Route data specification

This is the POSEIDON specification of route data for developers. A route specifies how to get from a start location to a destination. It needs to provide the instructions necessary to guide the person with Down's syndrome, and these typically need to be personalised by a carer. Route data can be used to present the route, in rehearsing the trip, and in real-time navigation guidance and tracking. The route data is stored as structured data in a JSON file, plus associated media files for the instructions. In the POSEIDON prototype system, these files are stored on the POSEIDON file server and accessed by all the applications (stationary, web and mobile).

Data model

As a basis for the route data, we have a generic conceptual data model, to capture the essence of what is produced by typical route planners and needed for route presentation and navigation. This model is based on route data formats of major route planning services, specifically OpenTripPlanner[1] and Google Directions[2]. The routes produced by these two systems are similar on the conceptual level, although they differ in some details. Adopting a similar model is done both because it is a proven approach, and because it facilitates interoperability, making it possible to use the same navigation algorithms etc. for personalised POSEIDON routes and routes from the route planning services. It is a hierarchical model, shown in Figure 9.

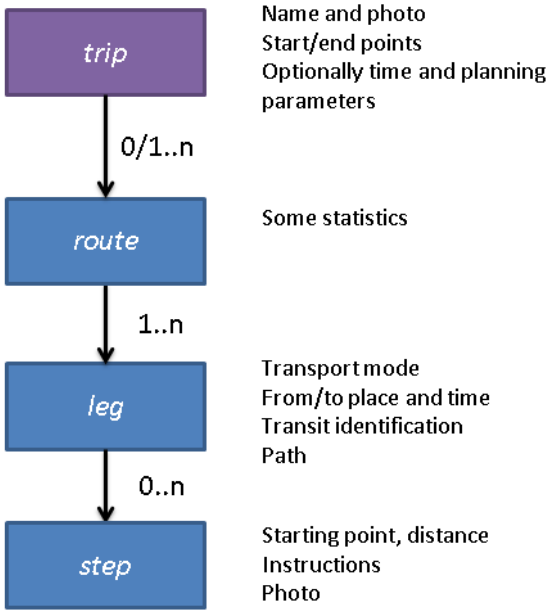

Figure 1: Route data model

The top-level element of our model is *trip*, which is a header or abstract definition for route data, as opposed to the route itself. It can represent meta-data for a planned route, or input parameters for route planning. As a minimum, it specifies a destination point, so that it can be used to get a route from wherever the person is currently located from an automated route planning service. It can also specify a start point, a time for the start or destination, preferred transport modes, and other parameters for route planning. A list of preferred routes or destinations for the user to choose from is a list of *trip* elements. For rehearsal or navigation, a *route* for the trip is necessary. There can be several *routes* for a trip in some cases – a route planning service may provide several alternative routes, and they can be processed until one is chosen for navigation.

The *route* element represents an actual route – the way from a starting point to a destination. The *route* element itself is mainly a container for a set of *legs*, though it can also have some overall statistics. The route is made up of *legs*, which is a way to partition the route into main parts, for instance for different modes of transport. Each leg specifies a transport mode, such as walking or bus, so a route should be partitioned into multiple legs if there are multiple transport modes involved. The main distinction in transport modes is between active and passive modes. Active

[1] http://www.opentripplanner.org/
[2] https://developers.google.com/maps/documentation/directions/

modes define places where the traveller needs to actively navigate. For our prototypes we have considered just walking, but cycling and driving are other active modes. Passive modes refer to places where the traveller is a passenger (i.e. the public transport modes). The ways to provide navigation and actions for deviation detection are different between active and passive modes.

A leg can further be partitioned into *steps*. Active mode legs need step information for giving step-by-step instructions. The step is the unit of instruction, specifying a location for where to give the instruction. Therefore, a step is typically located where there is a change of direction, a need to cross the street, etc. Given the limitations in accuracy of satellite positioning, there should be a lower limit on how close together steps can be placed. Steps placed just a few meters apart will not work well in navigation, as the satellite positioning is likely to place the user at the wrong step. We recommend a minimum distance of 50 meters. For transit legs, steps can be used to define the intermediary stops on the transit route (i.e. telling the user how many stops are left before the destination), and/or give a notification before reaching the destination. The instruction can include a text string, an image for the step (i.e. show a photo of the road to the user) and sound (for voice recording).

While the step locations make up a route to follow, the route data may also specify a *path*. The path is a list of geographical coordinates, typically based on map data and more detailed than the start/end coordinates of steps and legs. Google Directions and OpenTripPlanner include a path in their output for drawing the route on a map, and this will be as detailed as the underlying map data, typically following every bend in the road. As we do not require use of automated route planners, our model does not require a path, but it is included in the conceptual model, and path segments can be provided on either the leg or step level.

Our route data model is meant to be quite flexible, so that routes can be more or less well-defined, they can have time schedules or not, etc. A route defined by a carer will typically have more instruction properties but less formal data such as timing and statistics, and we want applications to be able to use both. Navigation algorithms should make a best effort to guide the user with what data is available.

Data file specification

For storing and exchanging routes between applications, we define a JSON format with required and optional fields. The main objects are:

```
{
      meta:{…},
      routes:[
            legs:[
                  steps:[…]
            ]
      ]
}
```

The fields of each object are described here. See Appendix 2 for an example JSON file.

Media for instructions (image and sound) can be specified for each step. These are specified with URIs pointing at the resources (image and sound files). Resources stored on the POSEIDON file server are identified by a resource ID. For resources stored on this server, a resource ID can be given instead of a full URI, so that the file server address can be changed without affecting the route.

Relation to existing formats

This format is based on that used by output from Google Directions (GD). It has the same structure of routes containing legs containing steps. Our specification for these elements is a subset of that used by GD. In addition we have custom fields for multimedia instructions.

A summary of the main differences from GD data:

- We have a *meta* object in addition to routes, corresponding to the *trip* level of our conceptual model.
- GD data may have two levels of steps (a list of steps inside a step), while our specification is for one level only (use legs for a high-level partitioning).
- We add step properties for text, image and sound instructions.

Our conceptual model is closer to the output of OpenTripPlanner than it is to GD. Our distinction between leg and step is the same as in OTP, and OTP specifies transport mode on legs and only has one level of steps. An application parsing routes in our and/or the GD format can therefore also be adapted to parse OpenTripPlanner routes with little effort.

meta object

The meta object corresponds to the trip level of the conceptual model, and gives information about the route without being part of the route. The required properties have been selected so that applications only need to parse the meta object, not the route itself, to be able to present a short summary of the route.

JSON identifier	Datatype	Required	Comments
`title`	`string`	Yes	Name for presenting the route in user interfaces.
`start_location`	`string`	Yes	Name of starting point, to tell the user where they need to start from.
`end_location`	`string`	Yes	Name of destination, to tell the user where the route goes.
`start_longitude`	`double`	Yes	Coordinate of starting point.
`start_latitude`	`double`	Yes	Coordinate of starting point.
`end_longitude`	`double`	Yes	Coordinate of destination.
`end_latitude`	`double`	Yes	Coordinate of destination.
`resource`	`string`	No	URI or resource ID for image representing the route.

route object

The JSON file has an array of route objects at the top level, supporting alternative routes as Google Direction does, although we only use single routes in the prototypes. The POSEIDON format only requires an array of leg objects at this level.

leg object

As with the leg level of the conceptual model, each leg object represents a part of the route. Note that, to be compatible with Google Directions, the travel mode is placed in each step. No fields are required, but a meaningful definition of a leg needs either a set of steps or a start location, as a minimum.

JSON identifier	Datatype	Required	Comments
distance	object	No	This object is not required, but if included must have a field *value* with the distance to travel, in meters.
duration	object	No	This object is not required, but if included must have a field *value* with the planned time duration of the leg, in seconds.
start_address	string	No	Street address at start of leg.
end_address	string	No	Street address at end of leg.
start_location	object	No	Coordinates of starting point. This is an object with two properties – lat and lng – each a floating-point coordinate value.
end_location	object	No	Coordinates of destination. Same type of object as start_location.
steps	array	No	List of step objects.

step object

As with the step level of the conceptual model, each step object represents an instruction and the part of the route it applies to. While none of the instruction properties are required, at least one of them should be given for the step to be useful.

JSON identifier	Datatype	Required	Comments
distance	object	No	This object is not required, but if included must have a field *value* with the distance to travel, in meters.
duration	object	No	This object is not required, but if included must have a field *value* with the planned time duration of the step, in seconds.
travel_mode	string	Yes	The prototype system uses the values WALKING and TRANSIT, distinguishing between a mode where the user is actively navigating and a mode where he is a passenger.
start_location	object	Yes	Coordinates of step point – the point the instruction pertains to. This is an object with two properties – lat and lng – each a floating-point coordinate value.
customFilePath	string	No	URI or resource ID for image for the instruction.
customTextInstructions	string	No	Instruction text.
customAudioPath	string	No	URI or resource ID for sound for the instruction.
html_instructions	string	No	Instructions provided by Google Directions, included in the specification for compatibility. Note that it may include html tags.
polyline	object	No	If included, this object must have a property *points,* which is an encoded

			polyline bean for the path (Google Directions provides this).

Appendix: Route data files

This appendix lists a formal definition and an example of route data.

Route file schema for POSEIDON mobile application

The prototype mobile application uses a formal definition to do validation and parsing of route files. This definition is in a proprietary XML format, which is not JSON-specific, but it is included here as it is a formal definition in a straightforward form.

```
<group name="root">
  <group name="meta" cardinality="1">
    <item name="title" type="string" cardinality="1"/>
    <item name="start_location" type="string" cardinality="1"/>
    <item name="end_location" type="string" cardinality="1"/>
    <item name="start_longitude" type="double" cardinality="1"/>
    <item name="start_latitude" type="double" cardinality="1"/>
    <item name="end_longitude" type="double" cardinality="1"/>
    <item name="end_latitude" type="double" cardinality="1"/>
    <item name="resource" type="string" cardinality="0-1"/>
  </group>
  <group name="routes" cardinality="1-n">
    <group name="legs" cardinality="1-n">
      <group name="distance" cardinality="0-1">
        <item name="value" type="integer" cardinality="1"/>
      </group>
      <group name="duration" cardinality="0-1">
        <item name="value" type="integer" cardinality="1"/>
      </group>
      <item name="start_address" type="string" cardinality="0-1"/>
      <item name="end_address" type="string" cardinality="0-1"/>
      <group name="start_location" cardinality="0-1">
        <item name="lat" type="double" cardinality="1"/>
        <item name="lng" type="double" cardinality="1"/>
      </group>
      <group name="end_location" cardinality="0-1">
        <item name="lat" type="double" cardinality="1"/>
        <item name="lng" type="double" cardinality="1"/>
      </group>
      <group name="steps" cardinality="0-n">
        <group name="distance" cardinality="0-1">
          <item name="value" type="integer" cardinality="1"/>
        </group>
        <group name="duration" cardinality="0-1">
          <item name="value" type="integer" cardinality="1"/>
        </group>
        <item name="travel_mode" type="string" cardinality="1"/>
        <group name="start_location" cardinality="1">
          <item name="lat" type="double" cardinality="1"/>
          <item name="lng" type="double" cardinality="1"/>
        </group>
        <item name="customFilePath" type="string" cardinality="0-1"/>
        <item name="customTextInstructions" type="string" cardinality="0-1"/>
        <item name="customAudioPath" type="string" cardinality="0-1"/>
        <item name="html_instructions" type="string" cardinality="0-1"/>
        <group name="polyline" cardinality="0-1">
          <item name="points" type="string" cardinality="1"/>
        </group>
      </group>
    </group>
  </group>
```

</group>

Example route JSON

The following is an example of a JSON route according to POSEIDON specification.


```
{
   "meta":{
              "title":"Test route",
              "start_location":"Karde",
              "end_location":"Problemveien",
              "start_longitude":10.7163861111111,
              "start_latitude":59.9438666666667,
              "end_longitude":10.72225,
              "end_latitude":59.9430305555556,
              "resource":"9fbe5760-90be-4f4e-b886-8d24a35b11f6"
   },
   "routes":[
      {
         "legs":[
            {
               "distance":{
                  "text":"0,5 km",
                  "value":"488"
               },
               "duration":{
                  "text":"6 min",
                  "value":"324"
               },
               "end_address":"Problemveien",
               "end_location":{
                  "lat":"59.9430305555556",
                  "lng":"10.72225"
               },
               "start_address":"Karde",
               "start_location":{
                  "lat":"59.9438666666667",
                  "lng":"10.7163861111111"
               },
               "steps":[
                  {
                     "distance":{
                        "text":"0,5 km",
                        "value":"488"
                     },
                     "duration":{
                        "text":"6 min",
                        "value":"324"
                     },
                     "travel_mode":"WALKING",
                     "start_location":{
                        "lat":"59.9438666666667",
                        "lng":"10.7163861111111"
                     },
                     "end_location":{
                        "lat":"59.9438666666667",
                        "lng":"10.7163861111111"
                     },
                     "customFilePath":"c3f0ec11-c98e-4e1a-9726-c52fa7ca00d9",
                     "customTextInstructions":" ",
                     "customAudioPath":" "
                  },
                  {
                     "distance":{
                        "text":"0,5 km",
                        "value":"488"
                     },
                     "duration":{
```

```
      "text":"6 min",
      "value":"324"
   },
   "travel_mode":"WALKING",
   "start_location":{
      "lat":"59.9431138888889",
      "lng":"10.71595"
   },
   "end_location":{
      "lat":"59.9431138888889",
      "lng":"10.71595"
   },
   "customFilePath":"30804632-005b-4d4b-9042-6f908c2017e8",
   "customTextInstructions":" ",
   "customAudioPath":" ",
   "polyline":{
      "points":"mszlJ{}k`AXT@UMGOIDg@PqB\\yA"
   }
},
{
   "distance":{
      "text":"0,5 km",
      "value":"488"
   },
   "duration":{
      "text":"6 min",
      "value":"324"
   },
   "travel_mode":"WALKING",
   "start_location":{
      "lat":"59.9428361111111",
      "lng":"10.717275"
   },
   "end_location":{
      "lat":"59.9428361111111",
      "lng":"10.717275"
   },
   "customFilePath":"69ff0f32-511c-4b05-b9e3-7cb640d86445",
   "customTextInstructions":" ",
   "customAudioPath":" ",
   "polyline":{
      "points":"yqzlJafl`ABOLRFMz@uCp@}BL{@DiB@q@TkAPu@"
   }
},
{
   "distance":{
      "text":"0,5 km",
      "value":"488"
   },
   "duration":{
      "text":"6 min",
      "value":"324"
   },
   "travel_mode":"WALKING",
   "start_location":{
      "lat":"59.9418722222222",
      "lng":"10.7204722222222"
   },
   "end_location":{
      "lat":"59.9418722222222",
      "lng":"10.7204722222222"
   },
   "customFilePath":"f03c983f-c186-42d9-b7df-19dbe48fd2a3",
   "customTextInstructions":" ",
   "customAudioPath":" ",
   "polyline":{
      "points":"skzlJyyl`AHm@@OKH[@UKa@w@sBuD_@q@"
   }
```

```
                    },
                    {
                       "distance":{
                          "text":"0,5 km",
                          "value":"488"
                       },
                       "duration":{
                          "text":"6 min",
                          "value":"324"
                       },
                       "travel_mode":"WALKING",
                       "start_location":{
                          "lat":"59.9430305555556",
                          "lng":"10.72225"
                       },
                       "end_location":{
                          "lat":"59.9430305555556",
                          "lng":"10.72225"
                       },
                       "customFilePath":"9fbe5760-90be-4f4e-b886-8d24a35b11f6",
                       "customTextInstructions":" ",
                       "customAudioPath":" ",
                       "polyline":{
                          "points":"{rzlJ_em`A"
                       }
                    }
                 ]
              }
           ]
        }
     ]
}
```