AIXM Procedure Seminar

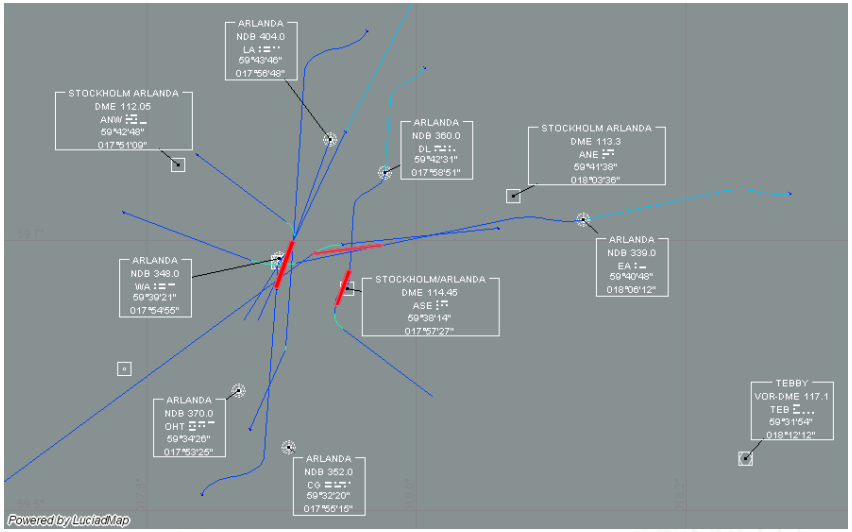
Luciad

Luciad introduction

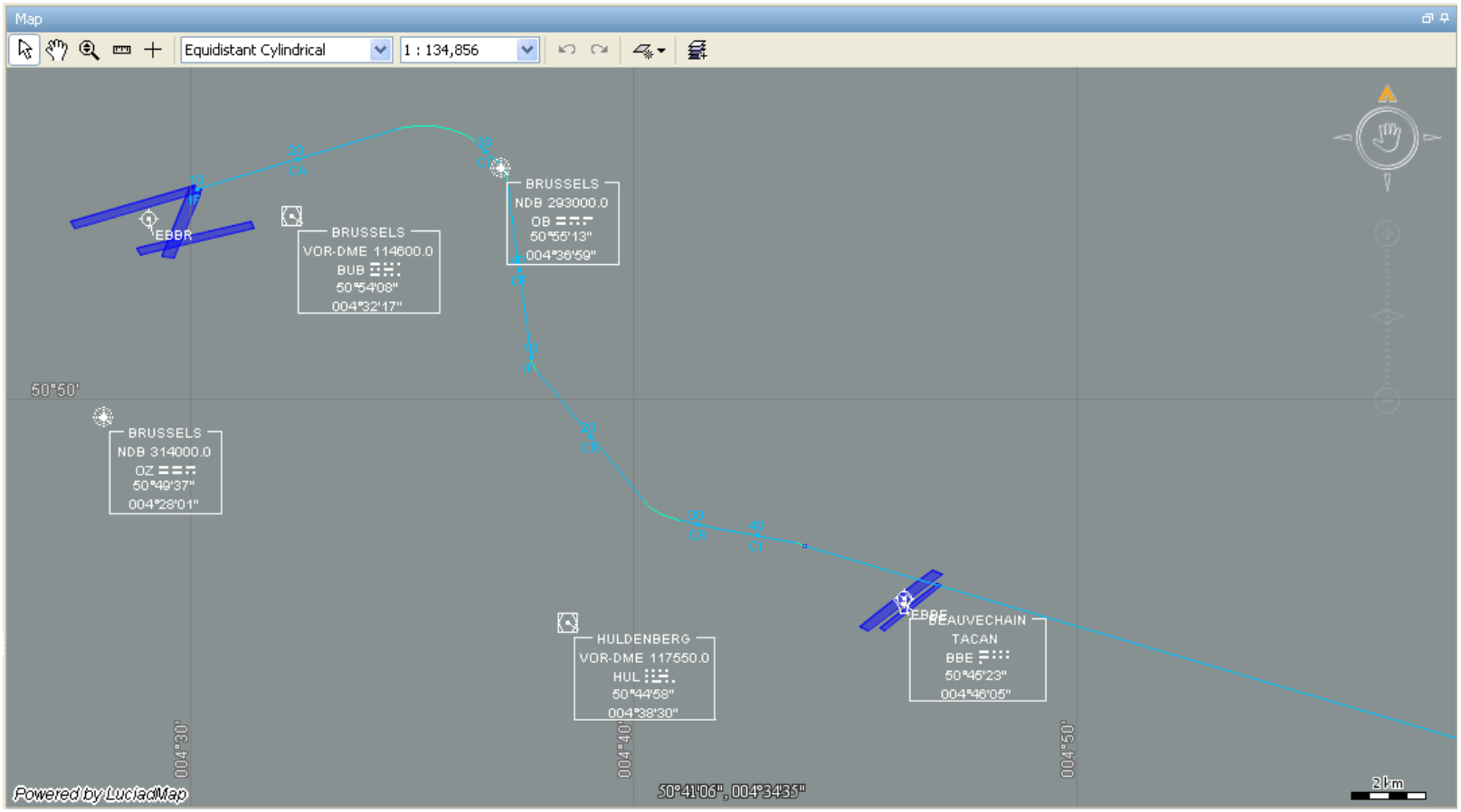
- **Software components for high performance visualization**
- **Key domains: Aviation, Defense and Security**
- **Adoption of aeronautical standards**
  - ARINC 424-13, 424-15, 424-18
  - AIXM 3.3, 4.0, 4.5, 5.0, 5.1
  - DAFIF Edition 7 and 8
  - ASTERIX
  - ASDI

Path and Terminator support (ARINC)

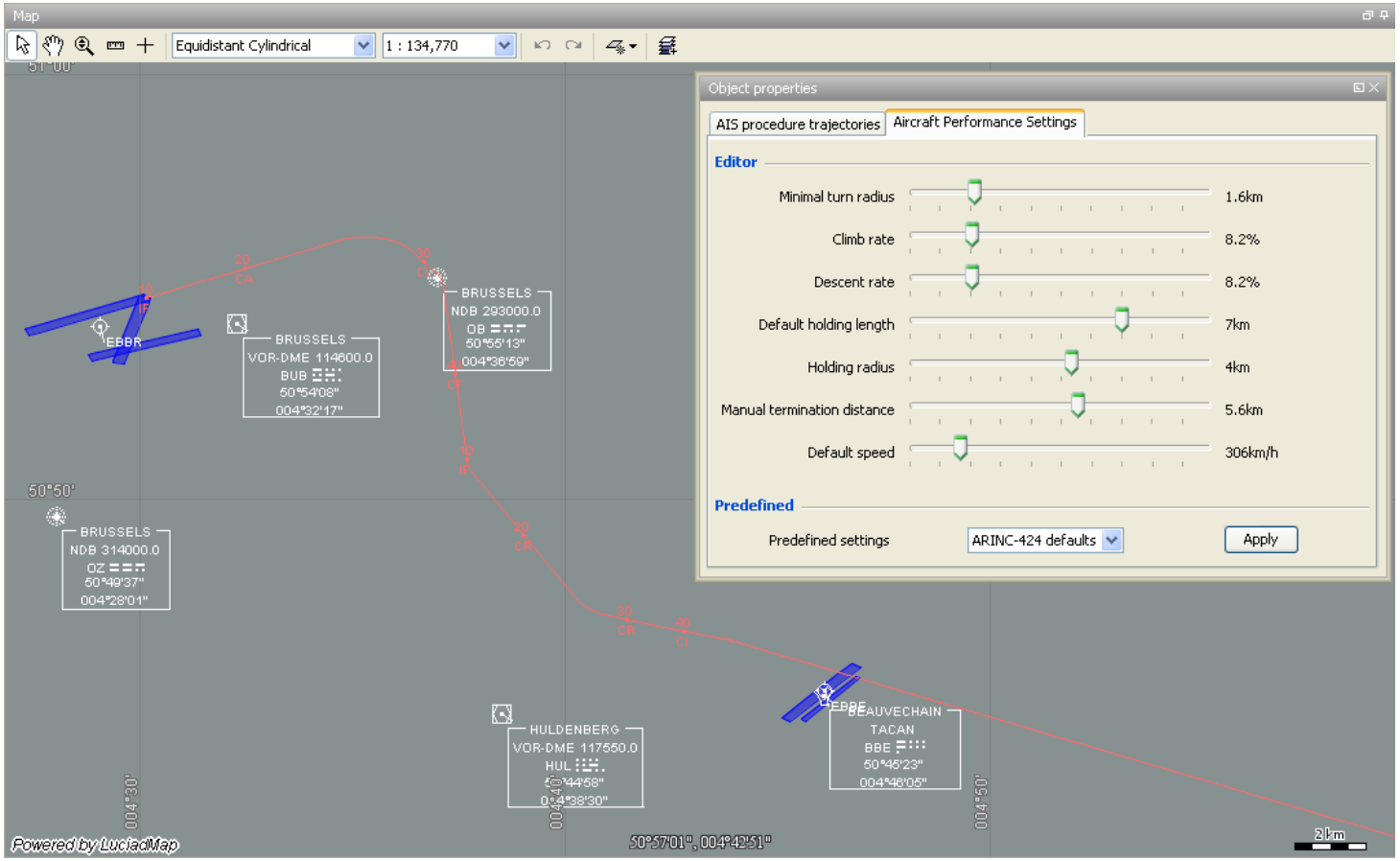
- **Adopted by ARINC 424, AIXM, DAFIF**
- **Software implementation**
  - Domain model to represent decoded procedure / procedure leg data
  - Algorithm to calculate the actual aircraft trajectory
    - Based upon procedure leg information
    - Taking into account aircraft performance settings



Procedure trajectory calculation example (1)



Procedure trajectory calculation example (2)



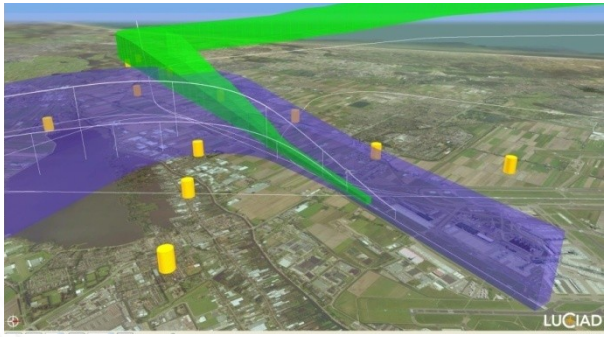
Path and Terminator support (ARINC)

- **Advantages**
  - Reduced number of waypoints
  - Close match with procedure charts
  - Decoupling of aircraft performance settings (e.g., convenient for simulation)

- **Disadvantages**
  - Calculation of the aircraft trajectory on a map (e.g., for simulation) is complex and costly
    - >> 20 procedure leg types
    - Complex procedures are easy to misinterpret
  - No performance model in most data formats

AIXM 5.x procedure support

- **New procedure model**
  - Procedure leg (path and terminator) modeling
  - Procedure trajectory (geometry) encoding
  - Performance model
- **More flexibility to support procedures**
  - Direct visualization of procedure trajectory
    - Useful for simple web clients or a quick preview
  - More advanced operations (analysis, simulation, …) possible through procedure leg/performance model information



AIXM 5.x procedure support

- **Experiments**
  - Data conversions from ARINC to AIXM 5.1
    - No real-world testdata yet
  - Procedure generation using both ARINC leg type information and calculated trajectory
- **Demonstrations in OGC OWS-6 and OWS-7**

```
<aixm:DepartureLegTimeSlice>
  <aixm:interpretation>BASELINE</aixm:interpretation>
  <aixm:legTypeARINC>CF</aixm:legTypeARINC>
  <aixm:course>87.05139231681824</aixm:course>
  <aixm:upperLimitAltitude uom="M">914.40</aixm:upperLimitAltitude>
  <aixm:lowerLimitAltitude uom="M">914.40</aixm:lowerLimitAltitude>
  <aixm:altitudeInterpretation>AT_LOWER</aixm:altitudeInterpretation>
  <aixm:endPoint>
    <aixm:TerminalSegmentPoint>
      <aixm:pointChoice_position>
        <aixm:Point>
          <gml:pos>-2.2255555555555557 49.47694444444445</gml:pos>
        </aixm:Point>
      </aixm:pointChoice_position>
    </aixm:TerminalSegmentPoint>
  </aixm:endPoint>
  <aixm:trajectory>
    <aixm:Curve>
      <gml:segments>
        <gml:LineStringSegment>
          <gml:posList>-2.2308943264287695 49.47676513538419
          -2.6477318124656692 49.462000110733754</gml:posList>
        </gml:LineStringSegment>
        <gml:ArcByCenterPoint>
          <gml:posList>-2.2297685267287077
          49.462452209403885</gml:posList>
          <gml:radius>1593.956776712</gml:radius>
          <gml:startAngle>1.622004315725133</gml:startAngle>
          <gml:endAngle>1.1445247816121895</gml:endAngle>
        </gml:ArcByCenterPoint>
      </gml:segments>
    </aixm:Curve>
  </aixm:trajectory>
</aixm:DepartureLegTimeSlice>
```

Conclusions

- **How to ensure data quality?**
  - XML schema allows too many degrees of freedom.
  - What data should always be available?
  - How to avoid ambiguities, conflicts?
    - e.g. not all leg types can occur everywhere.

- **Extensive model increases readability**
  - Compared with record data formats like ARINC
  - Equally extensive documentation can influence data quality