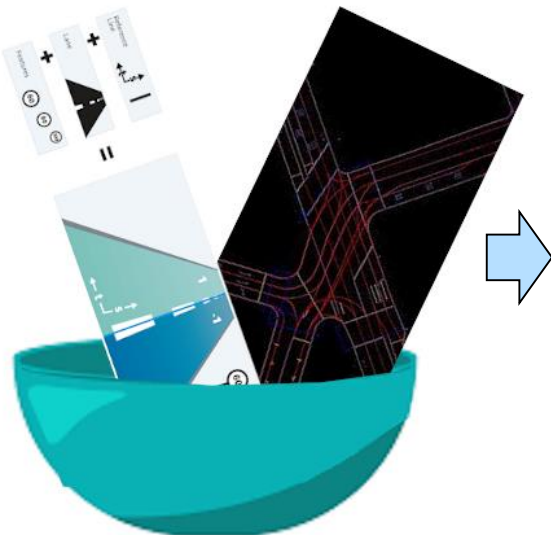


Mapstruct

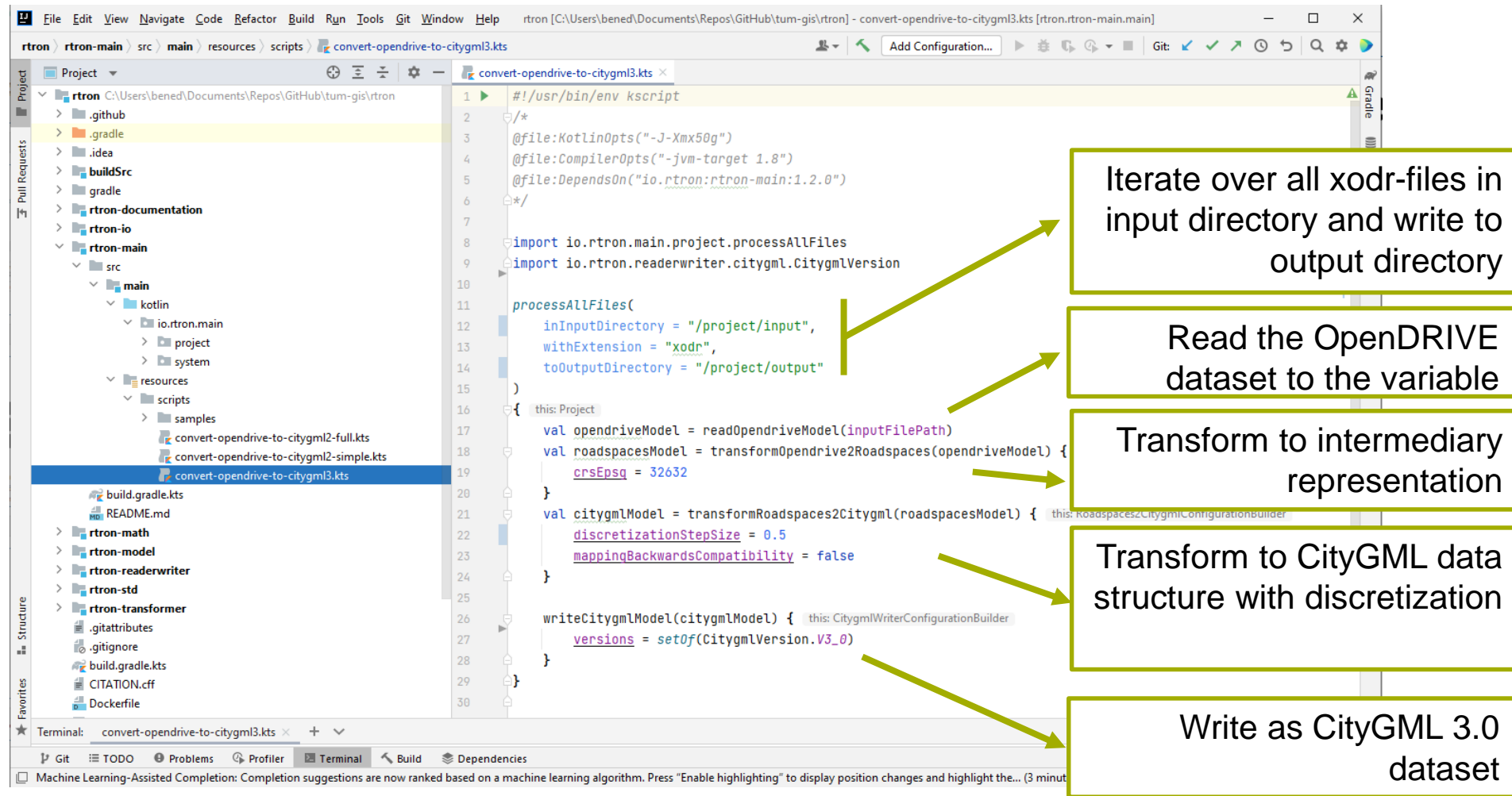
- ▶ Generation of mappings during compile time
- ▶ Annotations: OpenDRIVE 1.4 → 1.6 takes 300LoC
- ▶ Type-safe mapping, performant, inverse possible

Usage of r:trân – multiple options

- ▶ Recipe development in IDE
- ▶ Library for other projects
- ▶ Deployment via
 - kscript directly
 - **Docker**



Recipe in IntelliJ IDEA



Iterate over all xodr-files in input directory and write to output directory

Read the OpenDRIVE dataset to the variable

Transform to intermediary representation

Transform to CityGML data structure with discretization

Write as CityGML 3.0 dataset

```

1  #!/usr/bin/env kscript
2  /*
3  @file:KotlinOpts("-J-Xmx50g")
4  @file:CompilerOpts("-jvm-target 1.8")
5  @file:DependsOn("io.rtron:rtron-main:1.2.0")
6  */
7
8  import io.rtron.main.project.processAllFiles
9  import io.rtron.readerwriter.citygml.CitygmlVersion
10
11  processAllFiles(
12      inInputDirectory = "/project/input",
13      withExtension = "xodr",
14      toOutputDirectory = "/project/output"
15  )
16  { this: Project
17      val opendriveModel = readOpendriveModel(inputFilePath)
18      val roadspacesModel = transformOpendrive2Roadspaces(opendriveModel) {
19          crsEpsg = 32632
20      }
21      val citygmlModel = transformRoadspaces2Citygml(roadspacesModel) {
22          discretizationStepSize = 0.5
23          mappingBackwardsCompatibility = false
24      }
25
26      writeCitygmlModel(citygmlModel) { this: CitygmlWriterConfigurationBuilder
27          versions = setOf(CitygmlVersion.V3_0)
28      }
29  }
30  
```

Machine Learning-Assisted Completion: Completion suggestions are now ranked based on a machine learning algorithm. Press "Enable highlighting" to display position changes and highlight the... (3 min)

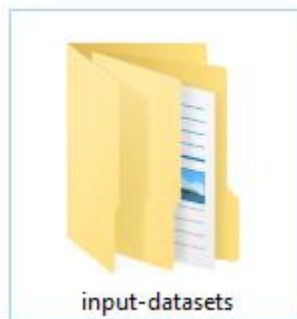
Hands-on Tutorial

► Prerequisite

- Commands are based Windows OS
- Docker installation
- Data inspector of FME (Safe Software)
- Optional 3DCityDB

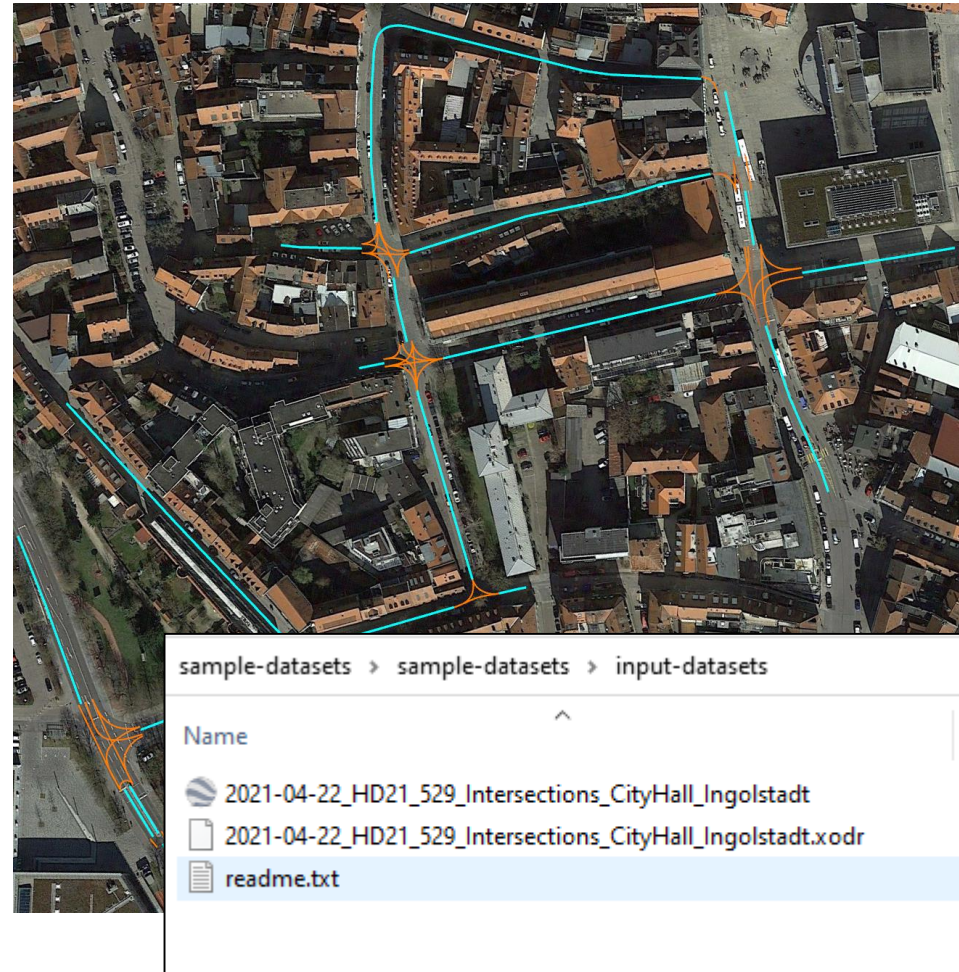
► Downloaded sample data:

<https://syncandshare.lrz.de/getlink/fiCYHWPtetehc1TybqWRpeRy/sample-datasets.zip>



Step 1: Unzip “sample-dataset” folder – Open “input-datasets” folder




- ▶ Info on the provided data
- ▶ KML visualization of OpenDRIVE reference lines (GoogleEarth)
- ▶ Light blue: standard line
- ▶ Orange: connection lines in junction areas



Step 2: Open “scripts” folder

- ▶ These are the recipes we will use
 - For creating CityGML2.0 or CityGML3.0 compliant data



Name	Änderungsdatum	Typ	Größe
 opendrive-to-citygml2.kts	06.09.2021 16:15	KTS-Datei	1 KB
 opendrive-to-citygml3.kts	25.08.2021 21:43	KTS-Datei	2 KB
 readme	25.08.2021 21:45	Textdokument	1 KB

Step 3: Open e.g. “opendrive-to-citygml3.kts” in any text editor

- ▶ Here you can adjust e.g. discretization steps
- ▶ For now we will leave everything as it is

```
processAllFiles(  
    inInputDirectory = "/project/input", // adjust path to directory of input datasets  
    withExtension = "xodr",  
    toOutputDirectory = "/project/output" // adjust path to output directory  
)  
{  
    val opendriveModel = readOpendriveModel(inputFilePath)  
    val roadspacesModel = transformOpendrive2Roadspaces(opendriveModel) {  
        crsEpsg = 32632  
    }  
    val citygmlModel = transformRoadspaces2Citygml(roadspacesModel) {  
        // if false, all classes according to CityGML3 are populated  
        mappingBackwardsCompatibility = false  
  
        // distance between each discretization step for curves and surfaces  
        discretizationStepSize = 0.5  
  
        // number of discretization points for a circle or cylinder  
        circleSlices = 12  
    }  
  
    writeCitygmlModel(citygmlModel) {  
        // write as CityGML dataset of version 3  
        versions = setOf(CitygmlVersion.V3_0)  
    }  
}
```

Step 4: Open “readme.txt” (within scripts folder) in any text editor and adapt paths

Replace C:\adjust\path\to\ with the correct path to the sample data directory on your system:

```
docker run -i --name rtron --rm^  
-v C:\adjust\path\to\sample-datasets\input-datasets:/project/input^  
-v C:\adjust\path\to\sample-datasets\output-datasets:/project/output^  
rtron/rtron - < C:\adjust\path\to\sample-datasets\scripts\opendrive-to-citygml3.kts
```

- Path to the downloaded sample dataset directory
- Container internal path to project input & output directory (leave unchanged)
- Transformation recipe script (select desired .kts script)

```
docker run -i --name rtron --rm^  
-v C:\adjust\path\to\sample-datasets\input-datasets:/project/input^  
-v C:\adjust\path\to\sample-datasets\output-datasets:/project/output^  
rtron/rtron - < C:\adjust\path\to\sample-datasets\scripts\opendrive-to-citygml3.kts
```

Step 5: Copy and run command

```
C:\Users\bened>docker run -i --name rtron --rm^ -v C:\Users\bened\sample-datasets\input-dataset:/project/input^ -v C:\Users\bened\sample-datasets\output-datasets:/project/output^ rtron/rtron - < C:\Users\bened\sample-datasets\scripts\opendrive-to-citygml3.kts
Unable to find image 'rtron/rtron:latest' locally
```


Step 6: Find result in “output-datasets” folder

- ▶ In case something didn't work we also provided the resulting datasets in the “output-datasets-backup” folder

sample-datasets > output-datasets-backup > 2021-04-22_HD21_529_Intersections_CityHall_Ingolstadt.xodr

Name



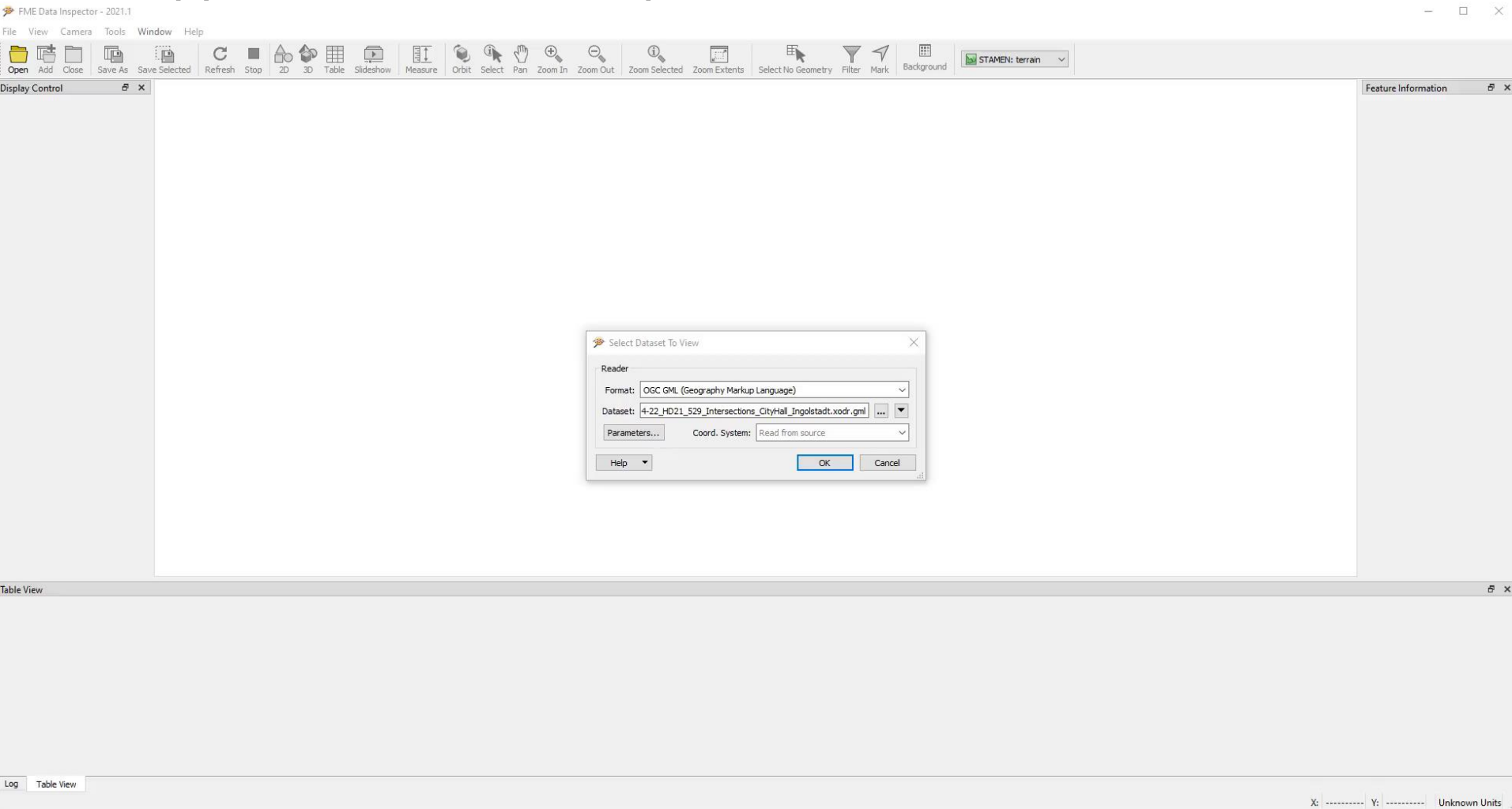
2021-04-22_HD21_529_Intersections_CityHall_Ingolstadt.xodr_V2_0



2021-04-22_HD21_529_Intersections_CityHall_Ingolstadt.xodr_V3_0

Step 7: Inspection of CityGML 3.0 data using FME

- Applications schemas provided in schemas folder



Conversion using a different script and different parameters

Script:
opendrive-to-citygml2.kts

```
opendrive-to-citygml2.kts
1  #!/usr/bin/env kscript
2
3  @file:KotlinOpts("-J-Xmx50g")
4  @file:CompilerOpts("-jvm-target 1.8")
5  @file:DependsOn("io.rtrn:rtrn-main:1.2.0")
6  */
7
8  import io.rtrn.main.project.processAllFiles
9  import io.rtrn.readerwriter.citygml.CitygmlVersion
10
11  processAllFiles(
12      inInputDirectory = "/project/input",
13      withExtension = "xodr",
14      toOutputDirectory = "/project/output"
15  )
16  {
17      val opendriveModel = readOpendriveModel(inputFilePath)
18      val roadspacesModel = transformOpendrive2Roadspaces(opendriveModel) {
19          crsEpsg = 32632
20      }
21      val citygmlModel = transformRoadspaces2Citygml(roadspacesModel) {
22          mappingBackwardsCompatibility = true
23          discretizationStepSize = 10.0
24          circleSlices = 3
25      }
26      writeCitygmlModel(citygmlModel) {
27          versions = setOf(CitygmlVersion.V2_0)
28      }
29  }
30
```

Step size for discretizing
parametric geometries = 10m

Circles and cylinders are
discretized with 3 slices

Inspection CityGML 2.0 using FME

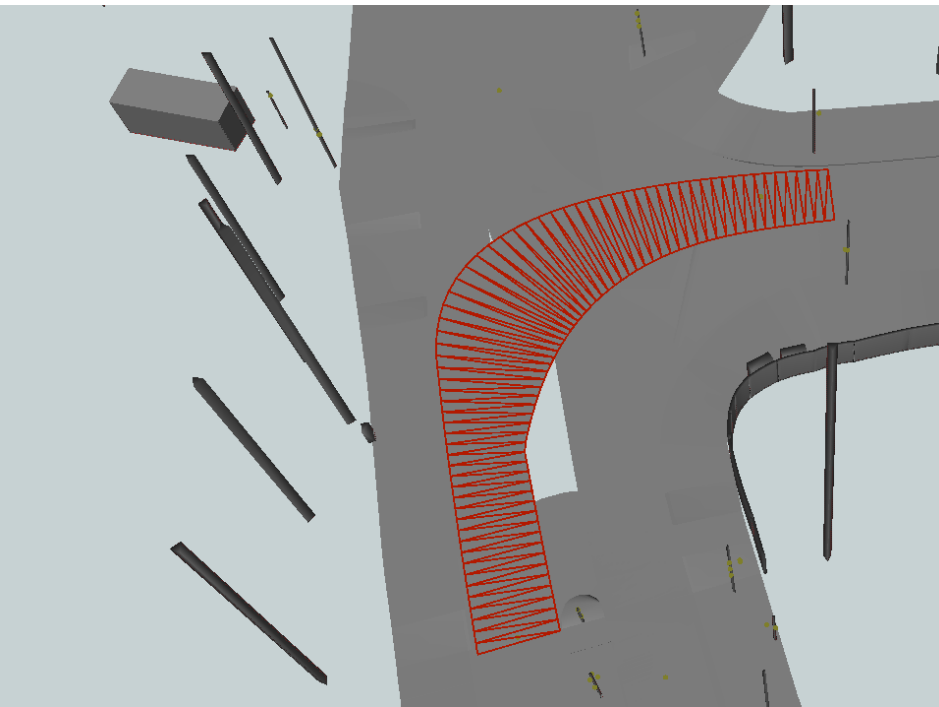
The screenshot displays the FME Data Inspector 2021.1 application window. The main interface is divided into several panes: a top menu bar (File, View, Camera, Tools, Window, Help), a toolbar with various icons for file operations and navigation, a central display area, and side panes for 'Display Control' and 'Feature Information'. A dialog box titled 'Select Dataset To View' is open in the center. It contains the following fields:

- Reader:** A dropdown menu set to 'OGC CityGML'.
- Dataset:** A text field containing the path '4-22_H021_529_Intersections_CityHall_Ingolstadt.xodr.gml'.
- Parameters...** A button next to the dataset field.
- Coord. System:** A dropdown menu set to 'Read from source'.
- Buttons:** 'Help', 'OK', and 'Cancel' at the bottom.

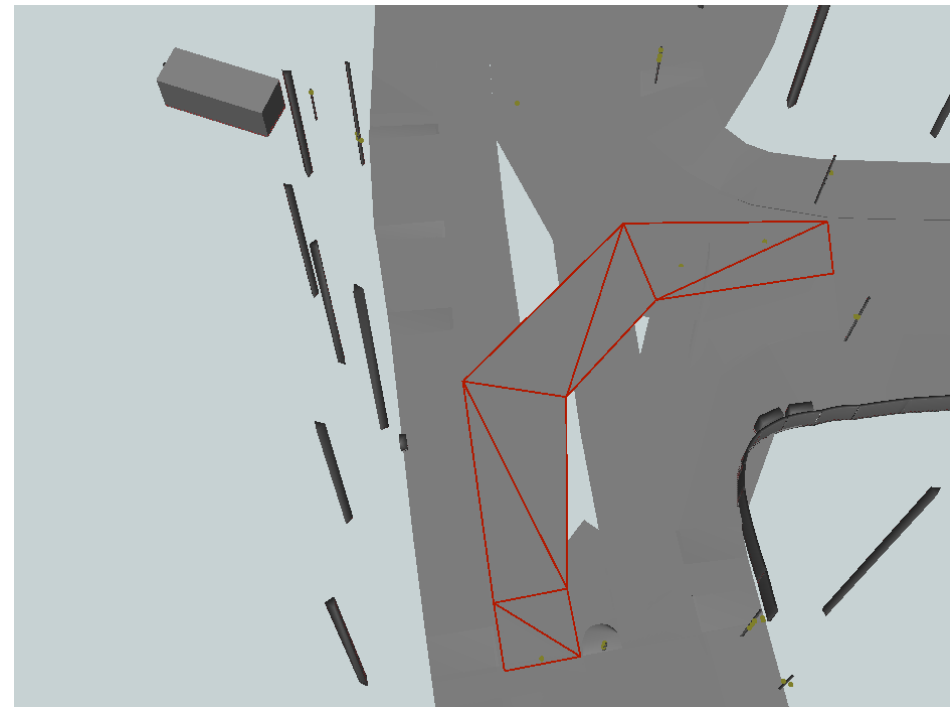
The status bar at the bottom indicates 'Log' and 'Table View' tabs, and the coordinate system is shown as 'Unknown Units'.

Effect of changed discretization step

Discretization step = 0.5 m



Discretization step = 10 m



3D City Database

► 3DCityDB:

- Free 3D geo-database to store, represent, and manage virtual 3D city models <https://www.3dcitydb.org/3dcitydb/>



► Importer / Exporter Tool:

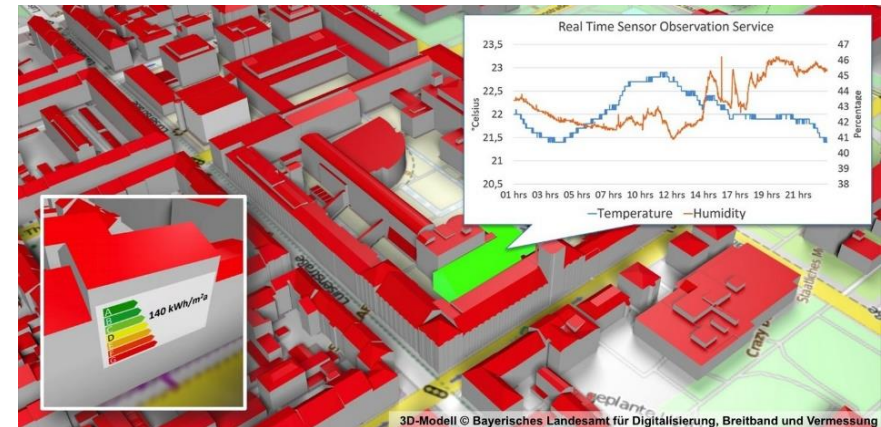
- Import: CityGML data (currently version 2.0, soon: CityGML 3.0)
- Export: CityGML, KML, COLLADA and glTF formats

► Web Feature Service

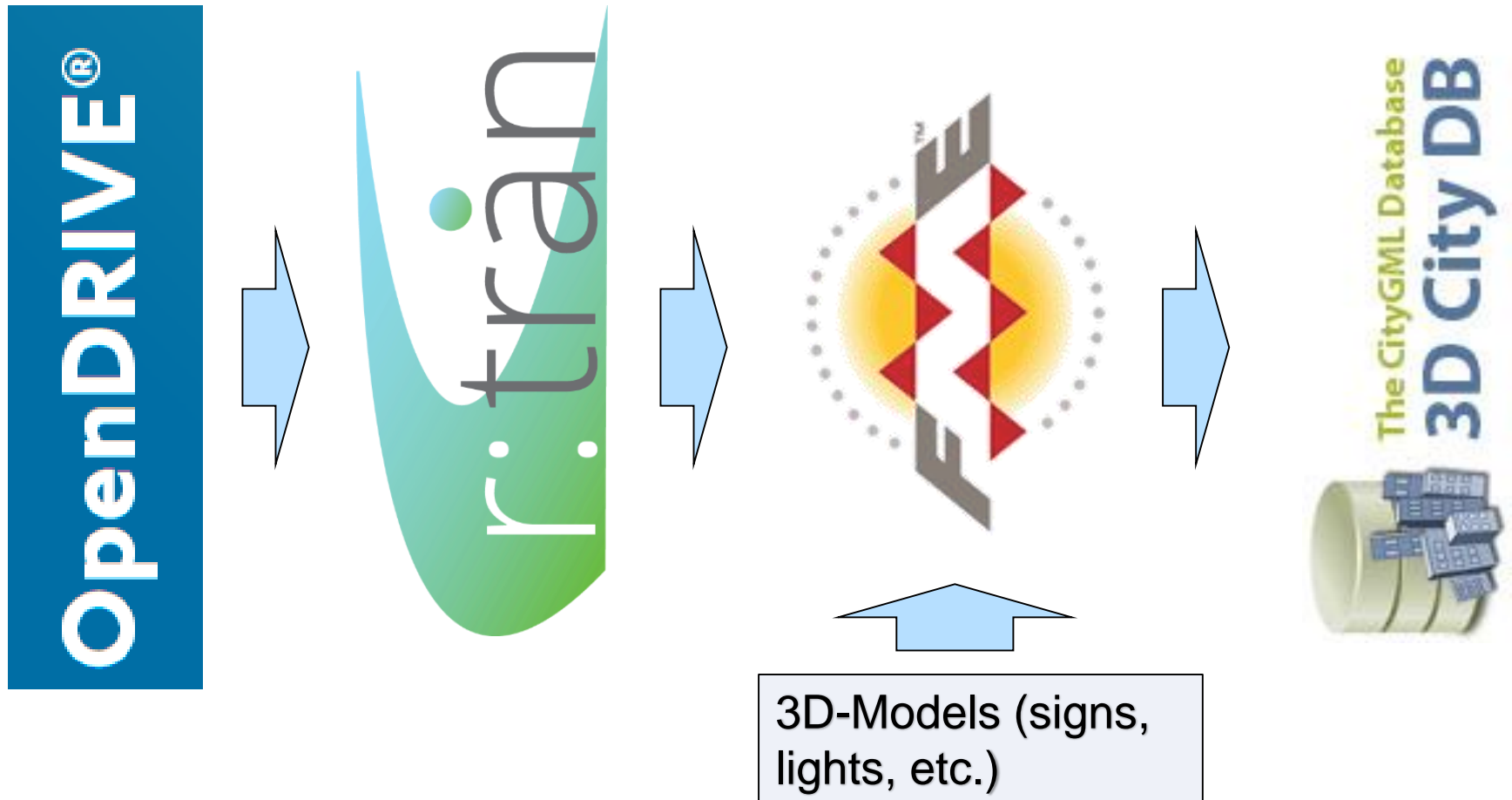
- Web-based access to the 3D city objects

► 3DCityDB-Web-Map-Client

- Interactive 3D visualization based on Cesium virtual globe



Workflow: Generating a detailed 3D Streetspace model



3DCityDB Web-Map-Client Visualization

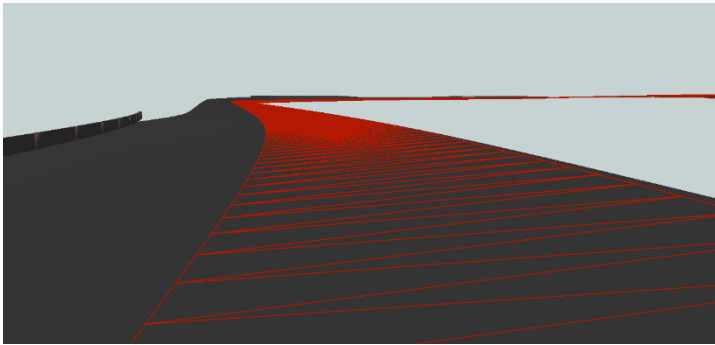
- 3D streetspace model of Ingolstadt

<https://wiki.tum.de/display/gisproject/Online+Demo+Collection>

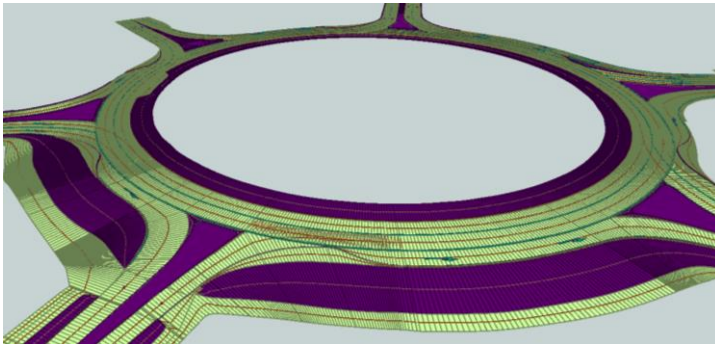


Further OpenDRIVE Datasets

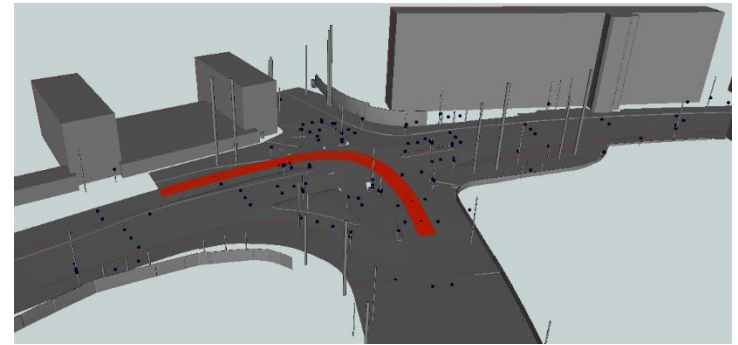
- ▶ 3D Mapping Solutions: <https://www.3d-mapping.de/en/customer-area/>
- ▶ Around 130 ready-to-use datasets



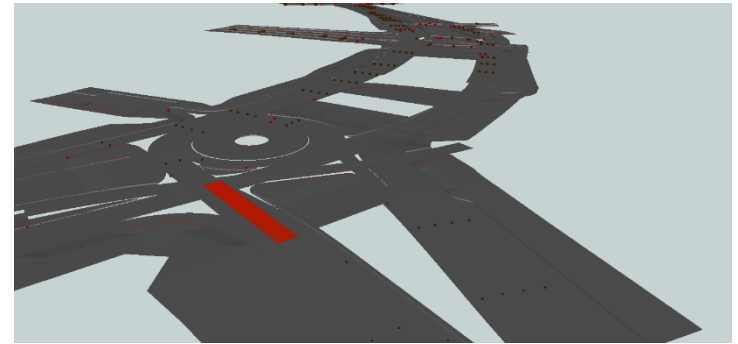
USA, CA: Highway Nr.1



Germany: Roundabout



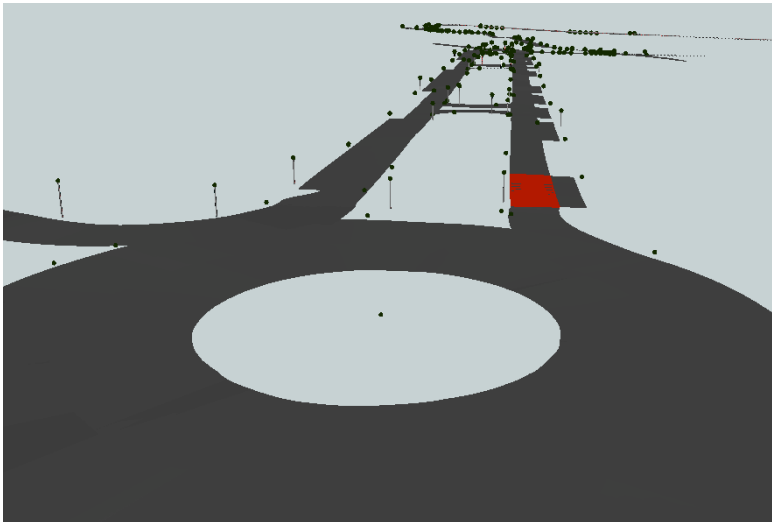
Germany: Complex intersection



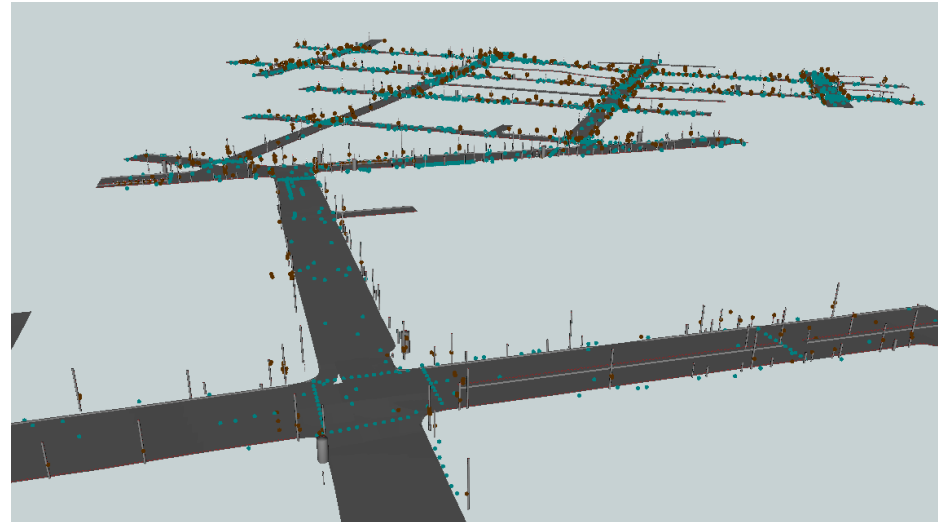
Germany: Complete inner city ring

Further OpenDRIVE Datasets

- ▶ Atlatec: <https://atlatec.de/en/getsampleddata.html>
- ▶ For more: <https://github.com/b-schwab/awesome-openx>



Spain: El Vendrell



USA: San Francisco

Conclusion

- ▶ Questions?
 - Issue at <https://github.com/tum-gis/rtron>
- ▶ Contributions welcome
 - New OpenDRIVE dialects
 - E.g. new transformers/ writers
- ▶ Feature ideas
 - OpenDRIVE inconsistency report or fixing (beyond the schema)
 - OpenDRIVE version upmapping
 - Geodata to OpenDRIVE conversions
 - Implementation of OpenDRIVE-ADE for CityGML
- ▶ **Try converting OpenDRIVE data yourself! Following the steps on slide 20 – 29**

References

- Beil, C.; Kolbe, T. H. (2017).** CityGML and the streets of New York-A proposal for detailed street space modelling. In Proceedings of the 12th International 3D GeoInfo Conference 2017 (pp. 9-16). <https://mediatum.ub.tum.de/doc/1368093/1368093.pdf>
- Beil, C.; Kolbe, T. H. (2020a).** Combined modelling of multiple transportation infrastructure within 3D city models and its implementation in CityGML 3.0. Proceedings of the 15th International 3D GeoInfo Conference 2020 (ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences), ISPRS. <http://mediatum.ub.tum.de/node?id=1554455>
- Beil C.; Kolbe, T.H. (2020b).** Detailed Streetspace Modelling for Multiple Applications: Discussions on the Proposed CityGML 3.0 Transportation Model. ISPRS International Journal of Geo-Information **9** (10), 603. <https://www.mdpi.com/2220-9964/9/10/603>
- Coduro, T. (2018).** Straßenraummodellierung mittels Mobile Mapping in OpenDRIVE und CityGML sowie Entwicklung geeigneter Visualisierungsmethoden. Technical University of Munich, Chair of Geoinformatics, Master's Thesis (in German). 2018. <https://mediatum.ub.tum.de/1451954>
- Kutzner, T.; Chaturvedi, K.; Kolbe, T.H. (2020).** CityGML3.0 – New Functions open up new Applications. In: PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science. <http://mediatum.ub.tum.de/node?id=1539643>
- Ruhdorfer, R.; Willenborg, B.; Sindram, M. (2018).** Coupling of Traffic Simulations and Semantic 3D City Models. *gis. Science*, 2018, Nr. 3/2018. <https://mediatum.ub.tum.de/1454442>
- Schwab, B.; Beil, C.; Kolbe T. H. (2020).** Spatio-Semantic Road Space Modeling for Vehicle–Pedestrian Simulation to Test Automated Driving Systems. *Sustainability* **12** (9), 3799. <https://www.mdpi.com/2071-1050/12/9/3799>
- Schwab, B.; Kolbe T. H. (2019).** Requirement Analysis of 3D Road Space Models for Automated Driving. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-4/W8, 99–106. <https://mediatum.ub.tum.de/node?id=1507292>