In 2012, researchers at Stanford (led by Walter Scheidel, with Elijah Meeks as a technical lead) developed ORBIS which offered a complex model of connectivity by reconstructing the duration and financial cost of travel in antiquity. Revealing the true shape of the Roman world, ORBIS provided a unique perspective on premodern history and became an object of envy for scholars working in other historical contexts. Since ORBIS was not designed to be easily adaptable to other contexts, a DH-team at the University of Vienna (led by Tara Andrews) organized a hackathon, where participants worked on a tool which historians with minimal DH skills could easily install and run, and, by supplying their own data, could explore their own historical networks in ways similar to ORBIS.

We used the al-Turayyā Project, as the sandbox, since it 1) re-uses a significant amount of code written in D3 for ORBIS 2.0, and 2) is modular enough to facilitate experimental development. Whereas the al-Turayyā Project used a modified version of the Dijkstra pathfinding algorithm, we chose to reduce the algorithmic complexity for OIB to the necessary minimum (i.e., dynamic network generation, weight modification, calculation of routes/networks using the “vanilla” Dijkstra algorithm), as not all potential modifications can be foreseen; historians will have full control over their networks through the modification of node/edge properties. With this approach, our application generates a network from supplied data, then continuously reconfigures it for specific queries by applying modifiers to edge weights (based on selected properties) and switching on/off specific nodes/edges; the visualization is then generated from the latest state of the network.

Figures in the middle provide an example. On [Figure-1], al-Turayyā shows the initial routes: RED-1 is the shortest route generated with the “vanilla” Dijkstra algorithm; GREEN-1 is the “optimal” route generated with a modified Dijkstra algorithm, searching for the next shortest route with a higher number of settlements along the way (under the assumption that such a route is safer). On [Figure-4], the OIB Sandbox shows two routes generated with the “vanilla” Dijkstra algorithm, but from differently configured networks: the RED-4 uses the initial network (~ RED-1); the GREEN-4 uses a reconfigured network: here, settlement type is applied as a modifier, making route sections that lead to larger settlements “shorter” and therefore preferable for the Dijkstra algorithm. While GREEN-1 (modified Dijkstra) offers a better alternative to RED-1—the suicidal option through the Syrian desert—GREEN-4 (modified network) not only avoids the desert, but also runs through all major cities in the region (Samarra/Samarrā > Mosul/al-Mawsil > Raqqah/al-Raqqat > Aleppo/Halab > Emeisa/Hims), a route usually found in medieval Arabic chronicles.

ORBIS is being developed as a modular application, whose functionality can be extended without any disruption (in addition to pathfinding, consider itinerary charting, flood networks with single and multiple centers, region modeling, etc.). The OIB requires users to supply CSV data files for edges and nodes, and to modify a YAML settings file. For example, the EDGES file should look as shown on [Figure-2], where RouteID, Start, End, Length, and Coordinates are required fields. In the network, Length is used by the Dijkstra algorithm to find the shortest route; additional fields — here Terrain and Safety — provide modifiers to Length values for network reconfiguration.

Additional fields are coded categorically and converted into numerical values via a config file such as shown on [Figure-3], through which one adjusts model parameters. Numeric values are used as multipliers for Length values. For example, the “weight” of the route section ID3 (25,432) becomes 63,580, if both Terrain and Safety modifiers are applied; that of the route section ID2 (34,567) becomes 25,925, which makes ID2 “shorter” than ID3, and therefore preferable within the “vanilla” Dijkstra algorithm. In a similar manner, nodes and edges of specific types can be excluded from the network.

Although this approach puts lots of weight on historians to produce appropriate data, it gives them the utmost freedom in modeling their research questions as well as makes OIB suitable for most use cases without any additional modifications of the tool itself. Much of the OIB Sandbox already works, yet an interface for dynamic network modification needs to be further developed, which we hope will happen in the near future.

**authors (alphabetically)**
- James Baillie
- Karl Grossner
- U Pittsburgh
- Maxim Romanov
- U Wien
- Masoumeh Seydi
- Leipzig U
- Rainer Simon
- AIT Austrian Institute of Technology
- Mária Vargha