Applying Ambient Characteristics, Personalization and Ontology Concepts to Tourist Guide Route Generation

Georgi Kostadinov, Kristian Milev, Stefan Staynov, Asya Stoyanova-Doycheva

Faculty of Mathematics and Informatics, University of Plovdiv "Paisii Hilendarski", Plovdiv, Bulgaria
georg.kostadinov@gmail.com, kristianmilew@gmail.com, stefan.t.st@gmail.com, astoyanova@uni-plovdiv.net

Abstract. Assisting tourists by guiding them on their trip is neither a new, nor a trivial topic. People have different considerations when choosing which objects to visit while on tour. There is plenty of factors that could influence their choice and the more of those dynamic characteristics and related circumstances the algorithms support, the more valuable they are.

Keywords: Algorithm, Ambients, Ontology, User Preferences.

1 Introduction

Algorithms for route generation can take various parameters into account in order to suggest the next object to visit. The most fundamental property common among those algorithms is location. Without tourist's coordinates and the (latitude, longitude) pairs of the available objects, such an assistant would be a little more than an encyclopedia. In addition to location, there are other significant ambient characteristics that could be beneficial in tourist guides. Working hours of an exposition in conjunction with the time range available to the tourist should be considered important when picking the next place for them. Another example is accessibility – we don't want to suggest a place which does not provide access via a wheelchair to a person with a physical disability. In order to be able to adequately consider such ambient characteristics, we also need to have the tourist's preferences at our disposal. If they are equipped with an umbrella and they have confirmed they don't mind raining at an exposition, we wouldn't want to have the tourist miss a rare event just because of weather conditions, and vice versa – if the tourist declared they have only the next few hours available for a tour, we shouldn't suggest them an exposition opening after two hours or later. Ambient characteristics bring value when we have personalized information about the user we are serving. That's why application flow of the Tourist Guide (which is a specific adaptation of the ViPS reference architecture (Miteva, (Miteva, Stoyanova-Doycheva, & Ivanovna, 2018)) starts with the questionnaire where we collect user's personal preferences which the algorithm intersects with the ambient characteristics maintained for each object in order to determine if the object would be a good fit for the tour. Building

Digital Presentation and Preservation of Cultural and Scientific Heritage. Conference Proceedings. Vol. 12, Sofia, Bulgaria: Institute of Mathematics and Informatics – BAS, 2022. ISSN: 1314-4006, eISSN: 2535-0366

an ontology around the static objects is also something that a tourist could take advantage of. Grouping the instances by category or simply tagging them could empower the user to drive the filtering of objects eligible for a route. Maybe the tourist is interested in seeing ancient architectural buildings but museums is not their cup of tea. Why wouldn't we allow them to filter out the museums, if that's the case? Location, though, is still a crucial characteristic of the objects, so the algorithm gives the option to see all existing objects near the user location or all within a specified radius relative to an object being selected as a candidate.

2 Tourist Guide Architecture

The Tourist Guide is an N-tier implementation of the Virtual Physical Social Space (ViPS) architecture (Doychev, (Doychev, Stoyanova-Doycheva, Stoyanov, & Ivanova, 2016) where the presentation layer is a mobile client installed on tourists' devices. It communicates with the business layer which is implemented as a RESTful API running on a remote server. That major back-end component, in turn, relies on a relational database for storage.

2.1 Front-end

The front-end component is responsible for walking the user through a *Questionnaire* in order to collect those user preferences which the algorithm needs in order to perform its calculations in the context of the person on tour. In other words, the data collected by the questionnaire builds the user profile for the current tourist session. Once the route (either virtual or real) is constructed, it's the job of the user interface component to present the route in an intuitive manner, where in the case of real route the ordered list of objects to visit is easy to follow, while for virtual routes the information about the objects can be consumed with pleasure. The *Route Visualization* also handles the search flow, in case the tourist has a name, category or tag in mind, as well as navigation between suggested objects.

2.2 Back-end

The back-end sub-system serves a single top-level purpose – generates the best possible route for the tourist using the app. In order to make the algorithm work, we created models across three major sub-domains – ontology, ambients and personal preferences. The *Ontology* is utilized by both virtual (static) and real (ambient-aware) routers as it contains the grouping and classification of the objects (individuals / instances) to visit. It's what makes it possible for the user to drive the process of filtering the available instances by some criteria. It also contains the knowledge for the algorithm to be able to recommend similar (by as many traits as possible) items when the tourist is reviewing an object.

Ambients are just the ontology instances decorated with their ambient characteristics. They are relevant to the real routing only as virtual routes can be generated without those characteristics being considered. On the other hand, real routing can benefit from the ontology rules in order to make the router even more helpful. Combining the ambient characteristics with the *User Preferences* is what the algorithm is doing in order to rate the instances in a way that the most suitable ones are suggested on top of the recommendation result list.

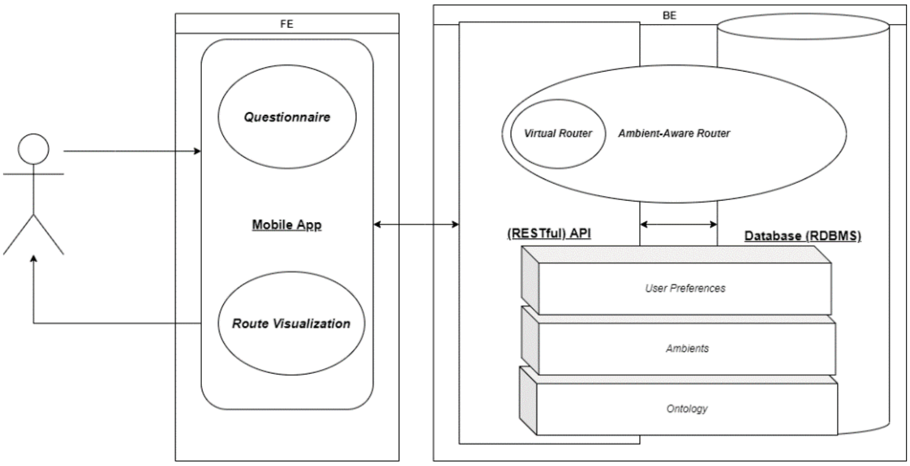

Fig. 1. Tourist Guide High-Level Architecture.

3 Prototype Implementation

One of the main features which distinguish this guide from similar tools (*e.g.,* Skripal, & Pyshkin, 2015) where the focus during route generation is around the user preferences and object popularity) is the ontology-based search. It takes into account predefined instance classification and ontology rules with or without matching ambient characteristics of those instances against user preferences during route construction. Its flow starts with a type-ahead field allowing the user to search through the hierarchy of the ontology. In case the search term matches a class of objects to visit and they click on the auto-completed suggestion, they get several different lists or trees relevant to the selected ontology class:

- A tree of sub-classes or related (with instances often shared) classes you can drill-down to.
- Individuals of the ontology class (or its sub-classes) ordered by distance where closest to the user device are on top.
- Individuals of the ontology class (or its sub-classes) filtered and ordered according to the user preferences in intersection with the objects' ambient characteristics and considering the location of the user device.

If the keyword resolves to a concrete ontology instance, the tourist is directly presented with the details of the instance with the ambient characteristics included. In addition, again, they receive several lists of alternative or follow-up options which the guide considers appropriate:

- Similar instances are those that have at least two shared ontology classes with the one opened in detail view. The more common classes they have, the more towards the top of the list they appear.
- Enclosing objects are those located near the selected one regardless of ontology classes.
- Members of any of the ontology classes of the object in context are filtered and ordered according to the user preferences in intersection with the ambient characteristics and considering the location of the object opened in detail view.

If the user doesn't feel like going through the search flow or the questionnaire, the route builder works in location-aware and working time-aware only mode, so that only nearby and currently open expositions are suggested as places to visit.

4 Route Generation Algorithm

There are two major flavours of route generation incorporated in the Tourist Guide which extend one another.

4.1 Virtual Router

The virtual routing strategy relies solely on the underlying ontology and uses reference data to allow the tourist to search through expositions, genres, categories, etc. and view detailed information about the ontology instances. Once the user opens a detailed view of an object, the virtual router automatically finds similar instances and suggests them (most relevant first) to the tourist as an option to virtually visit next. This strategy is user-agnostic and doesn't take into account ambient characteristics of the objects so it would provide the same results given the same input no matter what the user prefers or where they are located. Although it may feel like a fairly static, encyclopedia-like view of the world, this flavor of the algorithm is of high importance to the Tourist Guide in that it empowers the user to reduce the set of available objects down to only the sub-set they are interested in. Also, part of the recommendation functionality (suggesting objects with traits common with the selected one) resides in this segment of the algorithm.

4.2 Ambient-aware Router

Decorating the ontology instances with their ambient characteristics (*e.g.,* location, accessibility, exposure, working hours, *etc.*) and collecting user preferences in the context of those ambient characteristics allow the algorithm to transform an incoming static set of ontology instances (*primary route*) into a filtered, ordered list of ambients - *the final route*. The primary route is produced by either user searching through the ontology concepts or alternatively, the app retrieving all nearby objects available. The primary route forms the input for the ambient-aware real route generator. Each ambient characteristic is evaluated against its corresponding user preference and produces a real number between 0 and 2. Those numbers are multiplied which results in the final rating of the object of interest. So, if any ambient characteristic state is incompatible with the corresponding preference for the current user, the rating would be 0 which leads to excluding the object from the list of suggestions. For example, if the "Covid Certificate Requirement" characteristic for an exposition has the state `needsCovidCertificate: true` but the tourist declared they don't have a certificate, the object will be removed from the final route even if it's the closest one to the user.

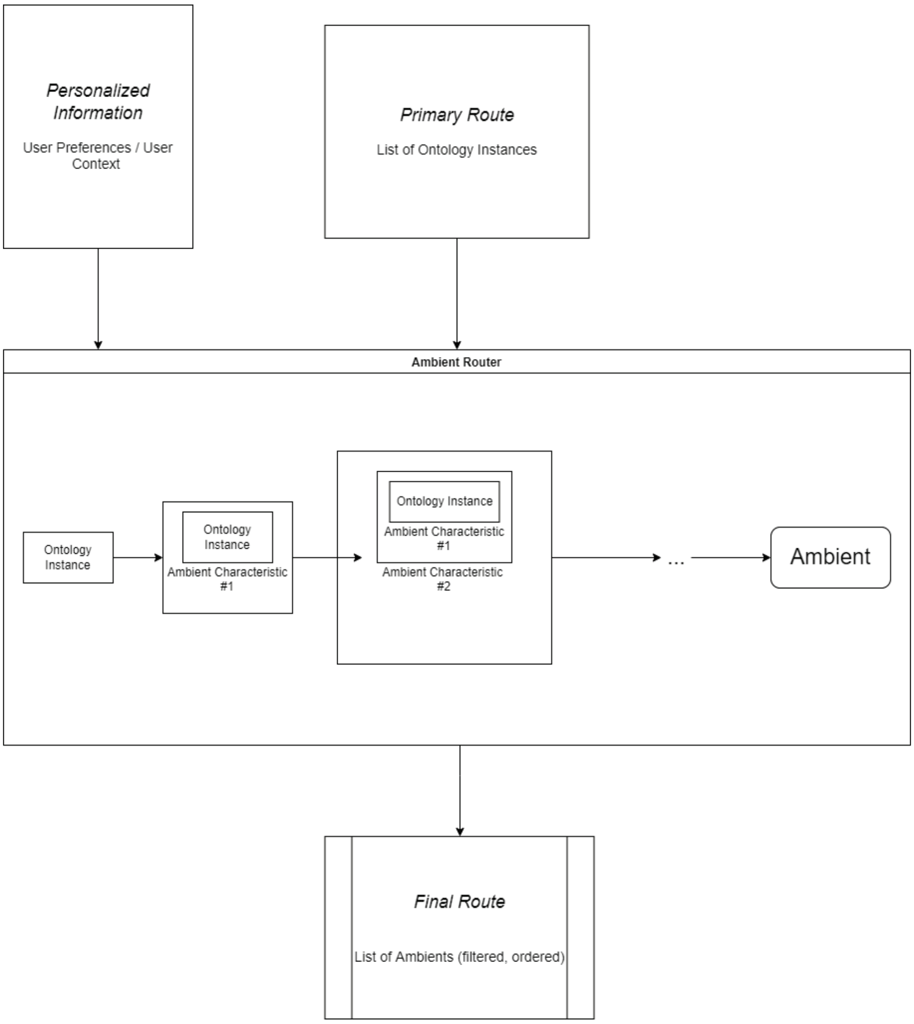

Fig. 2. Ambient-aware Route Generation Algorithm

5 Conclusions

The core logic behind the Tourist Guide is the algorithm for route generation. It started as a simple virtual tour assistant on top of an ontology based on recursive concepts which allow the user to perform both coarse-grained and fine-grained search operations which eventually resolve to a set of objects comprising the primary route. By applying shared ontology classes to the overall set of instances, the virtual tour assistant is able to suggest similar (to a selected one) objects for the tourist to visit virtually.

As a next step, the algorithm evolved into a location-aware semi-virtual router which by keeping track of both users' and ontology instances' coordinates manages to direct the user to objects located within configurable distance. As a starting point it applies either (latitude, longitude) captured by the device, or the geolocation of an object the user has already selected in the app.

The (currently) final stage of the generator is the ambient-aware implementation which extends the semi-virtual (and transitively the virtual) router. The algorithm body requires maintaining state of the ambient characteristics for the objects and keeping track of the latest tourist preferences related to those characteristics. The end result is a function transforming a raw set of ontology instances (primary route) into a refined ordered list of ambients (final route) taking into consideration various real world factors in context of personalization. In essence, the final route is a set of possible arrangements of a subset of the objects to be visited from the primary route (G. Kostadinov, (Kostadinov, Milev, Staynov, & Stoyanova-Doycheva, 2020).

References

Skripal, B., & Pyshkin, E. (2015). Automated Leisure Walk Route Generation for an Interactive Travel Planner. *Internetional Workshop on Applications in Information Technology*.

Doychev, E., Stoyanova-Doycheva, A., Stoyanov, S., & Ivanova, V. (2016). Agent-Based Support of a Virtual eLearning Space. *ICCCI 2016, Halkidiki, Greece, 28-30 September, 2016,* https://www.ieee-is.org/intelligent-systems-2016/conference-program/ .

Kostadinov, G., Milev, K., Staynov, S., & Stoyanova-Doycheva, A. (2020). Algorithm for Generating and Visualizing Routes of an Intelligent Tourist Guide. *2020 International Conference Automatics and Informatics (ICAI), 2020,* pp. 1-5, doi: 10.1109/ICAI50593.2020.9311310.

Miteva, M., Stoyanova-Doycheva, A., & Ivanovna, V. (2018). Architecture of an intelligent tourist guide. *Computer Sciences and Communications Journal, 7(1),* 39-45, https://csc.bfu.bg/index.php/CSC/article/view/205.

Received: June 05, 2022
Reviewed: July 07, 2022
Finally Accepted: July 18, 2022