Information Modelling and Knowledge Bases XXXII
M. Tropmann-Frick et al. (Eds.)

doi:10.3233/FAIA200817

TrailSigner: A Conceptual Model of Hiking Trail Networks with Consistent Signage Planning and Management

András J. MOLNÁR
Christian-Albrechts-University Kiel,
Department of Computer Science, D-24098 Kiel, Germany;
E-mail: ajm@informatik.uni-kiel.de
SZTAKI Institute for Computer Science and Control, H-1111 Budapest, Hungary;
E-mail: modras@ilab.sztaki.hu

Abstract.
Trail route networks provide an infrastructure for touristic and recreational walking activities worldwide. They can have a variety of layouts, signage systems, development and management patterns, involving multiple stakeholders and contributors, and tend to be determined by various interests on different levels and dynamically changing circumstances. This paper aims to develop the skeleton of TrailSigner, a sound geospatial conceptual data model suite of trail networks, waymarked routes and their signage systems and assets, which can be used as a basis for creating an information system for the effective, organic and consistent planning, management, maintenance and presentation of trails and their signage. This reduces potential confusion, mistrust and danger for visitors caused by information mismatches including incomplete, incoherent or inconsistent route signposting. To ensure consistency of incrementally planned signposts with each other and with the (possibly changing) underlying trail network, a systematic, set-based approach is developed using generative logical rules and incorporated into the conceptual model suite as signpost logics. The paper also defines a reference ruleset for it. This approach may further be generalized, personalized and adapted to other fields or applications having similar requirements or phenomena.

Keywords.
conceptual modeling, model suite, tourism and recreation, hiking trails, route network, signpost system, spatial data modeling, spatial information system, knowledge and information modeling, knowledge representation and management, collaborative information system, planning and facility management, change management, logical representation and reasoning

1. Introduction and Outline

Trail networks provide an infrastructure for touristic and recreational walking (hiking, rambling, trekking, mountaineering, etc.), biking or running activities, mainly in natural but also in built-up areas. Routes are based on publicly passable paths and road sections in the field [1, Walking Routes]. Waymarking, signposting on the field and presentation of the routes on maps, in apps and other media can effectively, reliably and safely support the visitors in planning their trips and their orientation on the terrain, if done properly. In this paper, we consider walking routes only, but the observations can be generalized for trails of other activities.

Hiking and walking trails have been developed in the last two centuries in many countries, and a substantial growth in the recent decades can be observed all around the world. New countries and regions are developing their trails, and existing networks face challenges of a higher demand for visitor management and effective maintenance. Slow tourism and outdoor recreation contributes to well-being and sustainability in many ways. With new target groups being involved with less expertise, proper information provision becomes more and more crucial. Community mapping has been growing in the last decades, websites and apps about trails are almost uncountable, and the first integrated platforms for trail network and facility management have been appearing on the market in the these years.

Even in our digital age, when navigation systems are in widespread use, the on-field information provided in various forms such as waymarks, signposts or info-panels, as well as off-line maps, guidebooks and other media seem to be essential. It provides a ceratin safety against possible technical failures or limitations (e.g. precision or battery power). More importantly, 'switching-off' and 'detoxication' from digital devices is one of the significant motivations for going out to trails [13,6]. The parallel development of digital tools with off-line information assets raises challenges not only towards the effective and direct support of trail users, but also the systematic and consistent management of trail facilities themselves and their related off-line information assets.

A recreational and touristic trail network is a complex reality, and different terminologies co-exist in different regions around the world, or even among visitors and professionals with different backgrounds in the same region (forestries, national park or other conservation authorities, tourism agencies, hiking or mountaineering associations, etc). Routes and their networks may have various layouts and signage systems. In many areas, routes or even networks with different thematics co-exist in parallel, lacking a unified network concept or signage system.

Trail development and management has its challenges, caused by various ownership, stewardship, institutional and societal patterns, and reflects many aspects of the local, regional and national context and its recent evolution. Actions are not always systematic, but often ad-hoc or step-by-step, initiated and implemented by some of stakeholders not necessarily considering the aspects of the whole network. Resources are often limited and a large-scale systematic development phase is not always followed by a long-term, reliable maintenance and update phase, with a proper change management. Such circumstances often cause mismatch, incompleteness or inconsistency in the information provided through various assets, including waymarks and signposts on the field, digital and printed maps. Improper information provision leads to confusion, mistrust and may lead to dangerous situations for the visitor.

More and more trail-related platforms and applications are available, including crowdsorcing, for various audiences and stakeholders, such as trail information and navigation for visitors, and facility management for trail stakeholders.

The aim of this paper is to develop the basis of TrailSigner, a sound geospatial conceptual data model suite for managing waymarked trail networks and their signage assets, that can go beyond the capabilities of existing systems, and consistently provide various view interfaces for different actors as they need for their tasks, especially supporting composition of trips for visitors, and proper signage planning and management for providers over dynamically changing networks in a partially automatized, incremental way. This work can be treated as a proposal for future development of existing systems, and ultimately, as a potential basis of an envisioned, collaborative, integrated trail information system, that can support the effective, organic and consistent planning, management and maintenance of trails and their signage, and provide rich touristic content for the visitors in a flexible way.

The used methodology is to observe and analyze the application domain, consider typical actors and their general needs, look at typical real-life static and dynamic phenomena related to trail networks and their management, identify the challenges these impose on data modeling and management, considering these in the light of the data model of current systems, especially the conventions of OpenStreetMap, and develop an integrated data model suite where each (sub-)model constitutes a specific part or aspect and has a well-defined role and function. For supporting signage management, both the actually installed signs and the needed, planned or recommended signs (changes) should be modeled, with semantic relationships to the trail routes and network, and this way, existing inconsistencies can be revealed, and any effects of network changes on signage can be generated and with the necessary user (signage planner) input, turned into actions of implementation (for the facility management).

The paper is organized as follows: Sect. 2 gives a systematic overview about the application domain with references, sets up a common terminology for further use and reveals some general requirements for modeling. Existing information systems and related data modeling issues are presented in Sect. 3. Sect. 4, 5 and 6 propose a unified conceptual model suite composed of sub-models, that can be a basis of an integrated information system for trail network and signage design, management, maintenance and presentation; and shows its capabilities through solutions to particular real-life issues. Signpost logics will be constructed in Sect. 6 and the Appendix, supporting one of the main challenges addressed: incremental and consistent signage planning over a dynamically changing trail route network. Sect. 7 concludes and adds some future prospects.

2. The Application Domain: Trails and Networks

2.1. A Running Example with Modeling Considerations

In order to introduce and illustrate the concepts and phenomena being discussed thoroughout the paper, consider the example of a *trail network* shown on Fig. 1. The network consists of four designated *routes* which are marked with different colors (both on maps and on the terrain with waymarks) and denoted with small letters. A route has a continuous linear geometry defined over already existing ways: paths, walkways, paved and unpaved roads of various types.[1] Named *locations*, denoted by capital letters, are places or localities along the trails, which are identifiable in the landscape, usually displayed on maps and have significance for visitors as possible destinations or anchor points in their orientation and trip planning. Pictograms denote types of features (including facilities) based on relevant selected *points of interest (POI)* next to the trails. They may be related to named

[1] This is being named *simple route* in our modeling, while *complex routes* will be a collection of simple routes.

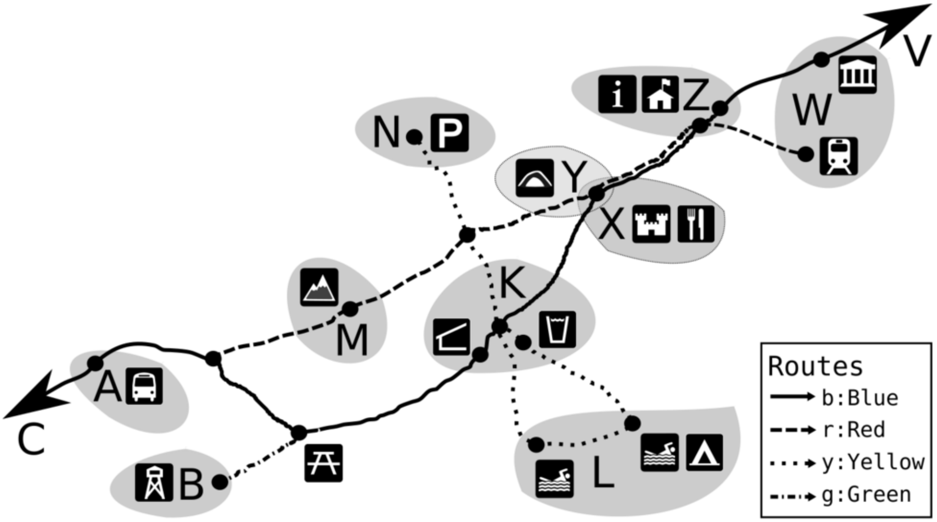

Figure 1. Hiking trail route network example

locations, but not necessarily. Black nodes denote *trail network nodes*, which are starting and ending points, junctions of routes, or places with similar or higher significance (aka. *pseudo-junctions*, selected points along the trails for anchoring locations, signpost positions and POI features). We call a part of a route between two neighboring nodes a *trail section*, so routes are composed of subsequent sections. In this example, we treat each route as bi-directional, and declare west-to-east as the primary direction of each.[2]

A high-level description of a route can be a *route location list*, i.e. a listing of selected subsequent locations passed-by; e.g. the blue route b on Fig. 1 can be described by the list $b : (C, A, K, X, Z, W, V)$. If another trail crosses or branches at an unnamed junction (such as the green route here), the name of the other trail can be used instead of the location: $b : (C, A, g, K, X, Z, W, V)$. This description is not a unique characterisation but rather provides a first simple overview for the users in guidebooks or other media. Any part of the route between locations or explicitly referred unnamed junctions is called a *route stage*. For longer (complex) routes, multiple levels of route location lists can be specified in a hierarchy.

In general, some relationships in trail-related data are based on pure geographical proximity (such as to collect all POIs along a route). Other relationships are, however, semantical, intentional, and up to the trail managers' decisions as part of touristic presentation or interpretation (to present the trail to the intended audience, e.g. a cultural route, educational trail, etc., or textual descriptions with selected media content such as photos). Explicit selection and edition adds high value of the presented information, and proper data modeling must provide a schema for doing it.

The network of Fig. 1 has the following characteristics and anomalies that occur in real-life situations, which must be considered by proper modeling (supposed the network itself is given as it is and changing or refining anything is not directly possible):

[2] By default, a route will have a primary direction in our model, and if it is marked bi-directionally, its reversed variant will a related route.

- Some routes are long-distance trails which extends towards locations outside the scope of the regional network (such as route *b* towards C, V).
- Names correspond to locations that can be clusters of nearby POIs and trail junctions (such as K or Z), or larger geographical entities such as towns (here: W) or lakesides (L).[3]
- A route may pass multiple times by the same location ($y : (N, r, K, L, L, K)$), different routes may lead to different points of the same location (W), and different routes may pass through the same location but at different POI features (K), or not even passing by the location's 'main' point where the facilities can be actually found (such as route *r* at Z). In the latter case, route *r* may still be described as passing through the location Z, but the signposting should ensure visitors can find the related facilities without accidentally passing-by.
- In some cases the POI feature(s) can extend the location name for unambigous identification and information (e.g. $W_{railway}, K_{shelter}$), or sub-locations should be defined and related hierarchically to the larger location entity (see later on Fig.3). This may become a particular issue when new routes are created, since there would be no ambiguity if only one of the routes *b* or *r* existed yet.
- Different location names can refer to the same place, and can be specific to one or more POIs found at the same (or nearby) place (see X and Y). Depending on the route thematics, one or another can be referred by specific routes (e.g. if the route *b* is more cultural-oriented, X with the castle will be referred in the location list of the route, while route *r* may be more nature-oriented, and the cave named Y will be used instead).
- Not all trail junctions or POIs are named, not even some starting or ending points of certain routes if they merge with other routes (see the example: $r : (b, M, y, Y, Z, W)$, $g : (b, B)$).
- Some routes may be one-directional, others are marked in both directions (we assume all routes in the example are marked in both directions, and use the notation b' for reversing a route *b*, such as $b' : (V, W, Z, X, K, g, A, C)$).
- Two or more routes may share the same section (such as *b* and *r* between X/Y and Z).
- A longer route may be technically split into shorter stages (*b* may be split at X, for example: $b_1 : (C, A, g, K, X), b_2 : (X, W, V)$). These shorter stages as simple routes should explicitly be related to form a complex route. The reversed version of a route should be split at the same points as its original. The split is not necessarily visible to the user of the complex route, or only as an additional information.
- The network or signage concept may define relationships among the routes (such as *g* is a side-trail of *b*, or *r* is a variant of *b*).
- Routes are mostly linear but there are circular routes or routes with loops (here: *y*). Circular routes which end at their starting point can be properly handled, but identifying directions at loop junctions where a trail diverts from itself can be

[3] Modeling of locations related to POIs, trail nodes and routes is non-trivial. They can be defined by the support of geospatial proximities or clustering, but due to the many types of locations and possible special cases related to route network situations, it is a problem of its own and can be taken as a future issue. What this paper aims is, to show some of the typical, non-trivial issues and to create a flexible data model that can handle these. But how to identify and define the named locations themselves, is considered as manual work by now and out of the scope of this paper.

problematic (e.g. y at K). A general recommendation is to split the route into two stages when modeled, so a (one-way) direction can be uniquely identified as the route continuation at each point or junction: $y_1 : (N, r, K, L_{camp}), y_2 : (L_{camp,swim}, L_{swim}, K)$. A complex route is then formed by linking these simple routes together.

2.2. Trails and Signage Systems

Structuring, presentation and on-field marking of a trail network is determined by its signage system. Countries and regions have developed their own systems during the last century [4], based on different network structuring principles. A general data model must be able to cope with such differences and give the opportunity to effectively and flexibly utilize the information inherently contained by the network and its signage system. [11] defines two following ground types, and lists the advantages and drawbacks of each.

Junction-based system (Knotenleitsystem) No routes are defined explicitly by the trail signage. The network graph (see Fig. 7 later as an example) is taken as it is, and each trail section is classically waymarked with a single, unified symbol. Multiple types of waymarks, if used, can denote properties of trail sections (e.g. difficulty by color or allowed transport modality by pictograms). Each trail network node is identifiable with its unique name. Directional information at each node must be provided towards the next nodes, and to some further, important destinations. It is up to the visitor to combine his/her own tour itinerary by route planning along the graph, and recommended trips can be given in form of trail node listing.

Route-based system (Routenleitsystem) The trail system is a collection of independent routes marked with their unique symbols (colors, logos or numbers), and these routes are self-contained, recommended trip itineraries, with their own names, thematics, and without specific attention to their connections with each other. Route information is usually provided using textual and visual media content (e.g. guidebooks), route-specific waymarks ensure the route can be followed at every junction and all along its way. Signposts are linked to routes and can provide auxilliary information, such as the distances of places directly located along the route.

The above ground types are rarely used in their pure form in practice. Most trail signage systems and network concepts are based on a combined approach [4]. A junction-based system can be enhanced with specific routes or a route-based system can be enhanced with side trails, route variants or connecting trails, using a standard set of waymark symbols or colors for local route identification. Routes can be properly aligned to each other to form a connected network, including routes which do not form a self-contained trip itinerary by themselves, but can be reached by other routes. Our running example is of this type. A specific mixture of route- and junction-based system is when the routes are numbered, and these numbers appear with standard waymark symbols at junctions (or even with destination signposting), and a standard, simplified waymark without route numbers is used along the trail sections between junctions.

Routes are marked with the symbol or color determined by the applied network concept and signage system. Some systems feature local distinguishability of routes by not allowing two routes of the same color or symbol cross at one place, while other systems use colors to determine the route type, difficulty or role in the network instead (in our running example, routes are distinguishable by their colors, however, the yellow route has a loop).

2.3. The Trail Signage Stack

The trail network is based on existing ways (roads or paths) accessible to the public. It is assumed that the routes are marked on the terrain with simple waymarks such as painted blazes with their respective colors (in case of a classical junction-based system, a route equals to a trail seciton), so they can be followed by visitors relying on the waymarks [3] (in both directions, unless explicitly stated otherwise). At trail nodes, properly designed signposting can be built on top of the basic waymarking signage, which inform visitors about directions and distances to named destinations along the routes, or reachable by their combinations. To construct a unified data model for signage, being able to describe the different types of above mentioned signage systems, we define the *signage stack* first, where signage facilities are layered, built on top of each other and must form a coherent and consistent system, all together with the digital information provided through various channels:

- *Passable* paths must exist which can be technically and legally designated as a waymarked trail, and made part of the network. It must be maintained.
- The path must be continuously *followable* by subsequent *waymarks* placed along it (usually painted *trail blazes*, or more generally, *trail markers* which may contain more information), especially at open spaces and junctions with other (not necessarily marked) ways. These marks are usually painted blazes or some simple signs on trees or objects next to the trail, and marking is considered more likely as a linear feature related to way sections ('the section is waymarked') rather than managing each single waymark as a point feature (which is also possible if needed).
- The route (and/or trail type) should be *identifiable* by identification elements in the signage, at least at certain points (junctions with other trails, start/end points). This can be a number, a logo, either painted, mounted as part of another sign or separately. We call it a *route sign*. The *trail marker* may already have this role.
- The route network should be *navigable*: Places along the route are recognizable using place marks (*location signs*, usually displaying the altitude of the place above sea level) wherever it is meaningful. Route directions with distances or walking times should be pointed at at junctions and trailheads, in form of *route destination signs*, which can contain pictograms of relevant POI features found at the displayed destination. Directional *sign boards* (such as Fig. 2) mounted on *guideposts* display one or more route destination signs, usually in the order of distance. They display the route or waymark information (color, name, number and/or logo), or a combination of routes to reach the displayed destination(s).[4]
- The network should be *overviewable* using maps, displayed on *info boards* at trailheads or touristic centres, and takeaway-maps in paper form or by digital media.

Distance or walking time, as well as difficulty (if displayed) must be computed in an unified way given the track of the path to the signposted destination. Walking time can mostly be determined by a formula based on an aggregation of lengths and altitude

[4] These are also called *signposts*, but we use this term in a more general meaning: either as *signposting a route* (adding location and route destination signs) or as any single item of the route signposting.

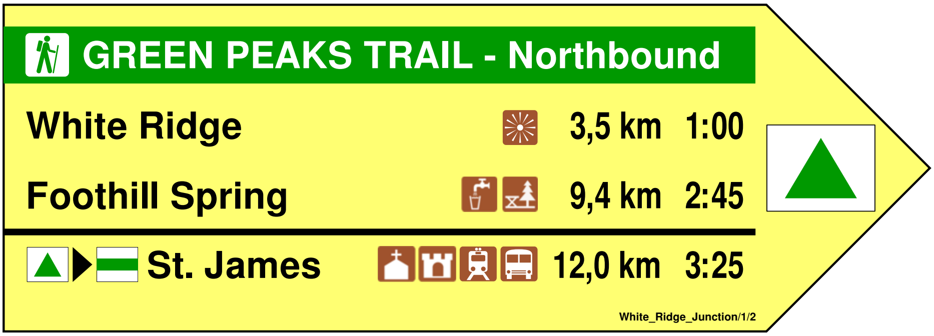

Figure 2. A sample sign board (signpost) with a route sign at its top and three route destination signs with location features, distances and walking times. The common trail mark of the route is conceptually part of the route destination signs, but by the actual design of the sign board, it is displayed only once (in the arrow part) - except for the last, combined destination sign indicating an indirect route connection.

profiles of involved trail sections (such as in [15]) with a reasonable rounding principle. However, some sections may need manual update due to specific path conditions. Difficulty may be aggregated depending on what exactly it means (c.f. [33]): the aggregated *technical difficulty* is the maximum difficulty level of the involved sections, while the necessary *effort* must depend on a sum of sections. These parameters should be computed the same way for each information asset (including signposts, maps, digital touristic information provided) and therefore, should be managed centrally on the route network level and queried for any presentation interface.

Additional signage, such as interpretative, educative, warning and regulatory signs can also be considered, but as they are not contributing to the orientation and navigation tasks, they are out of the scope of this paper.

2.4. Actors and Dynamics

Different actors have their own view of the same reality, and the key of an integrated data model is to unify these properly.

A *visitor* who is planning the trip needs only an overview, with appealing media content, and aggregated information (e.g. sum of kms or altitude gain, max-min altitude, max of technical difficulty, max of steepness). Simplified route representations and descriptions are sufficient, with the ability to take a route or combine their sections into an itinerary. On the field, the visitor may need detailed topographical and navigational information about trail sections and their junctions, and the exact places of POI features at specific locations.

A *trail crew* needs to implement changes or reparations and has to know on the field exactly where to do what, based on a detailed network plan, and data of existing and planned facilities. A *trail network designer* needs to know the existing segments and suggest extensions or changes, new routes or re-routing, and will give input to *touristic content editor*s to extend or adapt the presentation of trails for visitors. The *signage system designer* needs to know the rotes and his/her task is to propose what content should be displayed at which places conceptually (ideally, in coherence with the touristic content being provided, e.g. when defining route stages), while the *signage implementor* needs to have all this information compiled, enhanced with precise data, and has the task to design

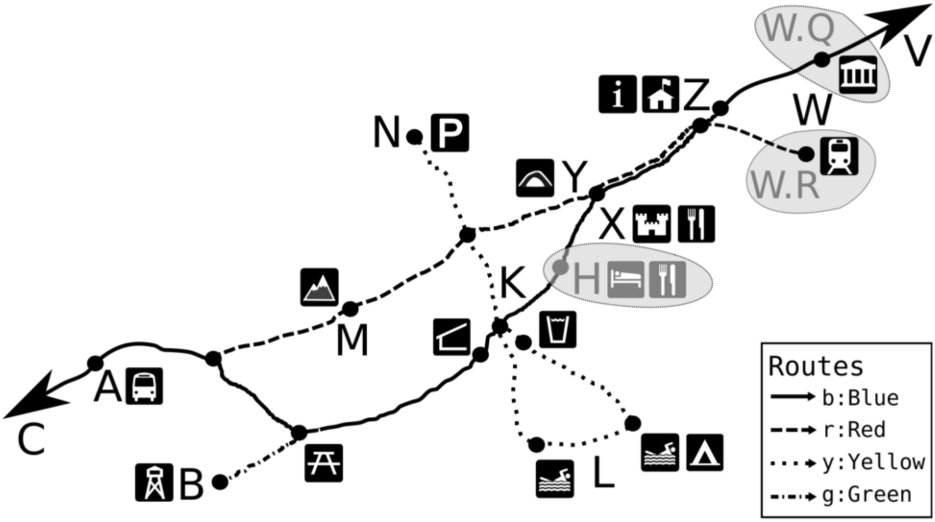

Figure 3. Hiking trail route network example with refined and added locations

and produce each signpost and pass to the trail crew for installation. Certainly, different role settings are possible, but these tasks are to be suported.

Trail routes or networks and their management patterns are often developed incrementally and organically, and change dynamically over time. They usually involve bottom-up civil and volunteer engagement and enthusiasm, and at the same time, are dependant on governmental authorities, must obey certain policies and regulations, and must take into account interests of private and public land owners as well. Development of trails is often determined by interests and activities of a multitude of stakeholders and contributors on multiple levels, and is usually incremental and organic, providing the user a 'local-as-design/global-as-view' ambiente. Sometimes there is no unified organization behind a trail or network, but only contributors. In other cases, network design and trail management is driven by strong public or private interests, and is more like a top-down process, involving less number of stakeholders and have a solid organizational structure. In such cases, it becomes similar to a 'global-as-design/local-as-view' system for visitors.

Trail networks are more versatile than public road systems. For example, new attractions or facilities are created along the routes, which may have substantial effect on the network layout and signage. Courses of paths may change for various reasons, sections or routes may be added to the network, temporarily or permanently closed, and routes may need to be re-routed. Signage is often developed incrementally, but the system should support its consistency. These processes include user decision and input, but some parts can be automatized. Example scenarios inculde:

- A new hotel is being built along one of the trails. It needs to be incorporated into the signage system and touristic contents provided (such as location H on Fig.3).
- A village can support installation of exactly 3 guideposts. Where to put them and what to indicate on them?
- A trail section is closed or modified and a route must be re-routed. Which sigposts have to be updated, moved or newly installed?
- One of the routes need full signposting at each junction. Plan it, implement it.

- Location names need to be refined in relation to route info and signage to aviod confusion (such as W on Fig. 3 where two routes arrive at different parts of the town W).

2.5. Signage Planning and Management

One of the major aims is to support the systematic and consistent design and maintenance of trail signage. Signposting is not always complete and up-to-date, and may be at different development stages for each route or network part. Revealing inconsistencies and generating actions for necessities and recommendations is an important task that can be supported, especially for complex networks.

Actual signposts (sign boards mounted on guideposts) can be composed of multiple route destination signs (c.f. Fig. 2), but for simplicity, we use the term *signpost* in this section as an abbreviation for route destination signs, and do not consider how they are actually presented on sign boards, but only what locations need to be signposted at which places via which routes.

Signposts are not independent of each other, they must be treated systematically, as a set of them along the routes. If a signpost is already placed, it will imply other signs we must place along the route(s), to ensure consistency and the continuous, unambigous followability of the signposted routes and directions. The European Ramblers' Association (ERA) has created common guidelines for the application of signage elements [3]. For signposts, it contains the following rules. These may be extended by other guidelines, specific to the locally applied signage system.

(ERA-8) 'Directional panels or fingerposts indicating the route of the marked path should be placed at the starting/ending points of the route, at junctions and at other necessary places along the route.'
(ERA-9) 'Fingerposts and directional panels should indicate one or more destination along the marked route and the distance in km (miles) or hours. If a destination is shown on a fingerpost or panel, it must be shown on all other subsequent fingerposts or panels until the destination itself is reached.'

Another, reasonable guideline can be that the locations need to be identifiable as they are reached: if a destination is signposted anywhere to be reachable along a route, a location sign of some form must be installed at the point where it is (meant to be) reached. It can be an explicit planning decision because named locations often do not correspond to a single trail junction or POI (c.f. locations K, L, Z, W on Fig. 1 and their connections to different routes). If main facilities are not trivial to find at a location, local signposts must be added.

Fig. 4 shows a signposting example for Fig. 1, which is neither complete for any route or the network, but is logically closed and consistent. It shows the implications of placing specific signposts at point A, towards Z along b and towards M along b followed by r: The latter one implies an extra signpost towards M to be placed at the next junction along r, and the existence of this one together with the previous one at A towards Z implies a sign towards Z via b (if rule (ERA-9) is taken strictly). Furthermore, location signs are implied as follow-up signs by signposts direcly pointing towards any destination to confirm the destinations are reached. We make the following observations for consideration:

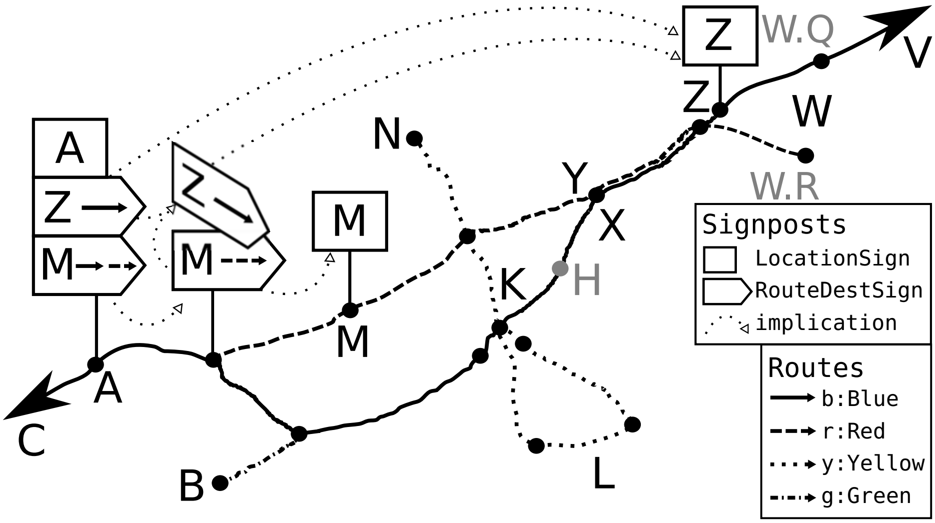

Figure 4. Signposting example

- Signposts do not necessarily mean the shortest possible way towards the displayed destination, and there might be different options, so the data model needs to handle it by explicitly or implicitly adding the whole course of the pathway meant to reach the displayed destination from the signpost (e.g. route *r* takes to Z as well via M, and it can be signposted additionally, as an alternative direction).
- If we placed another signpost at place K showing X is reachable via route *b* from there, it may cause confusion unless Z is repeatedly signposted there.
- In a route-based signage system, signposts primarily designed for the routes and display destinations directly reachable along routes (without changing to another route). There might be important destinations that fall outside of these, so they can be pre-announced by a signpost with a notice that the visitor must change for another route at some later point. With this flexibility, some of the strengths of a junction-based signage system can be adapted into a route-based system. This is the case of the mountain M along the red route, for example. It causes interdependence between the signposting of multiple routes.
- Regarding to what destination should appear on a signpost at location B (since route *g* does not end at a named or significant location), it is reasonable to choose a further location along route *b* (or its reversed b'). Existing or already planned signposts of routes b, b' at the end node of *g* (e.g. A, X, Z or K) impose a suggestion or possible choice for a planned signposting at location B for route *g*.

A common practice of systematic signpost planning is the *route destination table (or route signposting table)* method [14], where subsequent trail network nodes along the route are listed in a column as potential signpost places, and the destinations to be signposted are written in the rows towards each direction (forward and backward). The number of signposed destinations at each place in each route direction is limited by the used network signage system (usually 2 or 3, and they are actually displayed on a single signpost per route, in the order of reachability along the route). Fig. 5 shows an example for route *b* as an extension of the signposting at Fig. 4. Trail network nodes along the

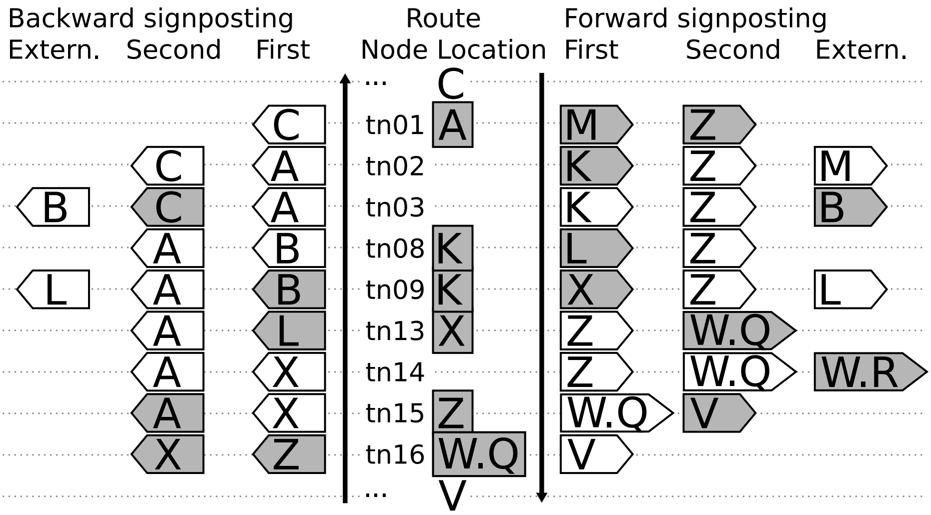

Figure 5. A full route signposting table example of route *b* (Blue) in both directions. Location signs are planned based on user-chosen route stages. Grey signposts are chosen by the user (signpost system designer). Signposts with white backrounds are implied by the already chosen grey ones. Note that some signposted destinations are outside of the route *b* and therefore, this route table is interrelated with signpost plans of other routes (through the *Extern.* columns). Some *Extern.* signs are duplicates as they appear in both directions.

route are listed in the middle column, and in each row, the right side contains planned signposts in forward direction (top-down), and the left side in reversed direction (bottom-up, it is actually the signpost table of route b'). At most two destination signs are planned for each possible signposting point per route and direction. Implications are shown with different backgrounds, but the exact places of location signs must be explicitly decided since there may be more nodes fall into the same location (e.g. at K). After all route signposting table is completed, the planned signposts are collected for each place (node) from all routes, and put onto a special form for each location (*Standortformular*, see [14]) where the precise content and arrangement of signposts and their planned installation is designed, based on the layout of the junction.

Note that some destinations fall outside of the actual route b, and they imply external signposts (pointing towards another route). The signposting table of a route has implications on the signposting table of another route. If the network is dense, and there are many routes, it becomes difficult to ensure consistency when designing the signposting system by hand, and might be more difficult to handle if changes occur in the network. So far, it is not known if any trail facility management software supports this functionality.

3. Trail Information Systems and Modeling

3.1. Models and Features of Current Systems

Many applications, websites and map tools provide information about trails and their networks, completely different trail systems exist around the world, related to various fields and levels of profession (tourism, transport, conservation, sport, forestry, etc), each having their own approaches. Technical documentation is in many cases not available, non-public, insufficient or not internationalized. This heterogenity makes it difficult to give a well-founded overview of the situation, but some visible trends can be outlined.

Apps or websites for visitors usually have two basic options for trip selection or planning: choosing a recommended, pre-compiled trip itinerary from a catalogue and take it as a whole trip or outing, or perform a totally flexible routing along a network graph and generate a trip this way.

In the latter case, most of the additional information provided for the pre-defined trips (tours) cannot be utilized (except for POI information if generated based on proximity) and the user is usually facing a low-level (too detailed) representation of the planned trip. Multi-level modeling has been existing already for a long time for road networks, such as [34,16] but haven't been adapted for walking/hiking trails. Even if it might not be needed for effective way finding, it seems to be reasonable for suitable information presentation and to establish a unified and semantically enhanced signage management.

Most touristic apps presenting trails do not handle networks, and the pre-compiled tour itineraries don't have any flexibility. An exception is the GuideAtHand app [21] which allows content providers to plan some optional detours or variations into its walks, and the system ask the user on the field which variant to take further. It is still far from trip plannig and adaptive walking-through by allowing a dynamic combination of pre-compiled *partial trips* forming a network, which might be a better option for users searching for some variation but feeling somewhat 'lost' with a totally flexible network routing in a complex network, and would utilize more linear-based, pre-compiled touristic information such as a route description and people's feedbacks on trail sections.

[15] mentions the lack of standard data description models for pedestrians and so, for hiking routes and their networks. It identifies various information requirements and introduces the *hikingtrack* concept as an extension in schema.org modeling. Not even difficulty rating is standardized: it defines 3 different types (effort, technical, pshycological) which should co-exist. Hiking time computation are based on various methods (such as [33] and others). Some of these are proprietary.

Trail and network planning and facility management can be supported by multiple GIS and cartographical tools, including general and custom solutions. Current trail managers most likely use custom solutions (if any), because there has been only recent widespread publicity on trail facility management systems. [5] presents Komoot, NatursportPlaner, OutdoorActive and RealityMaps as 'digital trail mangement software'. Natursportplaner [6] is generally used for planning, while the platform of OutdoorActive integrates trail information provision for visitors with facility management functionalities [19,32]. The publicly available data model of OutdoorActive [24] contains various point features (POI), a *tour* as a linear feature (corresponds to a trip in our terminology) and a *region* type (all originated from a common supertype *object of interest*) with *categories* applicable to them.

Another, newly developed platform is OuterSpatial [7,12], which provides information for visitors as well as for trail managers and partners, and an interface for community contribution with user feedbacks directly channeled into facility management. Its data model is based on *areas*, various types of *locations/POIs*, *trails* and *outings*, with map and media content in PDF.

Such platforms usually support the management of field assets, including signposts, based on their location, status, type, and can store the printable image of the signposts for (re)production. Some tools may support the computation of distance or walking time based on manual selection of displayed destinations on signposts. Since no public documentation is available for most systems, it is difficult to assess, but features are not known which would semantically link the signposts to each other and support their systematic design and change management in the way has been presented here. [9] reports a custom design process for a large-scale, long-distance trail signage design, but it does not consider the operational management phase, and no official feedbacks or follow-up information is available.

Besides commercial products, custom solutions and platforms provided by trail or tourism organizations, there is a wide variety of community-based map tools and applications, primarily based on OpenStreetMap ([1, Comparison of Android applications] shows more than a hundred of Android apps - not only related to trails, while [1, Hiking Maps] compares 9 different map services/platforms especially for hiking). Wikiloc [2] is also extensive and popular (featuring almost 15 million of trails at the time of writing this paper), but its functionality is limited to precompiled, recommended trips as trails.

A multitude of related work exists on surveying hiking preferences (such as [13]). In the recent years, analyzing user behaviour, personalized recommendations became widespread in general, based on tracking or social network data for mobility patterns (such as [10]) and some works especially for hiking and walking have recently been appeared (such as [20,26]). Personalized trip recommendations for flexible rambling is a promising future direction, and should be supported by proper modeling.

3.2. OpenStreetMap and its Trails Data Model

OpenStreetMap is likely to be the most widely used community mapping platform with a multitude of related services, websites and apps. Its data model is based on flexible tagging and conventions rather than strict structures or standards. Conventional tagging schemata of features related to hiking is documented in its Wiki [1, Hiking] and a substantal amount of hiking routes and facilities are already mapped.

It uses a route-based approach to model trail networks [1, Walking Routes], with the option to define short routes with similar signage, so a junction-based or mixed system can be simulated. There is a hierarchy of routes defining *international, national, regional* and *local* walking routes, and *hyperrelations* allow to create a hierarchic long-distance route with daily stages. There are no unified conventions for further semantic connections among routes.

Trail routes are so called *relations* composed from an arbitrary choice of elementary way segments. Direction of the route is not specified, and route geometries are not necessarily continuous linear features. Flexible trip planning is provided with general routing, which does not take into account the waymarked routes (services are currently limited for planning along waymarked trails). Trail and signage-related tagging varies by countries or regions, as the signage systems differ. Way segments do not generally match trail network sections as we define it (and our definition includes user decisions as well, by the pseudo-junctions - however we did not say anything about their actual implementation), they are usually shorter and not necessarily cut at trail junctions, but the route planner can work on partial way segments as well.

Modeling of routes on OpenStreetmap is lacking the explicit assignment of locations [1, Walking Routes], as providing intentional touristic content is out of the scope of that system. POIs or other features can be implicitly collected to a route based on geographic proxmity. Route descriptions with starting and ending points or important intermediary locations often appear in the textual description or name of the route. However, they have no explicit semantic relationship to the geographic features they refer to.

For our purposes, the basis of the route network can be extracted from OpenStreetMap, together with specified types of POI and other geographic features in proximity of waymarked routes, as well as manually selected non-marked way connections. [27] shows a method for road networks guided by a conceptual schema. Post-processing may include checking route geometries, cutting dangling geometries (each route should have a single continuous line course), making route directions explicit (generate reversed routes as necessary), and assigning named locations.

For signposting, there exists a tagging scheme for modeling route destination signs [1, Destination Sign], to aid navigation software when displaying the actual directions to follow (e.g. instead of just ’turn right’, ’turn right in the direction signposted towards X’). It is a relation containing an exact location, a junction or way reference point, the direction where the arrow points (in form of a way starting at the junction), the name of the destination, distance and/or time in hours written on the signpost, and a route reference. It is a purely formal representation of what the signpost is, without semantic relationships to the shown destination or the actual route or pathway meant to reach it. It may contain more than one sign items (route destination signs in our terminology) at once.

This tagging scheme may be enhanced to form an extended data model, where the signs of destination directions displayed on signposts contain actual references to the destinations they show, and to the pathway along the network of routes to reach it. This way, the systematic design and also the maintenance of signage can be supported, so that the displayed distances can be computed automatically, the validity of actually existing signposts can be checked, and it can give an input to the community mappers to map missing parts of routes referenced by signposts.

3.3. Related Work on Modeling

Considering what has been presented so far, the following works (tools, concepts and methods) seem to be especially relevant and useful for our modeling case. Their particular application should be elaborated in more details in the future.

The aimed integrated information system must provide multiple facets to various actors, who need to see the same reality from different viewpoints (actually in different models), also depending on their task and current situation (e.g. whether a visitor is at home and on the field). It assumes a complex database setting whose way of operation may also change during the phases of network or organizational development, but the operations should be well-defined. Therefore, *model suites* presented in [28] seem to be a proper means of specification in the long term as they can provide a framework for multi-level data settings, and coherent, but flexible exchange and collaboration patterns.

For conceptual modeling of data structuring, the higher-order entity relationship notation (HERM) is especially convenient [30,29] as it is well-founded and can compactly express complex settings. Traditional ER or UML class diagram could be also used.

Specification of interaction and functionality should be based on flexibly combinable, but self-contained (mini-)workflows where user interaction can be combined with system automatisms. For that, the concept of *mini-stories* for generic workflow presented in [31] could be a suitable framework. The interaction space can be specified in a flexible way, but specific stories can be featured and translated to workflows [23].

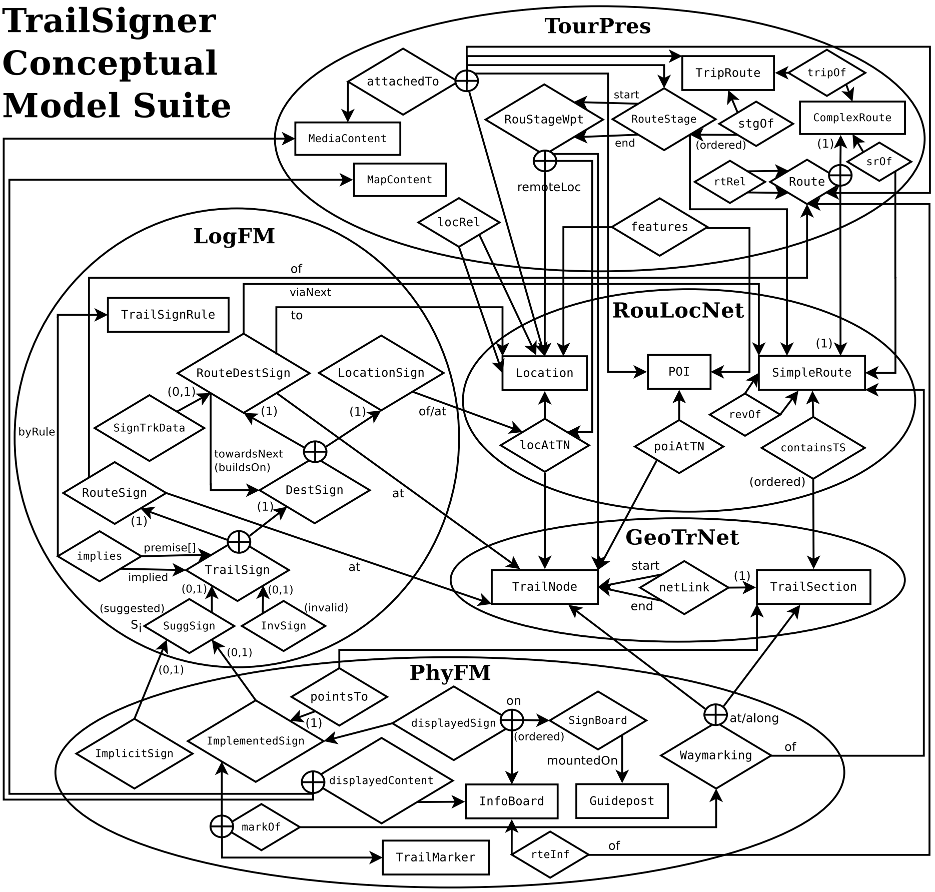

Figure 6. The TRAILSIGNER HERM model suite of trail systems. Cardinalities - wherever constrained - are shown in the participation notation. Attributes and some derived relationships are not shown, for simplicity. Some of them are subject to model variations as well.

Modeling data constraints and derived attributes or relationships presupposes the use of *active database* concepts [25]. Implementation of signage dynamics and implication can be either based on triggers and database constraints in a classical relational RDBMS architecture, or this part of the system can be oragnized separately, using a logic-based data platform, with the relevant considerations kept in mind (such as in [8]). From the user perspective, systematic signage design should involve some way of graphical and spreadsheet reasoning, as presented in [18,17] for the schema level, but here on the instance level (c.f. the route signposting table method).

4. Overview of the Proposed Model Suite

We propose the skeleton of TRAILSIGNER, a unified conceptual model on Fig. 6, in higher-order ER [30,29]. It can be defined as a model suite [28], in which consistency criteria and collaboration styles may be flexibly declared among its (sub-)models. Its (sub-) models and their functions are described briefly.

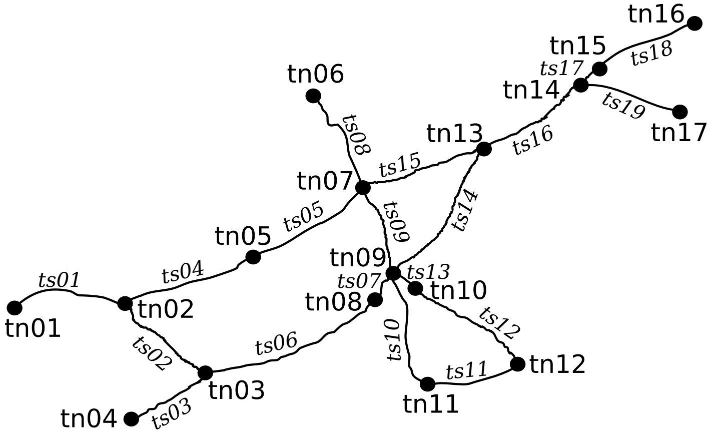

Figure 7. Hiking trail network graph

Implementation details are not subject of this paper and are left open for further refinement of the model, but a general versioning facility is in any case needed for tracking historical states and to explicitly publishing network changes. It is a conceptual model and different implementation schemata can be constructed to it. Some of the model constructs have more possible translations into logical database schemata. Some of the relationship types can be (at least partially) derived, based on geolocations and other relationships. The actual implementation may depend on the workflow or policy of the organization(s) as well. However, to assess the feasibility of it, a detailed version of this schema is being published by the author with one possible implementation as an on-going project under [22]. Nevertheless, the following sections briefly discuss how the model in its current level can cope with the challenges described in the previous sections.

5. Modeling Trail Route Network and Touristic Content

Each of the following three (sub-)models are generally a different view of the same reality for different purposes (with specific enhancements as their own data), and therefore, must be kept consistent. A refined definition of the model suite may further specify their relationships and implementation.

5.1. The Trail Network Graph (GeoTrNet)

The trail network model forms a classical geospatial routable graph structure. Its function is to represent the exact trail courses as they are on the terrain, mainly for trail work and on-site navigation (including the altitude dimension). Fig. 7 depicts a possible underlying local graph of routes of Fig. 1 with technical identifiers of trail nodes and trail sections connecting them. Note that this graph can basically be generated from the geometries of routes and their intersections, by splitting route geometries and adding a node to each route junction and endpoint, but must be refined by adding extra nodes at pseudo-junctions (facility locations). The graph is generally bi-directional, except for explicitly defined one-way trail sections. Technical difficulty, computed (and in special cases refined) walking time can be stored or computed on-the fly for each section, together with distance and elevation gain/loss derived from geometries. Status of a trail section may indicate a planned or a closed section and influence its permitted operations and visitibility to different actors. Changes of the network, new related locations or features, including sigposts may imply sections to be formally split. For such cases, update mechanisms must be defined for each of the relationships of trail sections if the split is implemented as the cut of the line (other possibilities also exist by linear referencing, but here we speak about a conceptual model and it is the question of the implementation schema). Minor changes of trail sections, which does not change the network structure might have effect on signposts, or stay below the rounding tolerancy of displayed data.

5.2. *Routes, Locations and POI (RouLocNet)*

Trail routes are represented with their full geometries (mutual consistency must be ensured with GeoTrNet, either by defining a master-slave data exchange pattern in either direction, restricting the modification of geometries in one of the models, or allowing any modifications supported by a two-way update mechanism). Its function is for network designers, managers and visitors, to represent the routes and how the locations and facilities are connected by them. Any trail section has to be covered by a route, as waymarked routes define the network (trail sections without an assigned route are considered to be planned or closed). Signposts and touristic content can be assigned to routes and not to trail sections directly. In a junction-based system, each trail section defines a route, or routes can be arbitrarily chosen to cover the (junction-based part of the) network. If there is no actual waymarked route along a section, and the connection is important, a pseudo-route can be generated, making the section as part of the trail network on the route level as well. Routes may have no or partial actual geometries, either in planning phase, or when a long-distance route spans beyond the regional scope of the system, in the latter case, derived data may be imported from elsewhere for the missing sections and locations, to enable seamless signposting and (long-)distance computation along routes. Routes have simple continuous geometries and do not intersect themselves, nor repeat or lead back on the same section. More complex routes must be split into simple routes and connected as a complex route on a higher level, in the *TourPres* model.

If a new route is added or its course is changed, the updated network must be generated with possibly new nodes (split of existing sections, their derived data recomputed) and it may imply changes to the signposting and touristic contents (see below). Route status and versioning is necessary for an integrated system which actually contains planned trails and historical data.

Routes are taken as one-directional. Bidirectional routes must be duplicated, inverted and related to their primary variant (*revOf*). This controlled redundancy has the advantage of simpler handling of waymarking, signposting assets, and touristic information.

Relations in this model are completely generated by actual geometrical relationships, but trail-related, relevant named locations and POIs must be defined or selected explcitily.

A location may be defined by a single POI or a cluster of nearby POIs (such as M or K on Fig. 1, respectively), or a geographic object with an extent, such as a village (W), lake or lakeside (L). A location can also represent a geographic place name without any specific POI feature type, or not originating from the POI(s) actually displayed as trail features (e.g. the bus stop at A may have got its name from a farm there called A, which itself is not relevant as POI for the trails, but gives the location name). Modifications of locations impose an update on the network and related entities.

5.3. Touristic Content and Presentation (TourPres)

Its function is to give a higher-level overview of the trails, their features and touristic interpretation to the visitors, making the trail network overseeable and browseable, and provide a platform for media content input (images, descriptions attached to locations, POIs, routes, or parts of them). Trip routes can be composed from parts of network routes (either simple or complex - the latter may have variants, or may be a multi-day trail with day sections defined as simple routes) and offered to the visitor, and the visitors can themselves assemble their own itineraries. Routes may be divided into longer *stages*, based on explicitly defined route location lists (possibly multiple sections, either defined by an unnamed node, a node of a location or a remote location not part of the local(ly stored) network). Stages are the basis of linear touristic content management, and visitors can select or combine their *trip routes* from them. Location priorities along routes can be added to the TourPres model by defining hierarchically nested route stages (location lists), for example. This gives a flexibility to assign content in route views of different details.

This part can be integrated into a content management system. Actual geometrical relationships can support the relationships in this model (by constraints and suggestions), but all of them are actually determined by the content editor. At this level, structural or semantic relationships can be defined between locations (e.g. hierarchy), locations and POIs (selected relevant features of places), and routes (e.g. reversed, variant, continuation of another route). If a new feature is added to the system and assigned to routes, a new trail node may be necessarily generated.

This model part acts as the space of personalized user content as well (future issue).

6. Models for Facility Management and Signpost Logics

6.1. Physical Facility Management (PhyFM)

The Physical FM (signpost and waymarking) model represents the inventory of the actual trail signage assets as they are situated on the field, and those signage elements that have been decided to be implemented and are under production and installation. Waymarking of a route is assigned to trail nodes and sections, while location, route and destination signs can be nested into info boards or sign boards mounted on guideposts (c.f. the Signage stack in Sect. 2). Status of the signs can be stored here (either valid or invalid; existing, lost or planned) together with facility management actions (not indicated for simplicity). Signs may be *implemented* (actually existing) or *implicit* (expressed by other signs, and therefore, not explicitly needed). Besides the actual graphical and textual content of each signage element, semantical references are also kept (e.g. not just the displayed name of a location, but the foreign key of the referenced location itself as well), so that the validity of signs can be checked against the actual (potentially changing) trail network data.

6.2. Logical Facility Management (LogFM)

The Logical FM (signpost) model is the most novel part of the proposed model suite. It has the function of supporting incremental and consistent signpost planning and change management. It is a platform for conceptual set-based signpost planning and a mediator for change management between trail routes and their effect on signposts in the physical model. An instance of it is a partly generated, partly intentionally selected set of ideal sign elements in form of logical facts with different priority levels, which are always consistent with each other and with the actual version of the trail route network. A logical location sign at a node of a named location indicates a need for placing a sign indicating the location name there. A logical route sign refers to the need of displaying the route identification and continuation at a node. A logical route destination sign is an expression that the direction of a specific destination should be signposted at a node via a route or route combination. The exact meaning of such a logical sign is modeled in a linked-list-like fashion, always pointing towards another, follow-up sign: if the signposted destination is meant to be reached by following a single route, then the follow-up sign is a location sign at the destination. If the visitor has to change routes at a later point, then the follow-up sign will be a logical route destination sign at the change point towards the destination (see the chain of signs towards M on Fig. 4). Such signs are composed by structural recursion, always having a location sign at the end (finiteness, without circles). This construction is modeled by a cluster type DestSign, which is a generalization of location and route destination signs. The chain of follow-up signs is part of the identification of route destination signs, thus forming a complex natural key domain. The sign track data can be computed from the trail network and route data using aggregation (e.g. total distance, computed walking time, max difficulty, trailmark(s) to show).

The above composition mechanism can be used to create possible logical signs that can be implemented in the physical FM model. Since we do not store any data such as route or location names, waymarks, distances etc. explicitly in the core sign data types of the logical FM model, only references to the actual network objects, an actual view of the logical signs is always consistent with the network (some logical signs may not be valid after a network change, though, as they will refer to obsolete versions of trail routes, locations, etc). Attached *SignTrkData* is always (re-)generated or validated automatically.

The signage planner user may take decisions on the logical level regarding to which signs should be implemented. These must be consistent, e.g. if a sign is to be implemented, then a follow-up sign is required too. Similar guidelines for signage application (for recommendations and obligations) can be formulated as *TrailSignRules*. Each time a logical sign is added, removed or changed, the set of signs will be updated as the logical closure generated by these rules. This way, consistency and signage guideline compliance can easily be checked and the needed signs or updates can be selected for implementation. Different levels of implementation priority (necessity, primary or secondary recommendations) can be defined by the rules. Implication is tracked for logical signs to distinguish between explicitly added and implied ones. The implied ones can always be re-generated.

A formal definition of signpost logics is presented in the Appendix.It defines a series of predicates S_i that indicate the necessity levels of logical trail signs (it can be implemented either as attributes or database relations in the LogFM model). In a planning process, the signpost planner user decides on initial (existing or initially planned) signs, and sets them as S_0, and optionally adds lower priority signs to S_1, S_2, etc. Then an iterative logical implication step is made using the rules to generate further necessary signs into S_0 and recommended signs with different priorities $S_i, i > 0$. The signpost planner then makes one or more choices and moves some signs from $S_i, i > 0$ to S_0 and a next implication is being performed for further necessary and recommended signs. Although there is no direct formal conceptual relationship between LogFM and TourPres, the signpost planner should consider harmonizing the signposting with the touristic content provided (route stages), and the system can give respective recommendations.

If the signs are settled down in S_0, they can be put into planned stage, which means the sign track data is computed and it becomes ready for implementation (graphical design, production and installation). This is performed in the PhyFM database. The signs marked for implementation are not always identical to the actual signs placed on the field: some physical signs can be obsolete, and not all of the logical signs are actually implemented as physical signs (*ImplicitSigns*, whose information is expressed in other ways). The actual situation may differ from the ideal and the difference induce necessary actions, which can be represented by signpost statuses in PhyFM. To handle this properly, versioning of the trail data is needed and the historical data kept in the database.

If the network changes, logical signs are checked first as some of them may become invalid after a structural change in the network. They are backed up (marked as obsolete and become inactive for the signpost logics) and the status of their implemented signs is changed in PhyFM as deprecated (and a maintenance action is generated for them to be removed from the field). After that, all the implemented physical signs are checked for validity in PhyFM (including a recomputation of the sign track data), and any mismatches between the ideal and the actual data are reported as update-or-remove requests for maintenance. The signpost planner can add or remove any logical signs or change their necessity level (S_i). The logical closure of the set of logical signs is, however, always re-generated after any changes and the signs eventually are in S_0 will be then marked as (to be) implemented signs. The sign track data is generated for them and they are ready for the signpost designer to craft them onto real sign and information boards, and the maintenance staff to organize their production and installation.

If such a sign is to be installed outside existing nodes, a new trail node in the network must be added. The system should implement it as a seamless way, similarly to how a new (pseudo-)junction is added, so users are not bothered with extra operations.

Implementation of the logical rules can be based on database constraints and triggers. The rules are safe as they have no implied negations, and finiteness of closure is ensured by: the finiteness of the underlying nodes and routes, the limitation of i for S_i and the fact that the only rules generate structurally new logical signs imply them into S_i's with higher i as of their preconditions.

A further characterization of allowed logical rules, or defining a more sophisticated logics (e.g. considering to frame it as a proper modal logic) and their implementation schemes is a future issue, but the aim is to make a generic system which is flexible for multiple signage systems with different signpost policies.

7. Conclusion and Future Work

In this paper, we have introduced a terminology and the skeleton of TrailSigner, a conceptual model suite for hiking trail networks of various signage system types, which goes beyond the particular models of current platforms and can handle special or complex situations in an integrated way. It can be refined and implemented as a basis of an integrated, collaborative information system for multiple stakeholders and visitors for the consistent planning, promotion, management and maintenance of trails and their signage assets. This will reduce potential confusion, mistrust and danger for visitors caused by information mismatches including incomplete, incoherent or inconsistent trail route information or signposting.

To ensure consistency of incrementally planned signposts with each other and with the (possibly changing) underlying trail network, a novel, systematic, set-based approach has been developed with generative rules for managing logical trail signs, which mediate the physical implementation and trail signage facility management. An extensive example ruleset is developed for signpost logics, and incorporated into the conceptual model suite.

The next steps can be the refinement of the model, defining model suite connections and collaboration patterns for different phases of the network developments, and interaction modeling with storyboarding and mini-stories for possible event scenarios. Creating a reference implementation and testing with an existing complex regional trail network and signpost system is an on-going work [22].

Future work may include personalization features for different visitors and situations (virtual signposts, adaptive route recommendation, feature engineering for ML methords, etc), designing an easy-to-use GUI, investigating on automatized location generation and poi assignments, enhancing the OpenStreetMap data model and implementing an on-line tool for checking the consistency of existing signposts and/or recommending trip routes, and ultimately, to develop a fully-fledged pilot system implementation with complex trail networks of different types. This approach may further be generalized, personalized and adapted to other application fields having similar requirements or phenomena.

References

[1] OpenStreetMap Wiki. `https://wiki.openstreetmap.org`, [Online; accessed 25-Jan-2020]

[2] Wikiloc - trails of the world, `https://www.wikiloc.com/`, [Online; accessed 26-Jan-2020]

[3] General principles for signing and marking footpaths. `http://www.era-ewv-ferp.com/fileadmin/user_upload/dokumenter/Walking_in_Europe/Waymarking_in_Europe/General_principles__waymarking.pdf` (2015), [Online; accessed 20-Jan-2020]

[4] Waymarking in Europe. `http://www.era-ewv-ferp.com/fileadmin/user_upload/dokumenter/Walking_in_Europe/Waymarking_in_Europe/ENG_WAYMARKING_2017_01.pdf` (2017), [Online; accessed 20-Jan-2020]

[5] Digitales wegemanagement - wege. Ein pfad voller daten. Tourismus Digital (2018)

[6] Online-software NatursportPlaner. digitales wegemanagement für reale erlebnisse. `https://wanderverband.de/conpresso/_data/PM_33_Wandern_Digital.pdf` (12 2018), [Online; accessed 20-Jan-2020]

[7] OuterSpatial - a stewardship-first approach to data, maps and apps for recreation (2019), `http://www.alaska-trails.org/2019-trails-conference-presentations.html`, [Online; accessed 25-Jan-2020]

[8] Abiteboul, S., Vianu, V.: Datalog extensions for database queries and updates. Journal of Computer and System Sciences **43**, 62–124 (08 1991). https://doi.org/10.1016/0022-0000(91)90032-Z

[9] Anikó, K., Dávid, M., Dénes, M., Ferenc, P., Gábor, T.: Az Országos Kékkör táblarendszerének kialakítása térinformatikai eszközökkel. In: Theory Meets Practice in GIS VI. - Térinformatikai Konferencia és Szakkiállítás. Debreceni Egyetem (2015), `http://giskonferencia.unideb.hu/arch/GIS_Konf_kotet_2015.pdf`

[10] Ayala-Gómez, F., Daróczy, B., Mathioudakis, M., Benczúr, A., Gionis, A.: Where could we go? recommendations for groups in location-based social networks. In: Proceedings of the 2017 ACM on Web Science Conference. pp. 93–102. WebSci '17, Association for Computing Machinery, New York, NY, USA (2017)

[11] Brämer, R.: Heimliche revolution. Was bringen die neuen wanderleitsysteme? `https://www.wanderforschung.de/files/wanleitneu1233090989.pdf` (05 2003), [Online; acc. 20-Jan-2020]

[12] Brancifort, R., Likins, Z.T., O'Connor, M., Kleinknecht, S., Nuñez, D.: Connecting 10 million people to the outdoors in Los Angeles County - Trails LA County mobile app (2019), https://www.americantrails.org/resources/connecting-10-million-people-to-the-outdoors-in-los-angeles-county, [Online; accessed 25-Jan-2020]

[13] Bühler, G., Hermann, M., Lambertus, M.: Sehnsuchtsort natur in der digitalen leistungsgesellschaft. https://sotomo.ch/site/sehnsuchtsort-natur (05 2019), [Online; accessed 20-Jan-2020]

[14] Bundesamt für Strassen (ASTRA), Schweizer Wanderwege, Bern: Signalisation Wanderwege (2013), [Online; accessed 22-Jan-2020]

[15] Calbimonte, J.P., Martin, S., Calvaresi, D., Zappelaz, N., Cotting, A.: Semantic Data Models for Hiking Trail Difficulty Assessment, pp. 295–306 (01 2020)

[16] Car, A., Frank, A.: Hierarchical street networks as a conceptual model for efficient way finding. Proceedings of the 4th European Conference on Geographical Information Systems **1** (02 1970)

[17] Demetrovics, J., Molnár, A., Thalheim, B.: Graphical reasoning for sets of functional dependencies. In: Conceptual Modeling - ER 2004, 23rd International Conference on Conceptual Modeling, Shanghai, China, November 2004, Proceedings. pp. 166–179 (11 2004)

[18] Demetrovics, J., Molnár, A., Thalheim, B.: Relationship design using spreadsheet reasoning for sets of functional dependencies. In: Advances in Databases and Information Systems, 10th East European Conference, ADBIS 2006, Thessaloniki, Greece, September 3-7, 2006, Proceedings. pp. 108–123 (09 2006)

[19] Gramer, E.: Outdooractive Facility: Der wanderhimmel wird digital. https://corporate.outdooractive.com/oa-blog/outdooractive-facility-der-wanderhimmel-wird-digital (06 2016), [Online; accessed 20-Jan-2020]

[20] Lera, I., Pérez, T., Guerrero, C., Eguíluz, V.M., Juiz, C.: Analysing human mobility patterns of hiking activities through complex network theory. In: PloS one (2017)

[21] Márkus, Z., Wagner, B.: GUIDE@HAND: digital GPS based audio guide that brings the past to life. In: Pavlov, R., Stanchev, P. (eds.) Digital Preservation and Presentation of Cultural and Scientific Heritage. pp. 15–25. Bulgarian Academy of Sciences, Sofia (2011), http://eprints.sztaki.hu/6712/

[22] Molnár, A.: The TrailSigner project repository on Github. https://github.com/zarandras/trailsigner, [Online; updated 15-May-2020]

[23] Molnár, A.J., Thalheim, B.: Usage models mapped to programs. In: New Trends in Databases and Information Systems. pp. 163–175. Springer International Publishing, Cham (2019)

[24] Outdooractive: Outdooractive - Data Model, [Online; accessed 20-Jan-2020]

[25] Paton, N.W., Díaz, O.: Active database systems. ACM Comput. Surv. **31**(1), 63–103 (Mar 1999)

[26] Semenov, A., Zelentsov, V., Pimanov, I.: Application suggesting attractive walking routes for pedestrians using an example of saint-petersburg city. Procedia Computer Science **156**, 319–326 (2019), 8th International Young Scientists Conference on Computational Science, YSC2019, 24-28 June 2019, Heraklion, Greece

[27] Steinmetz, D., Dyballa, D., Ma, H., Hartmann, S.: Using a conceptual model to transform road networks from OpenStreetMap to a graph database. In: Trujillo, J.C., Davis, K.C., Du, X., Li, Z., Ling, T.W., Li, G., Lee, M.L. (eds.) Conceptual Modeling. pp. 301–315. Springer International Publishing, Cham (2018)

[28] Thalheim, B.: Model suites for multi-layered database modelling. In: Information Modelling and Knowledge Bases XXI, volume 206 of Frontiers in Artificial Intelligence and Applications. pp. 116–134. IOS Press (2010)

[29] Thalheim, B.: Foundations of entity-relationship modeling. Annals of Mathematics and Artificial Intelligence **7**, 197–256 (03 1993). https://doi.org/10.1007/BF01556354

[30] Thalheim, B.: Entity-Relationship Modeling: Foundations of Database Technology. Springer (2000)

[31] Tropmann, M., Thalheim, B.: Mini story composition for generic workflows in support of disaster management. In: DEXA 2013. pp. 36–40. IEEE Computer Society (2013)

[32] Wimmer, H.: Outdooractive Facility oder Natursportplaner? https://corporate.outdooractive.com/oa-blog/outdooractive-facility-oder-natursportplaner (04 2019), [Online; accessed 20-Jan-2020]

[33] Witt, P.J.: The development of a predictive hiking travel time model accounting for terrain variations. In: T. Jekel, A. Car, J. Strobl, & G. Griesebner (Eds.), GI_Forum 2012: Geovisualization, Society and Learning. pp. 102–112. Salzburg (2012)

[34] Zhi-ren, S., Min, H., Hai-bin, W.: A conceptual multi-level data model for road networks. In: Fifth International Conference on Intelligent Computation Technology and Automation. pp. 712–715 (Jan 2012)

A. Appendix: A Reference Axiomatization for Signpost Logics

A.1. General Notations, Atoms, Predicates and Functions

N1. Letters *q, r, ...* denote (simple) trail routes.

N2. Capital letters *M, N, ...* denote trail nodes.

N3. $Location(N,D)$: Node N is at named location D (by RouLocNet and GeoTrNet).

N4. Route predicates (based on RouLocNet and GeoTrNet):
$Route(N,r,M)$: Node N is (not necessarily directly) followed by M along r.
$IsEndpoint(r,M)$: M is the end node of r.
$IsReversed(r,q)$: r is the reversed version of q (by RouRel in TourPres).
$HasIdenticalWaymark(r,q)$: routes having the same waymark (causes ambiguous junctions without direct identification)

N5. Sign types:
$locSign(N,D)$: location sign denoting D at node N,
$rdtSign(N,r,\sigma)$: route destination sign placed at N showing direction towards the destination of the follow-up sign σ (being either a $locSign$ or another $rdtSign$, by structural recursion) via route r,
$rdtSign(N,r,\top)$: a symbolic sign atom that means route r should be signposted at N, i.e. a sign $rdtSign(N,r,\sigma)$ with some destination should be added by the signage planner if there is no one yet. If no destinations are added, it is interpreted as a route sign (only a route identification showing its continuation).

N6. Place of a sign - the node where it is located:
$at(locSign(N,D)) = N$, and D is the name of the location: $Location(N,D)$
$at(rdtSign(N,r,\sigma)) = N$ and σ is either a $locSign$ or a $rdtSign$ (by structural recursion) or the generic $\top$ symbol.

N7. Destination of a sign - the location where it points to:
$to(locSign(N,D)) = D$, where $Location(N,D)$.
$to(rdtSign(N,r,\sigma)) = to(\sigma)$ and σ is either a $locSign$ or a $rdtSign$ (by structural recursion).

N8. Necessity (implementation priority) level of a sign:
$S_0(\sigma)$: planned/existing/necessary (exists or is going to be implemented),
$S_i(\sigma)$: recommended with priority $i = 1,2,...,d$ (the lower the more necessary), where d is a small defined natural constant.

N9. Invalidity of a sign: $Inv(\sigma)$: the sign is invalid (it was generated earlier and does not match the current network data) and must be removed from the set of current logical facts.

A.2. Checking Rules

If a checking rule fails, the corresponding sign gets an invalid status and must be inactivated.

C1. True location principle: location signs have to be placed faithfully (checking rule):
$\neg(S_i(locSign(N,D)) \rightarrow Location(N,D)) \Rightarrow Inv(locSign(N,D))$

C2. Pointing forward principle: signposts show destinations along routes in their direction (checking rule):
$\neg(S_i(rdtSign(N,r,\sigma)) \wedge \sigma \neq \top \rightarrow Route(N,r,at(\sigma))) \Rightarrow Inv(rdtSign(N,r,\sigma))$
$\neg(S_i(rdtSign(N,r,\top)) \rightarrow \exists M \, : \, Route(N,r,M)) \Rightarrow Inv(rdtSign(N,r,\top))$

C3. Propagation of invalidity by structural recursion:
$Inv(\sigma) \wedge S_i(rdtSign(N,r,\sigma)) \Rightarrow Inv(rdtSign(N,r,\sigma))$

A.3. Generative Rules

Note: every rule has an implicit pre-requisite that the referenced signs are valid ($\forall\sigma$ in the rule body: $\neg Inv(\sigma)$).

G1. Signposted location identification: Arrival to a previously signposted destination must be confirmed by a location sign:
$S_i(rdtSign(N,r,locSign(M,D))) \Rightarrow S_i(locSign(M,D))$

G2. Waymark disambiguation: if two different routes with the same waymark meet and continue diverted, both need signposting:

$Route(N,r,M) \land Route(N,q,L) \land r \neq q \land M \neq L \land \neg IsReversed(r,q) \land$
$HasIdenticalWaymark(r,q) \Rightarrow S_0(rdtSign(N,r,\top)) \land S_0(rdtSign(N,q,\top))$

G3. Related recommendations: if a route is signposted then recommendations on the next priority level should be generated:
$S_i(rdtSign(N,r,\sigma)) \land \sigma \neq \top \Rightarrow S_{i+1}(rdtSign(N,r,\top)), i < d$

G4. Formal completeness: if a route is specifically signposted at a point, it is also generally signposted:
$S_i(rdtSign(N,r,\sigma)) \Rightarrow S_i(rdtSign(N,r,\top))$

G5. Location identification recommendation: if a route is signposted at a named location then a location sign is also recommended (note that a node may belong to multiple locations, and in this case, an explicit decision is possible):
$S_i(rdtSign(N,r,\sigma)) \land Location(N,D) \Rightarrow S_{i+1}(locSign(N,D)), i < d$

G6. Completeness of routes at a signposted location: (only as recommendation on the next priority level, to make the system more flexible, but can be strengthened):
$S_i(rdtSign(N,r,\top)) \land Route(N,q,M) \Rightarrow S_{i+1}(rdtSign(N,q,\top)), i < d$
Note: This will imply the derived necessary signposts for route q generated as S_{i+1} and the related recommentations will be S_{i+2}

G7. Traversing: when a signed destination is out of the first followed route, the destination must be signposted at the place of route transfer:
$S_i(rdtSign(N,r,rdtSign(M,q,\sigma))) \Rightarrow S_i(rdtSign(M,q,\sigma))$

G8. Forward-continuity: if any destination has been signposted along a route, it must be repeated at each signposting place where any destination is signposted for the same route, until the destination is reached or the route is changed:
$S_i(rdtSign(N,r,\sigma)) \land S_i(rdtSign(M,r,\top)) \land$
$Route(N,r,M) \land Route(M,r,at(\sigma)) \Rightarrow S_i(rdtSign(M,r,\sigma))$

G9. Recommendation of subsequent place: If signposting is planned at a place, then it is suggested to choose the next signpost location (if named) or named location as a destination (note: for avoiding indetermininsm caused by this rule, all implications for level S_i must be derived before any implication for S_{i+1} is generated):
$S_i(rdtSign(N,r,\top)) \land Route(N,r,M) \land Location(M,D) \land$
$(NextSignpostAt(N,M,r) \lor NoNamedLocBtw(N,M,r))$
$\Rightarrow S_{i+1}(rdtSign(N,r,locSign(M,D))), i < d$ where:

(a) no signpost is along r between N and M:
$NextSignpostAt_i(N,M,r) = S_i(rdtSign(M,q,\top)) \land$
$\nexists L,j : (Route(N,r,L) \land Route(L,r,M) \land S_j(rdtSign(L,r,\top)) \land j \leq i)$

(b) no named location is along r between N and M:
$NoNamedLocBtw(N,M,r) = \nexists L,E : Location(L,E) \land Route(N,r,L) \land Route(L,r,M)$

G10. Recommendation of destinations signposted at subsequent place(s): If signposting is planned at a place, then it is suggested to choose the destination dispayed on the next signpost(s) along the route(s) (continued below in multiple variants):
$S_i(rdtSign(N,r,\top)) \land Route(N,r,M) \land NextSignpostAt_i(N,M,r)...$

(a) along the same route (primarily):
$... \land S_j(rdtSign(M,r,\sigma)) \Rightarrow S_{\max(i,j)+1}(rdtSign(N,r,\sigma)), i,j < d$

(b) along a different route (secondarily):
$... \land S_j(rdtSign(M,q,\sigma)) \land q \neq r \land \neg IsReversed(r,q)$
$\Rightarrow S_{\max(i,j)+2}(rdtSign(N,r,rdtSign(M,q,\sigma))), i,j < d-1$

(c) along a different route if the next sign is at the end of the route, which is not a named locaction (primarily):
$... \land S_j(rdtSign(M,q,\sigma)) \land q \neq r \land \neg IsReversed(r,q) \land IsEndpoint(r,M) \land$
$\neg \exists D : Location(M,D) \Rightarrow S_{\max(i,j)+1}(rdtSign(N,r,rdtSign(M,q,\sigma))), i,j < d$

G11. Recommendation of route endpoint: if a route is being signposted at a place, and the endpoint of the route is a named location, then it is recommended (secondarily) to be chosen for signposting:
$S_i(rdtSign(N,r,\top)) \land Route(N,r,M) \land Location(M,D) \land IsEndpoint(r,M)$
$\Rightarrow S_{i+2}(rdtSign(N,r,locSign(M,D)))$