Semantically Enriched Multi-Modal Routing

Thomas Eiter[1], Thomas Krennwallner[1], Matthias Prandtstetter*[2], Christian Rudloff[2], Patrik Schneider[1], and Markus Straub[2]

1. Institut für Informationssysteme, Technische Universität Wien, Favoritenstraße 9-11, 1040 Wien, Austria, T: +43 1 58801 18405,{eiter,tkren,patrik}@kr.tuwien.ac.at
2. AIT Austrian Institute of Technology, Mobility Department, Giefinggasse 2, 1210 Wien, Austria, T: +43 50550 0, first.last@ait.ac.at

* corresponding author

Abstract.

We present an innovative extension to routing: *intention-oriented routing* which is a direct result of combining classical routing-services with Semantic Web technologies. Thereby, the intention of a user can be easily incorporated into route planning. We highlight two use cases where this hybridization is of great significance: *neighborhood routing* where a neighborhood can be explored (e.g. searching for events around your place) and *via routing* where errands can be done along a route (e.g. buying ingredients for your dinner on your way home). We outline the combination of different methods for achieving these goals. The emerging framework is demonstrated by two case studies. Finally, we give a short outlook on future work within this novel research field.

Keywords:
Multimodal Routing, Semantic Web Technologies

1 Introduction

In this paper we describe the development of two extensions to an existing routing service. While many services exist for both motorized private transport (MPT) and public transport (PT) and to some extend to multimodal transport, there are still many restrictions to the functionalities of routing services. Two such restrictions are addressed in the Austrian research project *MyITS* (My Personalized Intelligent Mobility Service).

The first restriction addressed is that routing services only offer simple routes from a given starting point to a supplied end point along one route. Many services like the router offered by Google or PT routers available through PT providers (e.g., http://anachb.at in Greater Vienna) give the shortest route and some alternatives from one specified point on a map or address to another. Sometimes intermediate points can be specified, so trips with several legs can be found. However, to the best of our knowledge, routing is not offered where the user's trip intention is given to the router, offering a more flexible routing, that is not relying on the user's actual goal or address, but rather on her intended activity. In the remainder of this paper we describe our methodology of enriching the routing service in this way by using Semantic Web technologies in connection with routing algorithms.

Two applications of this intention-oriented routing are given. The first one is neighborhood routing. While several unimodal neighborhood routers exist (e.g. http://www.tom-carden.co.uk/p5/tube_map_travel_times/applet/, last viewed Sept. 2012) and some services give geographical neighborhoods around PT stations as the reachable area (e.g. http://www.mapnificent.net/, last viewed Sept. 2012), no service actually uses multimodal routing, i.e., includes for example the region reachable by foot around transit locations within the given time interval. The approach presented here searches for multiple locations within a certain travel time from a starting point using intermodal routing. This neighborhood routing does not offer the traveler different routes to a single location but the choice of locations plus intermodal routes to those locations, e.g. all supermarkets within a 5 minute journey by foot, bike, PT, MPT, or a combination of these modes.

The second application of intention-oriented routing is a routing service that offers the possibility to search for a route with specific start and end points but gives the users the chance to add an intermediate activity to their route (via routing). Corresponding intermediate points or via points are automatically determined and appropriate routes via these points are computed which are then presented to the user. For example, a pharmacy offering homeopathic products can be searched for which is located along a route home from work.

The combination of two technologies lies at the heart of this intention-oriented routing approach. The first is Semantic Web technology, which provides the data backbone for taxonomical information used to search for semantically-enriched points of interest (POIs). E.g., POIs having a category such as Chinese restaurant. The second is a routing algorithm that considers not only one mode of transport (MOT) but all available MOTs during the same routing request resulting in intermodal routes.

Flexible data integration and querying of multiple data sources is the key to our approach, as autonomous data vendors provide heterogeneous data that is used to answer queries associated with the intention of the user. Using various data sources of substantial size gives the opportunity to find intended POIs, which may fall into multiple concepts ranging from rather unspecific to more detailed such as “restaurant” versus “pizzeria.” Moreover, we can exploit the structure of the taxonomical information that is implicitly stored in the data sources by making them concrete with an ontology. Such ontology-based data access can be used to answer broad queries like “restaurants with Italian cuisine,” that should return pizzerias and classical Italian restaurants. Further, it is feasible to seamlessly add new data from fresh sources, which allows to generate more accurate answers for the extended domains.

Apart from a shortest route algorithm the routing also gives a methodology to find (intermodal) isochrones that give all the links in a routing graph that can be reached within a specified time as well as methodologies to find corridors around the routes that are used as input regions to the location-based semantic search. A more detailed description of the combination of routing and semantic search is introduced in the sequel. In Section 2 we introduce the technologies that are applied in the enriched routing tool together with the relevant background and existing work on the subject. The details of the interaction of routing and semantic search are given in Section 3. Afterwards the two applications of the novel routing service are given, in Section 4 we describe the neighborhood search and in Section 5 we give details and examples of the via routing. Finally we present conclusions to our work and give an outlook to future improvements.

2 Background Technologies

2.1 Path, Neighborhood, and Corridor Computations

Path Computations. The problem of finding a (shortest) route from a given origin to a given destination is defined on an (un)directed graph where a weight is assigned to each edge (arc). In the general case (where edge/arc weights are real numbers) the algorithm of Bellman and Ford [7] has to be applied whose runtime is in $O(n^3)$, with n denoting the number of nodes in the graph. However, with respect to transportation science, the edge weights given are in the most cases non-negative for all edges of the graph. Therefore, Dijkstra’s algorithm [9] can be applied which solves the shortest path problem using a simple implementation in time $O(n^2)$.

(a) Buffering around a route. (b) Enclosed “unreachable” areas. (c) Public transport.

Figure 1 - Exemplary corridors (green area) for a route, and a neighborhood using public transport. Unreachable areas are shown in white.

Neighborhood Computation. Although the term neighborhood is widely used, in this paper the term is used in its geographic meaning: a region around a point where each point lying inside the area can be reached within at most δ minutes. We also write in short δ-neighborhood. Since we assume that all edge-weights given in our transportation network are non-negative, a δ-neighborhood can be easily computed via Dijkstra’s algorithm [9] by aborting the algorithm as soon as every unfinished node has costs greater than δ. All nodes finished so far, i.e., all nodes for which the minimal path from the origin has already been computed, are those points lying within the δ-neighborhood of the origin.

Corridor Computations. Although the methods presented above are theoretically sufficient to compute shortest routes as well as neighborhoods in the given setting they are not sufficient to handle real-world location based data: in most cases GPS (or other geo-referenced) data sets are not directly assigned the edges and/or nodes in an underlying graph. To determine whether a POI, e.g. a pharmacy, is located along a route (or can be reached within at most δ minutes, it is necessary to map this POI to edges and nodes of the graph. There are several methods to do this: e.g. by classical map-matching [17]. These methods suffer, however, from the problem that each time an update is applied to the setup (e.g. map update) the mappings for all POIs have to be (re)computed in the worst case.

Therefore, we decided to apply corridor methods which compute a hull enclosing all edges and/or nodes in question. E.g. for a given route the enclosing hull (or corridor) is computed by applying a buffer method, where a buffer is “a geometry that contains all points whose distance from [the route] is less than or equal to [a predefined threshold]” (also referred to as the *dilated hull*), cf. http://www.postgis.org/docs/ST Buffer.html.

While these methods apply well to routes (see e.g., Fig. 1a), they are not satisfactory for neighborhoods since artifacts may occur. For example, blocks of buildings might be marked as unreachable although they are in fact reachable, cf. Fig. 1b. Therefore, we investigated several other hull computation methods giving convex or non-convex hulls for the neighborhood computation. The convex hull computation is one of the standard hull computation methods in literature. An apparent advantage of it is the existence of efficient algorithms, e.g., Graham’s scan, with the lower bound of $O(n \log n)$, where n is the set of points in the plane to be enclosed (see [2]). However, for certain applications the convex hull does not provide desired results. In particular the convex hull for long routes or corridors may contain regions that are not sensible search regions for the semantic search (e.g. the convex hull of a route along a large ring road would contain most of a city). Several non-convex hull computation were introduced to obtain finer shapes. One of these methods are alpha-shapes, which were introduced by Edelsbrunner et al. [10] as a generalization of the convex hull. The construction of an alpha-shape is performed by an intersection of all closed discs with a given radius that contain the set of points in the plane.

Please note that in case public transport is incorporated, the corridor might consist of non-connected and/or overlapping regions, see Fig. 1c, since public transport stations act as sub-origins. Hence, advanced hull computations need to be applied for this case.

2.2 Semantic Web technologies and Geospatial Databases

Official data sets (provided by governments for public use) and collaborative projects like OpenStreetMap (OSM) are becoming large sources of spatial data. Geospatial databases are the backbone for storing and querying these data (cf. [8, 13]). These databases often have the drawback that querying them is complicated, inference mechanisms are non-existent, and extending them to the new data sources is difficult, due to extracting, transforming, and loading steps. In response, Semantic Web technologies are becoming more interleaved with geospatial databases ([cf. 4, 17]), which should lead to an easier integration and querying of spatial data.

We refer to the seminal article of Berners-Lee et al. [3] for an outline of the ideas and to the Semantic Web Stack (http://www.w3.org/2007/03/layerCake.svg) for an architectural overview of the Semantic Web. In particular, ontologies are used for modeling knowledge domains, by expressing relations between terms and by modeling them as a taxonomy. In the Semantic Web context, the standard modeling language OWL [14] with its (formal) logical underpinning of Description Logic (DL) plays a central role. The vocabulary of a DL consists of individuals, classes, and roles. A knowledge base consists of a terminological box, which contains axioms about relations between classes and roles, and an assertional box, which contains factual knowledge about individuals (cf. [1]).

As shown in the MASTRO system [5], these technologies are more than an advancement of the WWW. Therewith, ontologies are applied as a feasible way to integrate data sources through a global schema expressed as them. The integration, also called global-as-view approach [15], is a set of mapping assertions which maps source schemas to a global schema by first-order logic (FOL) queries.

3 Methods

Although the methods applied within this project (routing and semantic search technologies) are well-understood on their own, the proposed combination of these techniques is rather novel and may lead to computational expensive tasks if not done properly. We propose to perform the incorporated methods in sequential order, meaning that either a semantic search constrained by the routing results (pre-filtering) or route computations are performed based on the result set returned by the semantic search (post-filtering). Furthermore, we illustrate the interleaving of Semantic Web technologies and geospatial databases in the MyITS project.

3.1 Pre- and Post-Filtered Semantic Search

During a pre-filtered semantic search the query itself is constrained by the results of a previously performed routing phase. This approach is especially helpful whenever either the number of candidate points of interest (POIs) returned by an unconstrained semantic search is rather large or if the constraints stated by the user with respect to the routing are rather hard (e.g. only POIs no farther away than five minutes). Therefore this technique is mainly applied for neighborhood routing (cf. Sec. 4). Nevertheless, neighborhood routing takes advantage of this technique, if the number of POIs returned by the semantic search is too large (cf. Sec. 5).

Contrary, post-filtered semantic search consists of an unconstrained semantic search first returning all candidate POIs, which are subsequently processed for routing. Since route computations have to be performed for each POI obtained, this technique is currently only applied for the via routing (cf. Sec. 5) if the number of candidate POIs is rather small. This approach, however, imposes no limitations on the semantic search as well as the routing procedure.

3.2 Method Selection and Route Choice

Since our framework supports two possible sequential executions of its tasks, we need a way to determine which one should be applied for a concrete user request (this only applies for via routing). For this purpose, a simple determination heuristic is applied. First we perform an unconstrained semantic search and therefore obtain a set of candidate POIs. Instead of the cardinality of the set, database statistics regarding the concept assignments could also be obtained. If the cardinality of this set is above a predefined threshold pre-filtering is applied. Otherwise we continue with the post-filtering procedure.

Independent of the method applied (pre- or post-filtering) in most cases more than one possible route to choose from is returned (mainly because in most cases the number of candidate POIs is larger than one). Therefore, it is further necessary to rank all obtained routes which can done using straightforward evaluation strategies like (shortest) travel time or (shortest) distance. Furthermore, more advanced methods can be applied. For example a weighted sum of the route length/travel time and the evaluation of the POI (how much does the POI match the search criterion originally entered by the user) can be computed for each POI. The final decision is, however, done by the user who chooses her preferred route/POI among the ordered list of proposals.

3.3 Upper Level Ontology

In our case the Ontology-Based Data Integration with the global schema is represented by an OWL 2 QL [6] upper level ontology. The upper level ontology is tailored on the one hand to geospatial data sources and beyond to MyITS specific sources (e.g., the restaurant guide of Falter at http://www.falter.at/web/wwei/). For the top level of the ontology, we build on already defined work with GeoOWL (http://www.w3.org/2005/Incubator/geo/XGR-geo/). However, for our needs the (geospatial) feature concept of GeoOWL is too general, so we introduced a more detailed categorization for the second level based on the nine top-features of GeoNames (such as area feature, road feature, building feature, etc. from http://www.geonames.org/). The third level is build mainly from OSM categories which are relevant for the MyITS project and is designed to be extended, if we include new data sources. A further extension of the first level is the concept attributes, which is meant to extend the feature classes through roles to attribute classes with additional taxonomies. For instance, we extend the restaurant class by an attribute concept that defines the served cuisines.

3.4 Mapping and Identities of/between Spatial Objects

The mapping between spatial objects and the ontology can either be done on-demand by FOL queries (as in MASTRO) or can be calculated preliminarily and materialized in an extension of the assertional box (ABox) (similar methods are used in annotations engines as KIM [16]). We adapted the latter for the following reasons:

(1) Rewriting is difficult, since domain specific heuristics are crucial for the assignment of objects to appropriate concepts and are hardly expressible in FOL queries (e.g., guessing the type of cuisine according to the name of the restaurant).
(2) A part of the mapping relies on external computation sources like geometry and string metrics engines.
(3) Data cleansing (e.g., duplicates) and inconsistency management is much easier to handle, if the mapping is materialized.

We apply a rule-based framework based on HEX-programs [11] for the purpose of declaratively defining the materialization. Furthermore, with HEX-programs default negation, recursion, and constraints are straight forward expressible. The following example illustrates a

mapping rule:
Example 1. IF objectType is “Amenity” AND objectName contains “Beisl” THEN assert it to concept *AustrianRestaurants*, to role *hasCusine Viennese*, and to role *hasGeometry Point*.

Having several data sources with similar objects assigned, large amounts of duplicate results will occur on query evaluation (e.g., between OpenStreetMap and Falter). For the purpose of reducing these duplicates, we estimate the identities between objects via a similarity measure. The calculation applies textual information (e.g. names), location, shape, and the position in the ontology for measuring the similarity. For this purpose, we materialze the identities by the rule-based techniques described above.

3.5 Query Answering

Query Answering is based on the evaluation of conjunctive queries (CQ) with a spatial and DL part over the combined spatial database and DL knowledge base. For the DL query part, we apply the first-order-rewritabilty of OWL 2 QL, which allows us to compile the terminological box into a SQL statement that can be evaluated over the ABox. The spatial query part is directly rewritten to spatial functions, which are available through GIS extensions (e.g. PostGIS for Postgres) in relational databases. Finally, the spatial and DL query part are combined by a JOIN based on the previous described mapping. Note that standard CQs can be immediately rewritten to SELECT−PROJECT−JOIN SQL statements. At this point, we do not considered keyword-based queries to CQs.

4 Neighborhood Routing

The focus of the first case is the so-called neighborhood routing where a user desires to explore the neighborhood for a specific class of POIs. We see several usages for this case, e.g.:

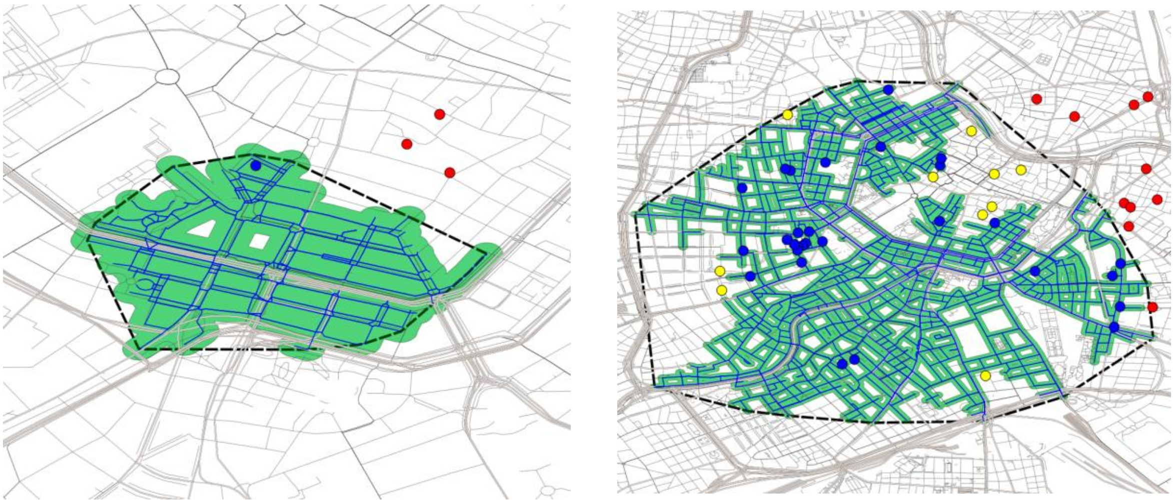
- a tourist looking for a coffee shop after visiting an attraction,
- a resident looking for a particular event (e.g. concert) after dining, or
- a newcomer checking reachable supermarkets by a certain modality in his area.

More formally, as input we consider a start coordinate *A*, a POI query *Q* as described in the previous section, a modality choice *M*, and a distance threshold δ in minutes. Please note, that the modality choice may consist of multiple modes of transport if intermodal routes are accepted. Then the following procedure is applied:

(1) compute the δ-neighborhood of A considering modality M resulting in a graph I of all reachable edges and nodes;
(2) based on graph *I*, calculate a corridor *C* by either the convex (or non-convex) hull or the dilated hull (buffer) operator using a computational geometry engine; and
(3) perform a semantic query on the knowledge base using the corridor *C* as spatial restriction leading to a set of POIs. The final result is built by calculating routes for a selection of POIs (e.g. the closest ones).

At this point it is assumed that at least one POI is located in the neighborhood. If this is not the case, there are basically two possibilities how to proceed: The user is informed that there is no matching POI in the δ-neighborhood. Alternatively, the threshold δ is iteratively enlarged until at least one matching POI can be reached. The user is then informed that the proposed POI is outside the originally intended δ-neighborhood. However, the second case may result in an infinite loop if no upper bound for the number of iterations is given.

Case Study. Assume that an opera buff, who does not favor the second act of Richard Wagner’s Parsifal, wants to have a snack at one of Vienna’s famous Würstlstände (some kind of Viennese hotdog stand). Due to the duration of the second act, she assumes that any Würstlstand no farther away than 5 minutes (either by walk or by taxi) would fit (cf. Fig. 2 showing the 5-minutes-neighborhoods for walking (a) and taxi (b).



(a) 5 minutes walking distance (b) 5 minutes taxi ride

Figure 2 - 5-minutes-neighborhood (blue edges), convex hull (dashed area), dilated hull (green area), and Würstlstände (differently colored points) starting at Vienna opera house.

Our case study is based on the POIs of greater Vienna contained in OSM. Due to the ongoing extensions of the mapping framework (e.g., with tourist spots, public buildings, etc.), only about one third of POIs contained therein are annotated with concepts (ca. 11k instances). In Figure 2a we can see that filtering the search results for “Würstlstand” by a convex or by a dilated hull are similar to each other, while in Figure 2b it is observable that 40 POIs are in the convex hull (yellow and blue points), while only 30 POIs are covered by the dilated hull (blue points); 10 POIS are not reachable at all (red points). It is noticeable that – depending on the distance and modality – the shape of the neighborhood may become more tree-like, having the effect that the proper reachable areas are covered more accurately by dilated hulls. More detailed investigations reveal that six of the differing yellow points shown in Fig. 2b are in a neighborhood of Vienna with limited car access.

5 Via Routing

In the via routing use case we are focusing on finding a route between a given origin-destination pair via some POI, which is dynamically determined by a semantic query.

We have identified several applications for this case, e.g.,

- a tourist wants to eat in a restaurant with a particular cuisine on his way to the hotel,
- a commuter needs to stop at a pharmacy, or
- a person invited for dinner needs to buy some flowers for the host.

Formally, given the origin A, destination B, POI query Q, mode choice M, and distance threshold δ in minutes, the task is to generate a route from A to B using M via one particular POI within a δ-neighborhood that is included in the answers to Q. In contrast to neighborhood routing the procedure applied within this case may vary depending on the size of the result set $\boldsymbol{P}$ of POIs obtained during a first (unrestricted) semantic search. For a small number of results, the procedure will continue with first computing a via route for each $P \in \boldsymbol{P}$, i.e., a route from A to P followed by a route from P to B, and then analogously to neighborhood routing, the obtained routes will be sorted and presented to the user.

However, if the cardinality of $\boldsymbol{P}$ is too large, a pre-filtered semantic search will be applied:

(1) compute a few alternative routes $\boldsymbol{R}$ from A to B optimizing a defined criterion, e.g., shortest travel time;
(2) compute corridor C for $\boldsymbol{R}$;

(3) obtain a new result set ***P*** of POIs via a semantic search constrained by *C*, i.e., disregard POIs located outside of *C*;
(4) compute for each $P \in \boldsymbol{P}$ a route consisting of two legs (from *A* to *P* and from *P* to *B*);
(5) sort the obtained set of routes and present it to the user.

In the second case, it is possible that despite a large result set *P*, we have no POI located within corridor C. As a consequence either the width of the corridor is iteratively enlarged or additional alternative routes are computed using (for example) adapted optimization criteria.

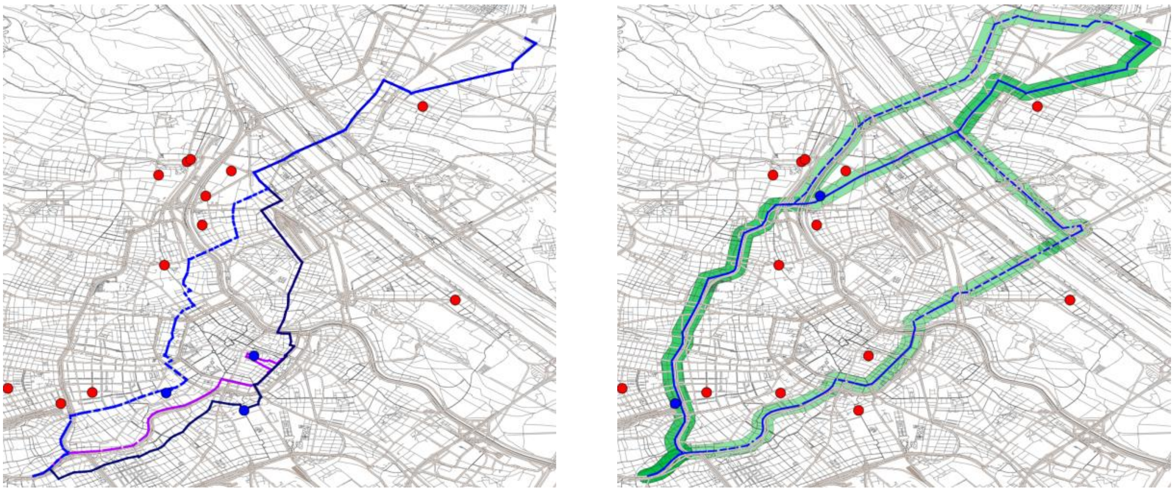
Case Study. Take, as an example, a father of two children, who is late from work and has to buy some Chinese food for his family on his way home. For this purpose, he is looking for a Chinese restaurant reachable by car, having a parking lot in front of the door and selling take-away food.

Since the number of Chinese restaurants in Vienna with a parking lot is around 17, both described methods could be applied. For the post-filtered semantic search, we evaluated the semantic query for the concept Restaurant with Chinese cuisine. Then, three candidate POIs were chosen for calculating a via route, resulting in six partial routes. In Fig. 3a the results are represented after checking if the candidates were within the distance threshold δ. The pre-filtered search is illustrated in Fig. 3b, where three alternative routes were calculated, which composed the base for calculating three corridors. We only considered the dilated hull for the corridors, since the convex hull for the three routes would cover almost the entire city. In contrast to the previous use case, the buffer size for the dilated hull was chosen 3 times higher; otherwise solely restaurants directly on the routes would have been obtained. Finally, the corridors where applied as a filter for the semantic query, resulting in two possible results.

The larger the set of possible POIs is becoming, the more efficient the second approach is becoming, since only a few alternative routes have to be calculated. On the other hand, when less POIs exists, the second approach is inefficient, because alternative routes or the corridor might have to be recalculated. One of the goals for the ongoing work will relate to a cost model regarding the choice of an approach with different sized sets of POIs.

6 Discussion and Outlook

In this paper we proposed a novel *intention-oriented routing* approach, arising from the combination of two state-of-the-art technologies: routing and semantic web technology. We developed a general framework incorporating the central methods and components and presented the application of them on two use cases: *neighborhood routing* and *via routing*. Both methodologies are executed sequentially in the framework leading to two possible strategies: pre-filtering and post-filtering. While the former consists of a semantic search which is constrained based on the results of a previously performed routing phase, the latter computes routes for all results obtained via an unconstrained semantic search. Due to challenges arising during data integration, we observed that it is necessary to provide corridors rather than concrete routes during a first phase of the final system. Within these corridors candidate POIs are then extracted via Semantic Web technologies. Therefore, the shape and composition of the corridors is of essential significance so that a proper result set meeting the user's intention can be generated. We primarily investigated two corridor computation methods, namely *convex hull* and *dilated hull computation*, both leading to distinct results for neighborhood routing. Furthermore, it could be observed that convex hulls are not meaningful with respect to via routing, where more complex hull computations could be performed, e.g. alpha-shapes.



(a) Routing via three distinct restaurants

(b) Two restaurants in the computed corridor

Figure 3 - The two alternative approaches for the rides home with Chinese restaurants shown as red spots with the best one shown in blue.

Within this work, we focused only on via routing via one specific point (which is extracted using Semantic Web technologies). It is, however, imaginable to extend the intention-oriented routing such that multiple via points are automatically determined, e.g. for finding a route from home to work via a super-market, a pharmacy, and a laundromat. It can be shown by a reduction from the well-known Generalized Traveling Salesman Problem that this problem is NP-hard, i.e., no polynomial time algorithm is likely to exist. However, if the sequence is predefined the problem can be solved via dynamic programming methods, cf. [12].

The second restriction addressed in MyITS but not within this paper, is that routing algorithms currently applied in online routing services are based on a shortest route algorithm, where either the shortest or fastest route is presented to the traveler. Sometimes other single optimization criteria are used to find the route. However, when travelers decide on a route, in particular in PT, there are often other criteria that are used to choose a particular route. People might prefer a route that uses commuter trains over buses, or they take a longer route rather than changing from one mode to another. This will be addressed in future research.

Our ongoing work focuses on a seamless integration of the developed components. This leads to the compare of the different methods regarding the quality of results and performance. Thereafter, we aim to address optimizations techniques and benchmarks for evaluation.

Acknowledgements

Supported by the Austrian Research Promotion Agency (FFG) project P828897, and the Austrian Science Fund (FWF) projects P20841 and P24090-N23. We also thank our project partners Rosinak & Partner ZT GmbH, Fluidtime Data Services GmbH, ITS Vienna Region, and new turn / DI Klaus Heimbuchner for their valuable input and support.

References

1. Baader, F., Horrocks, I., Sattler, U.: (2009). Description logics. In: Handbook on Ontologies, 21-43. Springer.
2. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M. (2008). Computational Geometry: Algorithms and Applications. Springer.
3. Berners-Lee, T., Hendler, J., Lassila, O. (2001). The semantic web. Scientific American 284(5), 34-43.
4. Bishr, Y.A. (2008). Geospatial semantic web: Applications. In: Encyclopedia of GIS, 391-398. Springer.
5. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R., Ruzzi, M., Savo, D.F. (2011). The Mastro system for ontology-based data access. Semantic Web 2(1), 43-53.
6. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R. (2007). Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. Journal of Automated Reasoning 39(3), 385-429.
7. Cormen, T.H., Leiserson, C.E., Rivest, R.I., Stein, C. (2001). Introduction to Algorithms, 2nd Ed. MIT Press.
8. DeMers, M.N. (2008). Fundamentals of Geographic Information Systems (4. ed.). Wiley.
9. Dijkstra, E.W. (1959). A Note on Two Problems in Connexion with Graphs. Numerische Mathematik 1, 269-271.
10. Edelsbrunner, H., Kirkpatrick, D.G., Seidel, R. (1983). On the shape of a set of points in the plane. IEEE Transactions on Information Theory 29(4), 551-558.
11. Eiter, T., Ianni, G., Schindlauer, R., Tompits, H. (2005). A uniform integration of higher-order reasoning and external evaluations in answer set programming. In: IJCAI 2005, Proceedings of the 19th international joint conference on Artificial intelligence, 90-96. Morgan Kaufmann.
12. Hu, B., Raidl, G.R. (2008). Effective neighborhood structures for the generalized traveling salesman problem. In: Evolutionary Computation in Combinatorial Optimisation – EvoCOP 2008, volume 4972 of LNCS, 36-47. Springer.
13. Jones, C.B. (1997). Geo. Information Systems and Computer Cartography. Prentice Hall.
14. Krötzsch, M., Patel-Schneider, P.F., Rudolph, S., Hitzler, P., Parsia, B. (2009). OWL 2 Web Ontology Language Primer. W3C Recommendation (Oct 2009).
15. Lenzerini, M. (2002). Data integration: A theoretical perspective. In: PODS 2002, Proceedings of the twenty-first symposium on Principles of database systems, 233-246. ACM.
16. Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., Goranov, M. (2003). Kim - semantic annotation platform. In: The Semantic Web - ISWC 2003, volume 2870 of LNCS, 834-849. Springer.
17. Wu, D., Zhu, T., Lv, W., Gao, X. (2007). A Heuristic Map-Matching Algorithm by Using Vector-Based Recognition. In: ICCGI 2007, Proceedings of the International Multi-Conference on Computing in the Global Information Technology, p. 18. IEEE Computer Society.