Overview

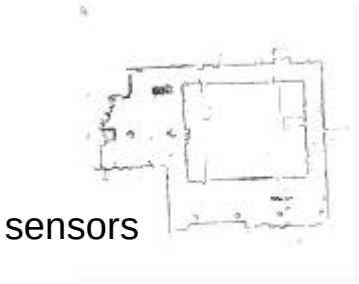
- Goal scenario of project A1-[RoboMap]
  - Different subproblems require the integration of quantitative and qualitative information
  - Safety requirements(I4-[SPIN]) and shared control(I3-[SharC])
- RouteGraph as an integrative data structure
  - Focus on a generic Implementation together with a formal Specification
  - RouteGraph exchange formats
- Examplary RouteGraphs
  - based on a navigation algorithm
  - cyclist observation
- Conclusion, future work

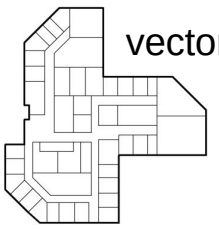
Goal scenario - motivation

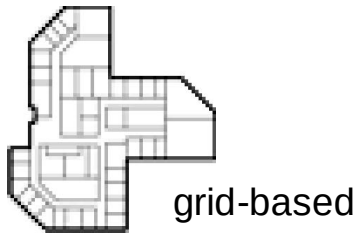
- Wheelchair as an autonomous vehicle, processing spatial knowledge from...
  - sensor data
  - speech input
  - pre-existing maps

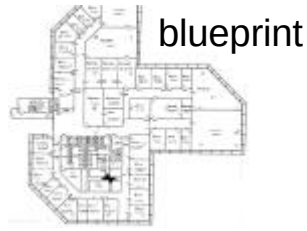
Goal scenario - tasks

- Map abstraction and interpretation
  - Use of sensor based mapping algorithms for indoor robot navigation
  - Abstraction of pre-existing maps to RouteGraphs
  - Realtime RouteGraph generation and interactive manipulation







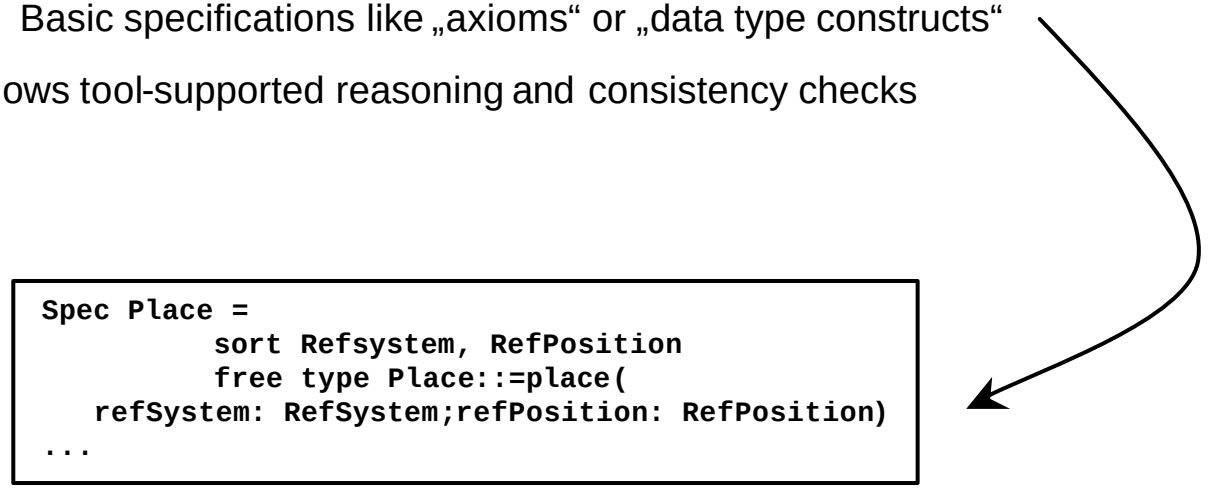


Goal scenario - tasks

- RouteGraph augmentation
  - Natural language statements about spatial relations
  - Combining qualitative and quantitative information for navigation
  - Merging RouteGraphs from different sources

Goal scenario - collaboration

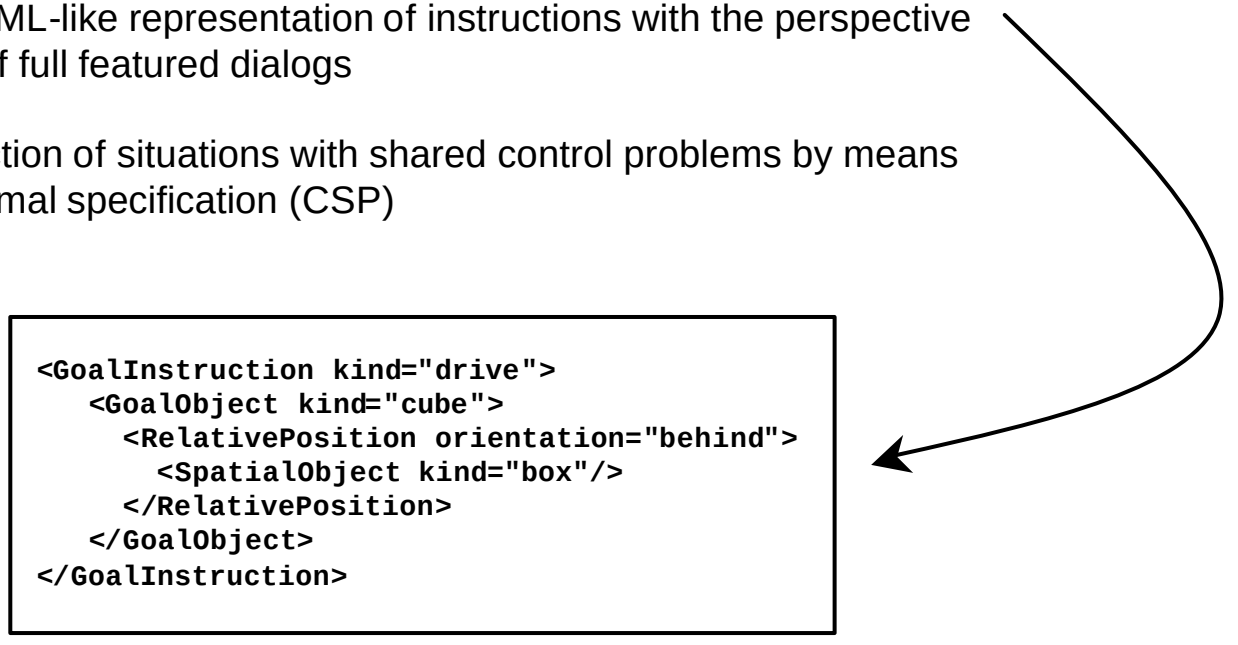
- I4-[SPIN]
  - Formal specification of RouteGraph concepts using CASL
    - Architectural specifications such as „implementation units“
    - Basic specifications like „axioms“ or „data type constructs“
  - Allows tool-supported reasoning and consistency checks



```
Spec Place =
          sort Refsystem, RefPosition
          free type Place::=place(
   refSystem: RefSystem;refPosition: RefPosition)
...
```

Goal scenario - collaboration

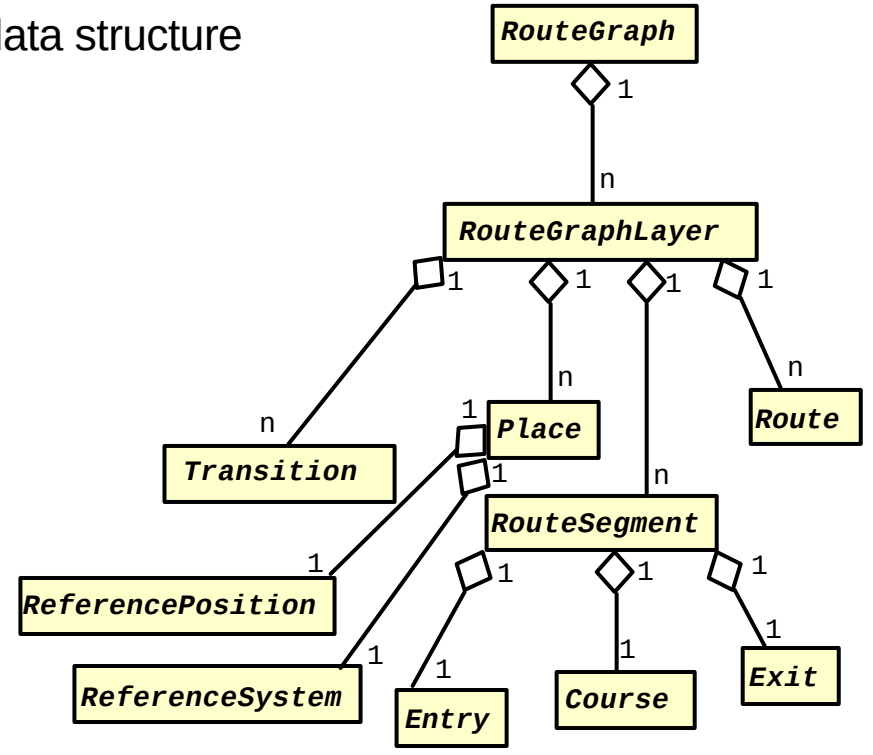
- I3-[SharC]
  - Providing formal descriptions of natural language dialog
    - XML-like representation of instructions with the perspective of full featured dialogs
  - Detection of situations with shared control problems by means of formal specification (CSP)



```
<GoalInstruction kind="drive">
   <GoalObject kind="cube">
     <RelativePosition orientation="behind">
       <SpatialObject kind="box"/>
     </RelativePosition>
   </GoalObject>
</GoalInstruction>
```

RouteGraph[1] as an integrative data structure

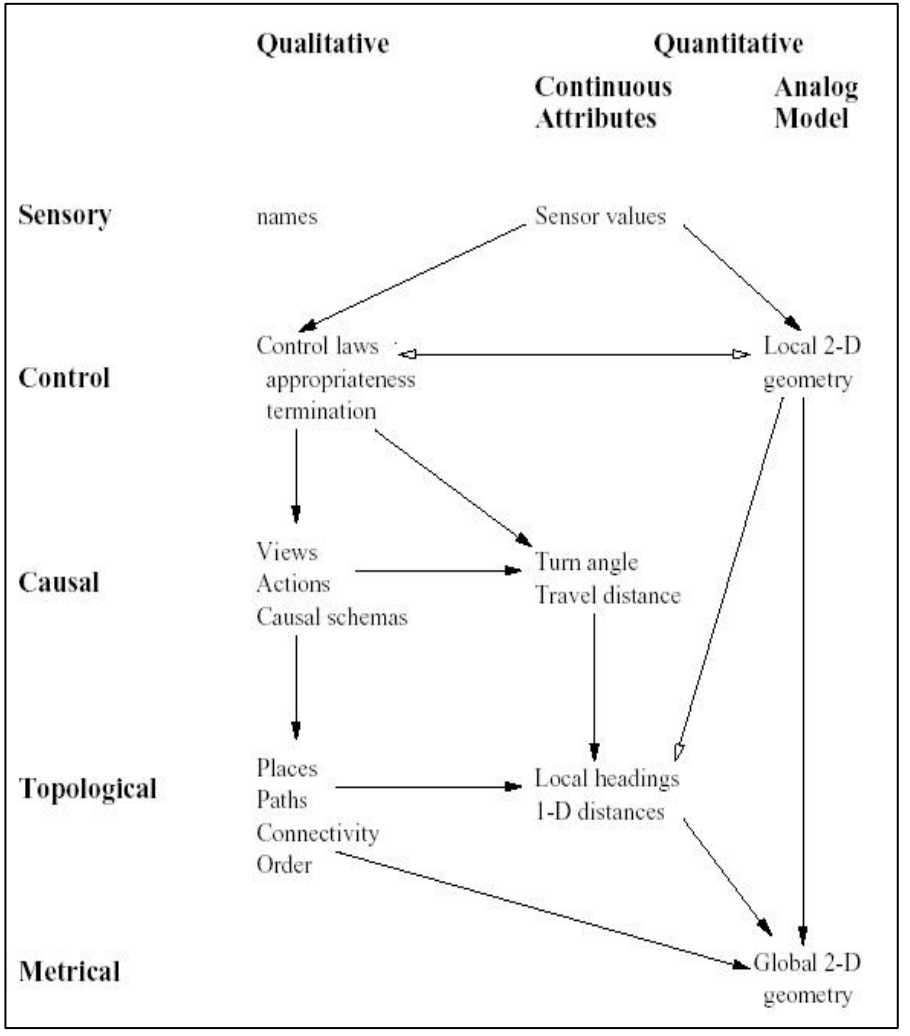
- Terminology
  - Kind
  - Place
  - Reference Position
  - Reference System
  - Route Segment
  - Entry
  - Exit
  - Course
  - Route
  - Transition
  - RouteGraph Layer
  - RouteGraph



[1] [Werner, Brückner, Herrmann] „Modelling Navigational Knowledge by Route Graphs“

RouteGraph as an integrative data structure

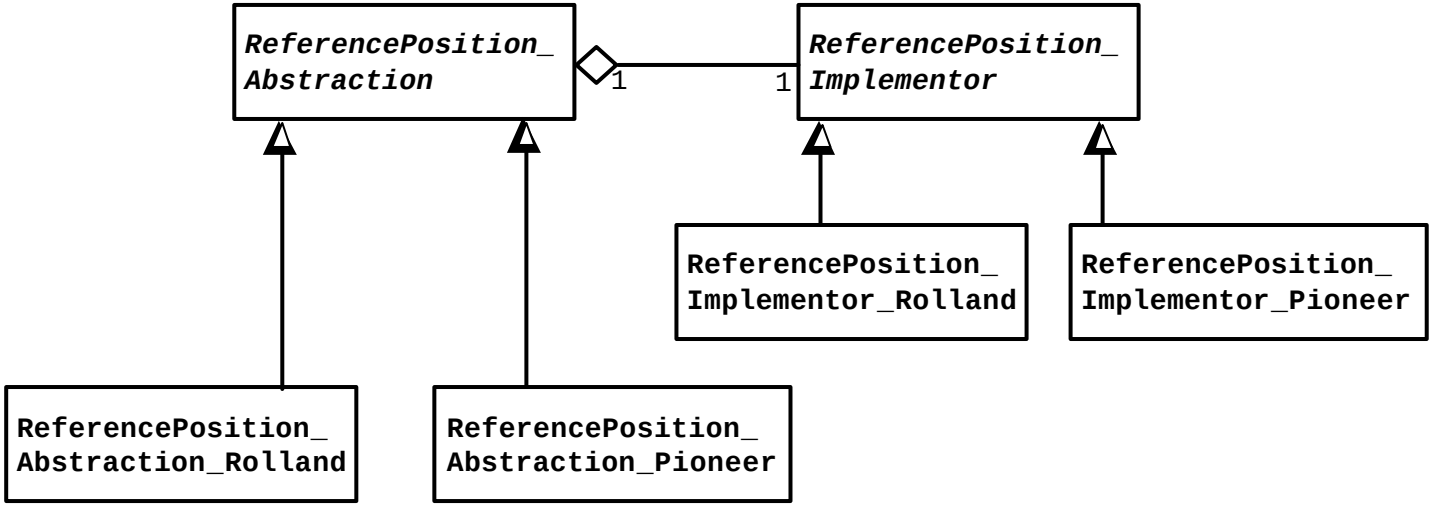
- Relations to previous work (Spatial Semantic Hierarchy[2])



[2] [Kuipers et al.] „The Spatial Semantic Hierarchy“

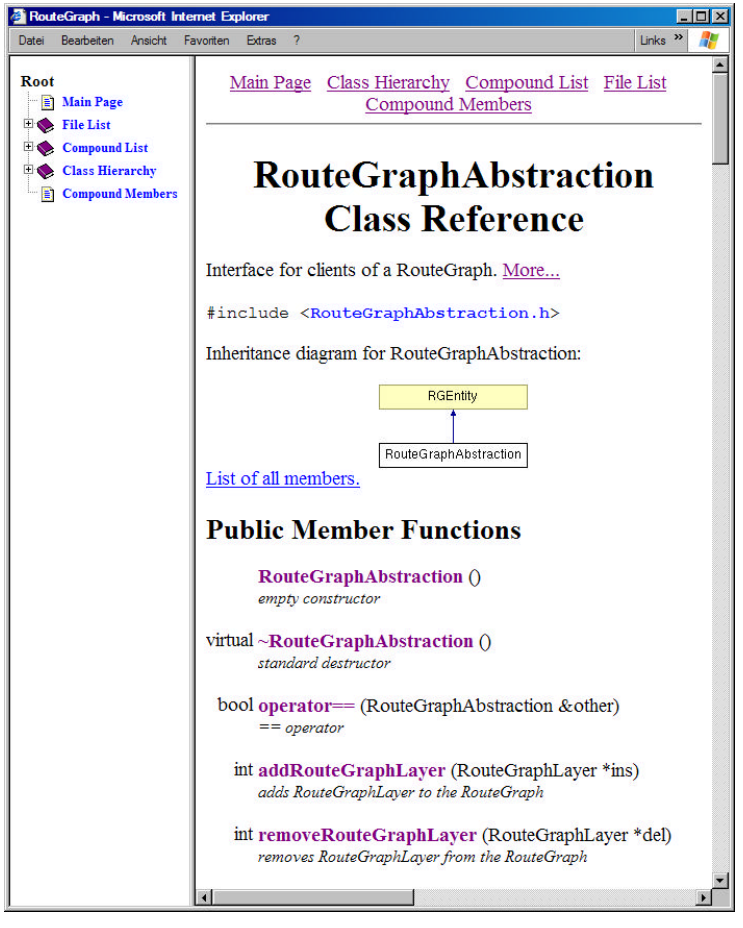
RouteGraph as an integrative data structure

- RouteGraph API
  - C++ implementation
  - Generic by means of consequential use of the bridge-pattern



RouteGraph as an integrative data structure

- RouteGraph API
  - HTML-based documentation simplifies porting to other platforms



RouteGraph as an integrative data structure

- RouteGraph exchange formats
  - XML – object tree
  - current tools supporting RouteGraph.XML :

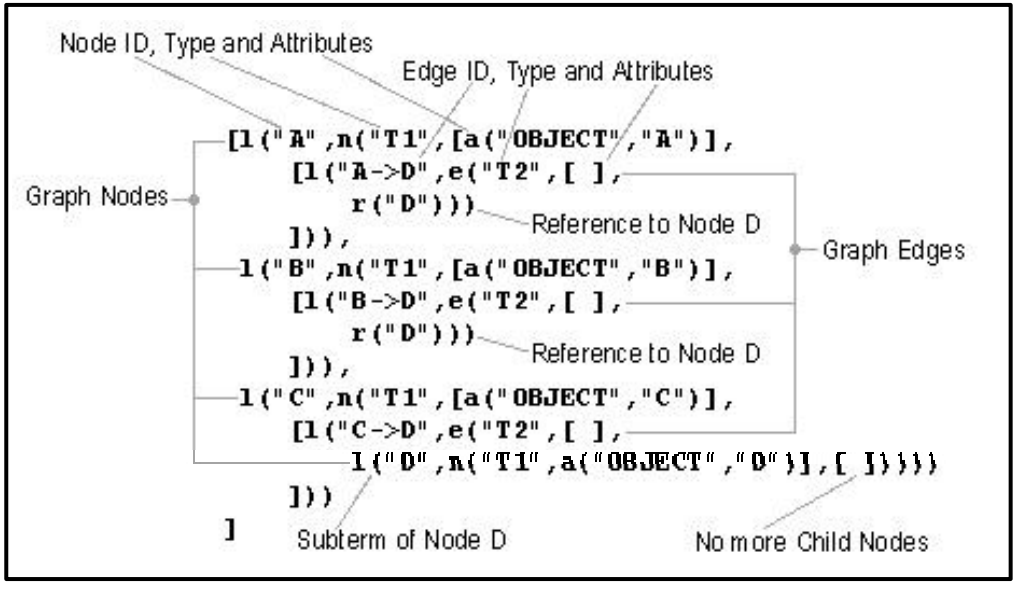
    XMLSpy
    (xml-viewer)

    Matlab
    (content visualization)

```
-<Place
    -Id="3"
    -Kind=KI_ROLLAND_METRIC">
    -<ReferencePosition
        -Id="1"
        -Kind="KI_ROLLAND_METRIC"
        -pos.driving_speed="0.000000"
        -pos.heading="0.000000"
        -pos.steering="0.000000"
        -pos.steering_speed="0.000000"
        -pos.x="0.000000"
        -pos.y="0.000000"
        -/>
-</Place>
```

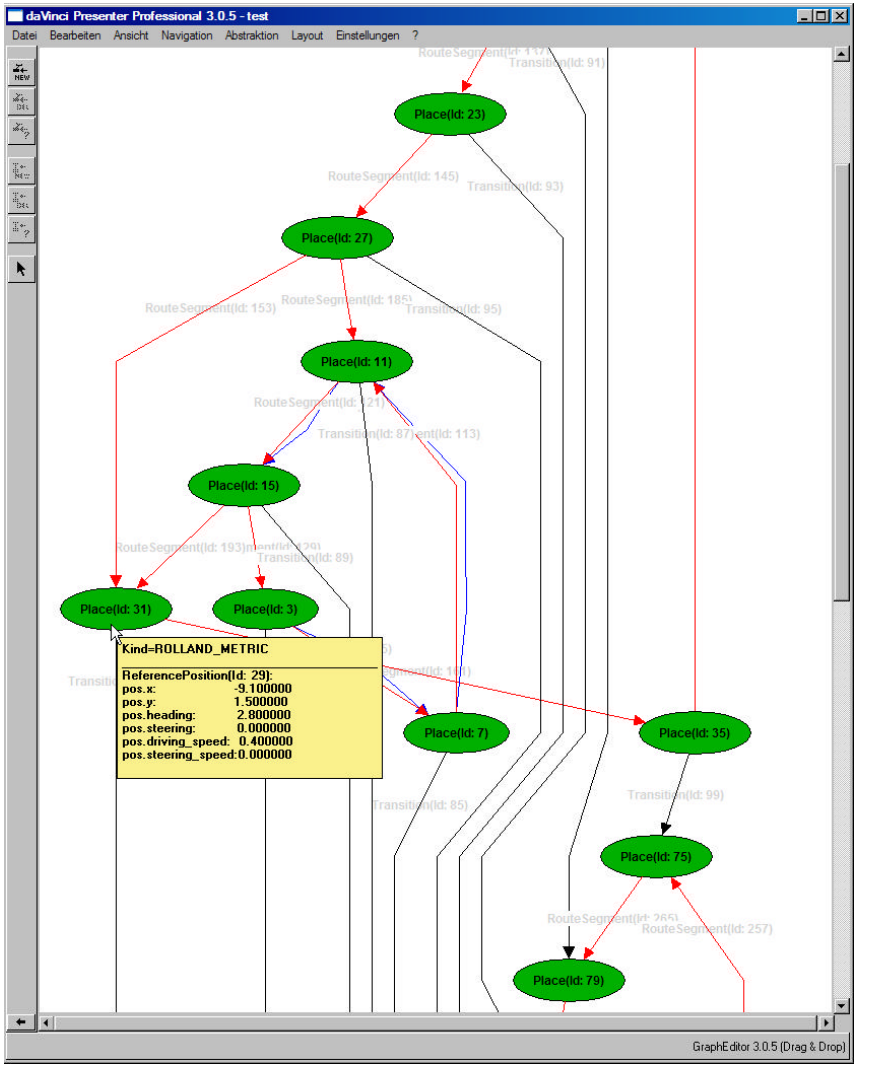
RouteGraph as an integrative data structure

- RouteGraph exchange formats
  - daVinci – term representation



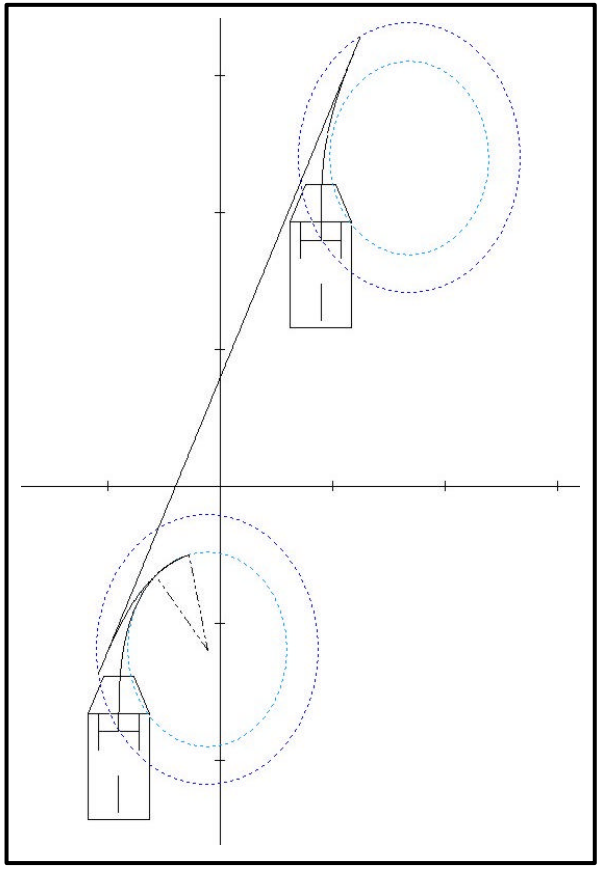
RouteGraph as an integrative data structure

- RouteGraph exchange formats
  - daVinci – example



Examplary RouteGraph

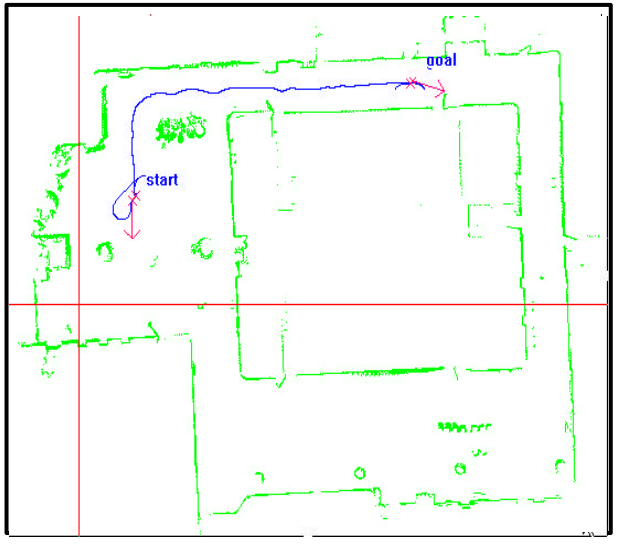
- Essential navigation algorithms (1):
  Steering with „Continuous Curvature Paths“[3]
  - Basic strategy for driving in obstacle free space
  - CCPs consist of circular arcs, clothoids and line segments
  - CCPs correspond to the nonholonomic properties of a car-like robot



[3] [Fraichard, Scheuer] „Planning continuous-curvature paths for car-like robots“

Examplary RouteGraph

- Essential navigation algorithms (2):
  Navigating with CCPs and Voronoi Diagrams
  - Extraction of navigational knowledge from sensoric input
  - OpenGL hardware allows fast computation of voronoi faces[4]
  - Edge detection via convolution filter
  - Solution of A* search in pixel space combined with start- and goalCCT

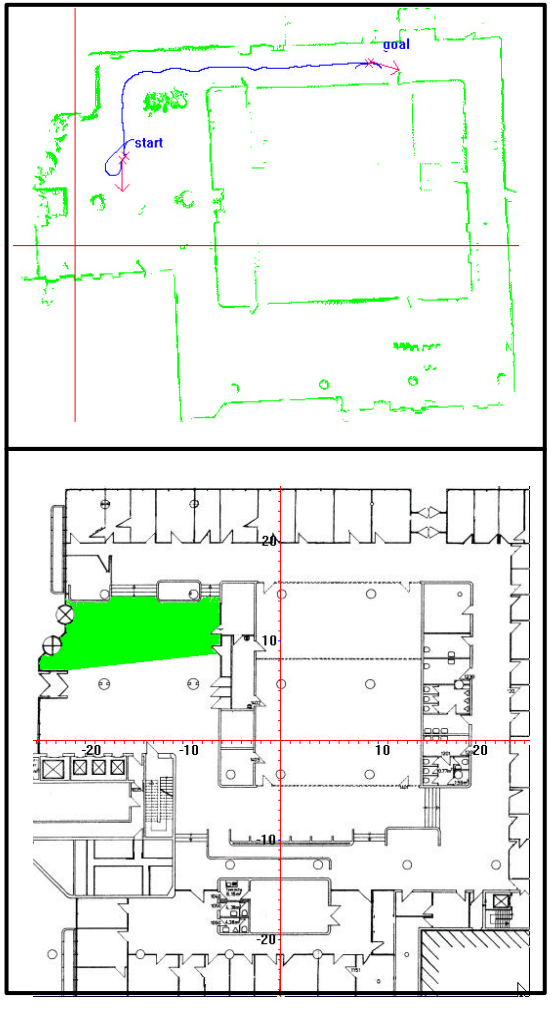


[4] [Kenneth et al.] „Fast computation of generalized voronoi diagrams using graphics hardware“

Examplary RouteGraph

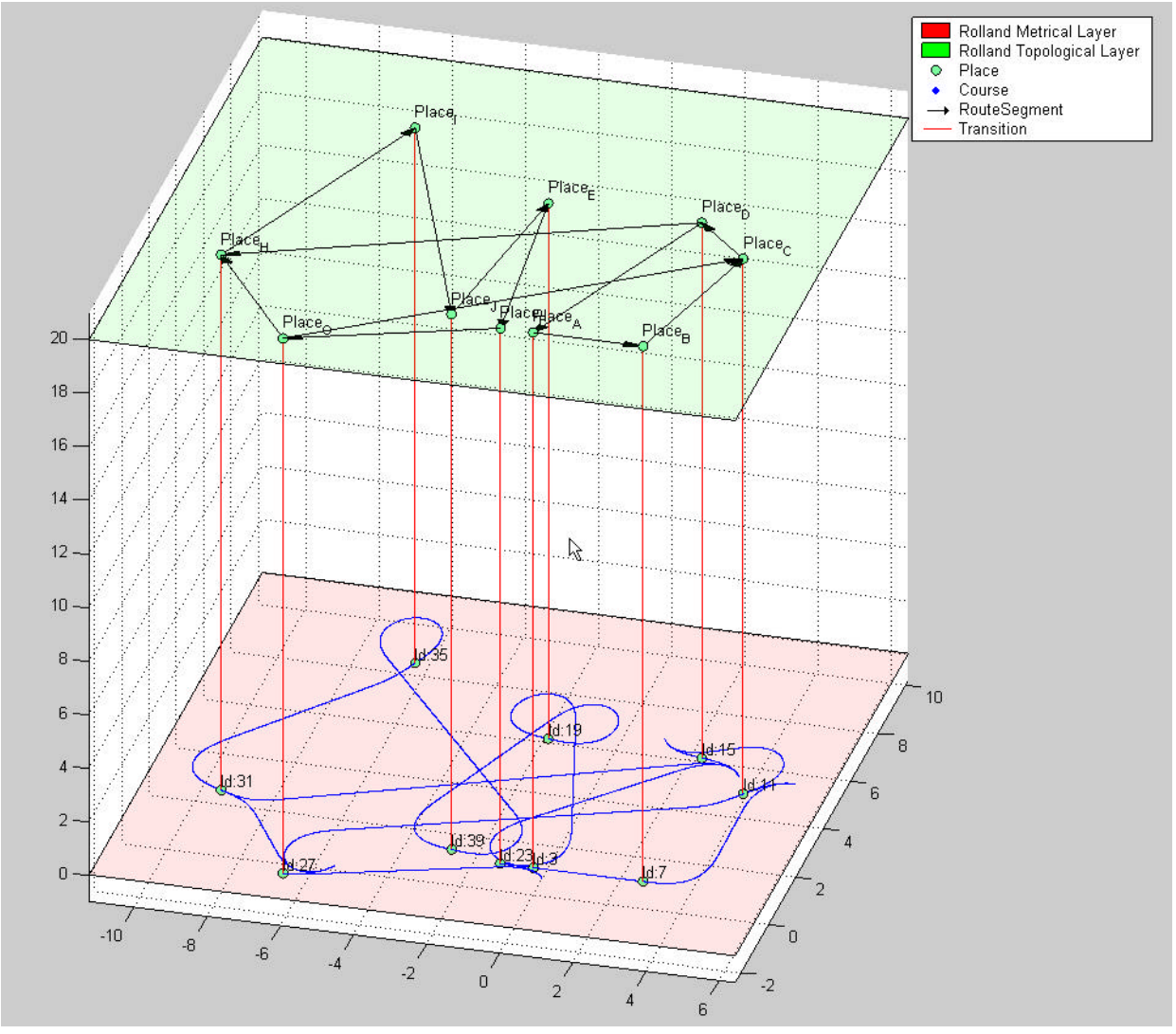
- Essential navigation algorithms (3): Lack of navigational knowledge when input is limited to real sensors

- Virtual laser scanner as an example for abstracting pre-existing maps to RouteGraphs



Examplary RouteGraph

Rolland
CCT-PathPlanner

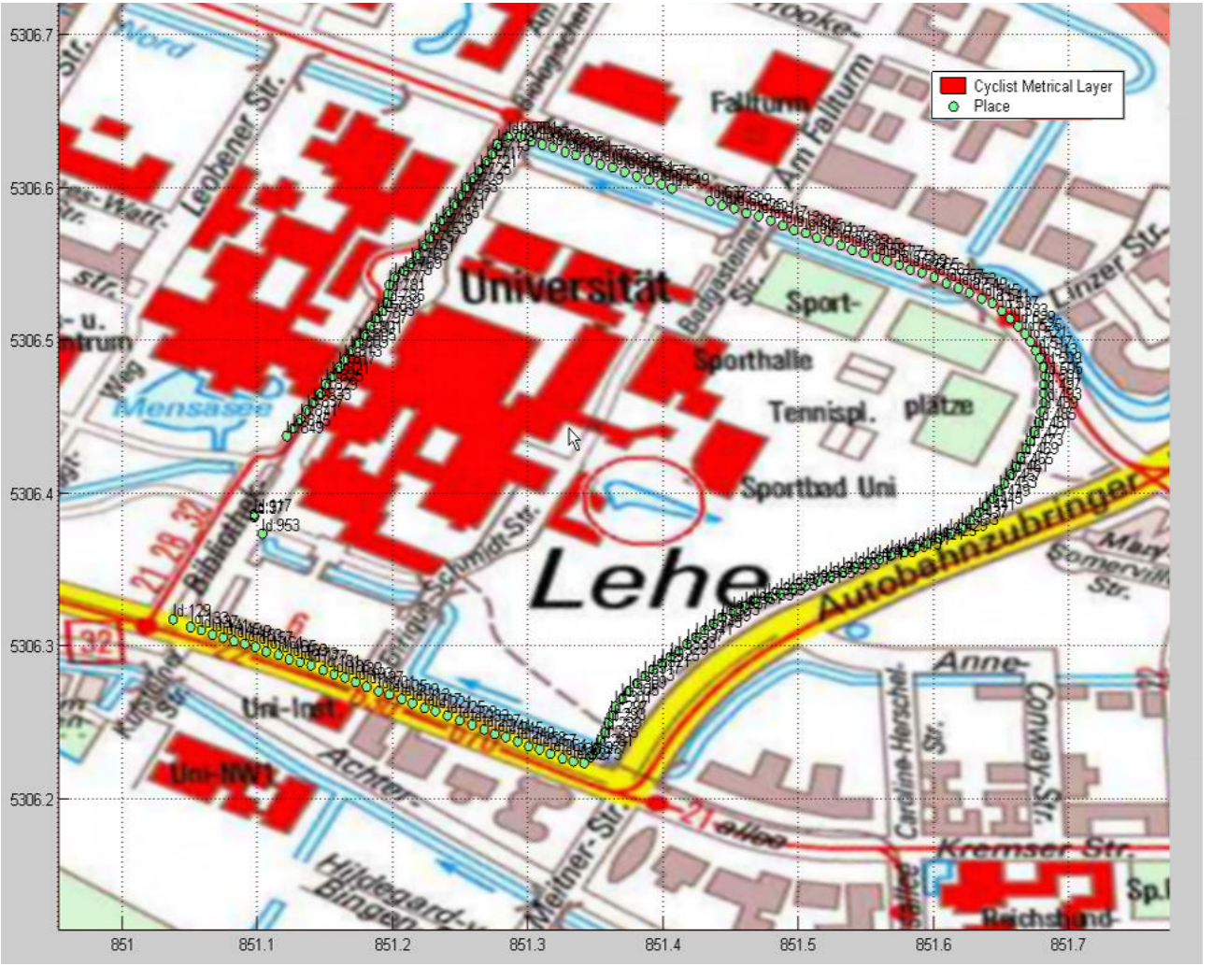


Examplary RouteGraph

Cyclist Navigation

GPS

Hand-Keyboard



Conclusion, future work

- RouteGraph as an agent independet structure for navigational tasks
- Current implementation supports metrical and topological layers
  - Integration of reasoning calculus within the topological layer
- Investigation of new Layers (dialog-layer, ...)
- Extraction of spatial knowledge from pre-existing maps (virtual laser scanner, ...)
  - Detection and Classification of architectural features within blueprints
- Support of RouteGraph-porting to other platforms and domains

Thank You ;-)