A Human-centered Semantic Navigation System for Indoor Environments

Vassileios Tsetsos[1], Christos Anagnostopoulos[1], Panayotis Kikiras[2],
Tilemahos Hasiotis[1], Stathes Hadjiefthymiades[1]
[1]*Pervasive Computing Research Group, Communication Networks Laboratory,*
Dept of Informatics & Telecommunications, University of Athens, Panepistimiopolis, Athens 15784, Greece,
{b.tsetsos, bleu, thasiot, shadj}@di.uoa.gr
[2]*Electronic Sensors Laboratory, Dept of Electrical & Computer Engineering*
National Technical University of Athens, 15773 Zografou Greece,
kikirasp@ieee.org

Abstract

In this paper we discuss the very important issue of indoor location services. Location services have been in use, and studied, for a long time in mobile networks. With the proliferation of wireless networking technologies, users are mostly interested in advanced services that render the surrounding environment (i.e., the building) highly intelligent and significantly facilitate user activities (pervasive computing paradigm). Our focus is on navigation, one of the most important location services. Existing approaches for indoor navigation are driven by geometric information and neglect important aspects like the semantics of points/areas and user preferences. The derived applications are not intelligent enough to catalytically contribute to the pervasive computing vision. In this paper, a novel navigation mechanism is introduced. Such navigation scheme is enriched with user profiles and the adoption of an ontological framework. These enhancements introduce a series of technical challenges that are extensively discussed throughout the paper.

Keywords: location services, navigation ontology, indoor navigation, human factors, inclusive design

1. Introduction

During the last few years, the continuously increasing demand of individuals to be "*always connected*" and the technological advances in mobile devices and applications caused a boost in the penetration of wireless personal communications. This revolution facilitated the vision for ubiquitous services, which aid users in their every-day life activities in an intelligent and unobtrusive way, in close agreement with the provisions of the "pervasive computing paradigm" [4][7].

Another key enabler of pervasive computing apart from the ubiquitous networking infrastructure is the enrichment of the different systems with semantics (mainly through the definition of proper ontologies). Such semantically enriched system-modeling aims at developing applications with enhanced functionality and advanced reasoning capabilities. Hence, pervasive computing environments can achieve the envisaged "Ambient Intelligence" by combining domain knowledge with advanced reasoning mechanisms, allowing the deployed services to explore hidden relationships between the system entities and to provide solutions to problems that were otherwise infeasible [8][9]. Currently, semantic technologies are mainly driven by the Semantic Web initiatives and are used in a variety of application domains such as life sciences, automotive services, translation services, smart spaces and location based services (LBS).

In this paper we further investigate the aforementioned semantic LBS. More specifically, we present the design of OntoNav, an integrated, navigation system for indoor environments, which is based on a hybrid modeling of such environments (i.e., both geometric and semantic). OntoNav is purely user-centric in the sense that both the navigation paths and the guidelines that describe them are provided to the users depending on their physical and perceptual capabilities as well as their particular routing preferences. In fact, the system is mainly inspired by the widely adopted visions of Ambient Intelligence [7] and Design for All [6] (also known as Inclusive Design) and has been designed by taking into account people that have various limitations on way-finding and when moving in indoor environments. However, the system can be further extended to accommodate different virtual constraints (or preferences) of "normal" users. For example, a doctor who has no difficulty in moving around a hospital may want to plan her/his visits according to her/his scheduled tasks so as to be more productive and efficient.

The organization of the paper is as follows. In Section 2, we review similar work. In Section 3, we discuss the overall architecture and functionality of the system. In Section 4, we define the key concepts of our navigation model and their classification in an Indoor Navigation Ontology (INO). We also describe, in detail, the user modeling and some indicative classification of user categories. In Section 5, we describe the geometric algorithms that are used for the determination of all possible paths, irrespectively of user capabilities. In Section 6, we discuss the reasoning tasks involved in the path selection, most of which involve ontological reasoning. The paper concludes with a discussion on issues for further support of the navigation procedure and some possible directions for future work.

2. Related Work

As already mentioned, the primary goal of this work is the design and development of an integrated user-centric navigation system for indoor mobile environments. Till now, many outdoor pedestrian navigation systems have been proposed, which, in their majority, utilize Geographic Information Systems (GIS). The preference to the outdoor navigation systems can be merely attributed to the fact that there are better positioning infrastructures for outdoor environments [1]. Moreover, one could argue that indoor navigation is not as important as outdoor navigation, since building environments are more constrained geographically than their outdoor counterparts. Furthermore, even if one fails to discover a route towards her destination she can try again, without having spent (in general) much time and effort. These arguments generally hold true but not for cases of disabled people, elderly people or in very large environments such as hospitals where both the working staff and patients need to find and use the "best" traversable (accessible) navigation path. The term "best" in the context of this work may occasionally refer to the shortest path, the easiest path (e.g., without stairs), the path that passes from many points of interest, the most popular from a set of possible paths, etc. In addition, the presentation of the selected paths is very important, as stated in the relevant bibliography [18] and should also be performed in a way inclusive of the user's special characteristics.

In general, research on indoor navigation has not progressed significantly and is mainly motivated by robot navigation [2][3]. One of the first non-robotic systems was the Cyber Guide system [16], which provided both indoor and outdoor navigation. It was designed as a tourist assistant based on the knowledge of their position and orientation.

A similar system is the indoor component of the MARS system [13], which provides to visitors, students, and faculty staff information regarding the buildings of the Columbia university campus. MARS uses inference mechanisms and path planning to guide users towards their targets.

However, the semantic modeling of navigation systems is still in its infancy. A quite interesting approach for spatial modeling with emphasis in navigation services is presented in [5]. We have borrowed the concept of exits from this work, since OntoNav navigates the users inside floors and buildings but it does not provide navigation instructions within rooms. Other semantically enriched navigation systems are presented in [14][20]. PoLoS [21] is an enhanced LBS platform for indoor/outdoor navigation, which aggregates both GIS information and user's location for human navigational presentation purposes. An approach that is based on ontology and the way humans navigate, select and mentally represent routes is the Navio project, described in [19].

To summarize, most systems, although they take into account the geographical coordinates of the navigation destination, they do not do the same with the users' physical and perceptual capabilities as well as their routing preferences; in particular, they are using weights in order to compute the navigation path in the geographical – topological layer, based on the specific characteristics of the available positioning technology. On the other hand, OntoNav is a hybrid navigation system, since it transforms a problem of geographic path determination to a problem of both semantic and geographic path selection by utilizing ontologies and rules defined based on the physical and perceptual/cognitive characteristics of the users and on the semantic meta-information of the path elements (passages, corridors, etc.).

3. Architecture Overview

OntoNav consists of the following building blocks (see Figure 1):

Navigation Service (NAV): It is the main interface between the user and the system. It receives users' requests for navigation and responds with the requested path (if any), in a form tailored to each user's special characteristics (perceptual and physical). The Navigation Service aggregates the Geometric Path Computation Service (GEO) and the Semantic Path Selection Service (SEM) and can also be interfaced, depending on the deployment configuration, with other systems such as user authentication or directory services and ontology repositories.

Geometric Path Computation Service (GEO): This service is responsible for the computation of all the geometrical paths from a user's current location to a specified destination (Point of Interest, POI). Therefore, it utilizes a spatial database, where the building's ground plans (blueprints) are stored. For the computation of the navigation paths the system executes a variant of a traditional graph-traversal algorithm on a graph representation of the stored geometry. A graph creation algorithm, whose description is not in the scope of this paper, produces this graph. The paths that are computed by the searching algorithm are sent to the SEM Service for further filtering based on the user characteristics and routing preferences. The GEO Service is depicted in Figure 2 and is described in more detail in Section 5.

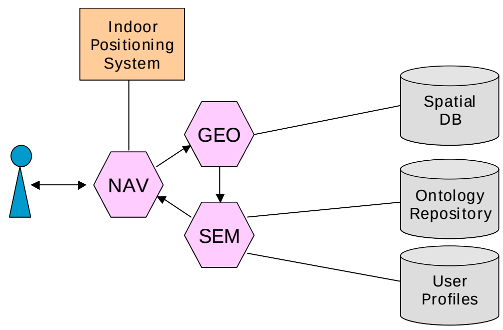

Figure 1. Overview of the OntoNav architecture

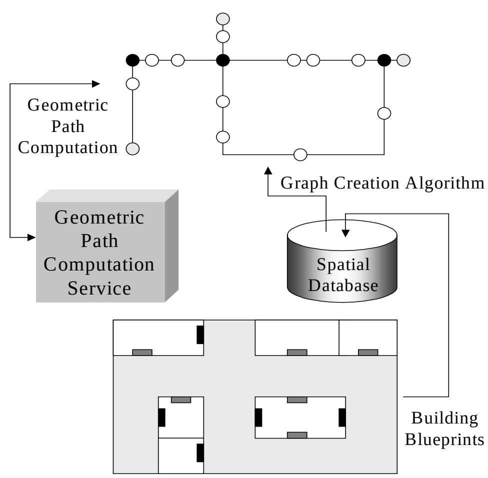

Figure 2. The GEO Service functionality

Semantic Path Selection Service (SEM): This service provides the main functionality of our system and is responsible for the selection of the *best traversable* navigation path among those established by the GEO service. This path is one that matches all the capabilities and preferences of the user and it is, thus, selected based on predefined rules and on a user profile registry, which contains these user capabilities/preferences (see also Section 4.2). This task is achieved with the aid of a navigation ontology (see Section 4.1), which enables the required reasoning:

- path selection according to the physical capabilities and routing preferences of the user, and
- selection of the proper navigation guidelines (anchors), according to the physical and perceptual capabilities of the user.

4. OntoNav Semantic Model

4.1 Indoor Navigation Ontology (INO)

The proposed navigation scheme is largely based on semantic descriptions of the constituent elements of navigation paths, which, in turn, enable reasoning functionality. Thus, we developed an Indoor Navigation Ontology (INO), which supports both the path searching and the presentation tasks of a navigation system. The basic taxonomy of this ontology is depicted in Figure 3. INO apart from concepts includes also roles (binary relationships between concepts), axioms, and constraints on these roles. Providing the full INO specification is out of the scope of this paper. However, we should mention that some of the reasoning tasks described in Sections 6 and 7 are performed by utilizing the transitive and symmetric properties of the INO roles.

A human-readable documentation of this ontology follows:

User: this concept represents the users of the navigation service, which have specific physical and perceptual capabilities/constraints. A (incomplete) classification of users is: blind, having mobility difficulties, elderly people and "normal" users. Additionally, a user could be classified according to her navigational status (e.g., she may have deviated from a path or be probably lost).

Point_of_Interest (POI): any physical or virtual location or object, which may be of interest to a user and may serve as a navigation destination (e.g., room, printer).

Passage: any spatial element that is part of a path and has specific accessibility properties. We can categorize passages to *horizontal* (connecting corridors in the same floor) and *vertical* (connecting corridors in different floors). The main types of vertical passages are elevators and stairs. The main types of horizontal passages are ramps for wheelchairs and doors. At this point, we should distinguish the term "door" from the term "exit", described below. An exit is always attached to an indoor region (e.g., room), while doors connect corridors and/or passages and are always perpendicular to the corridors.

Navigational_Point: special types of points that connect more than two corridors or enforce change of direction to users or indicate the end of corridors (e.g., representing a waiting public area - not room - leading to different corridors etc.).

Exit: an exit or entrance of an indoor region. Such region may be the whole building, a room, an elevator etc. This

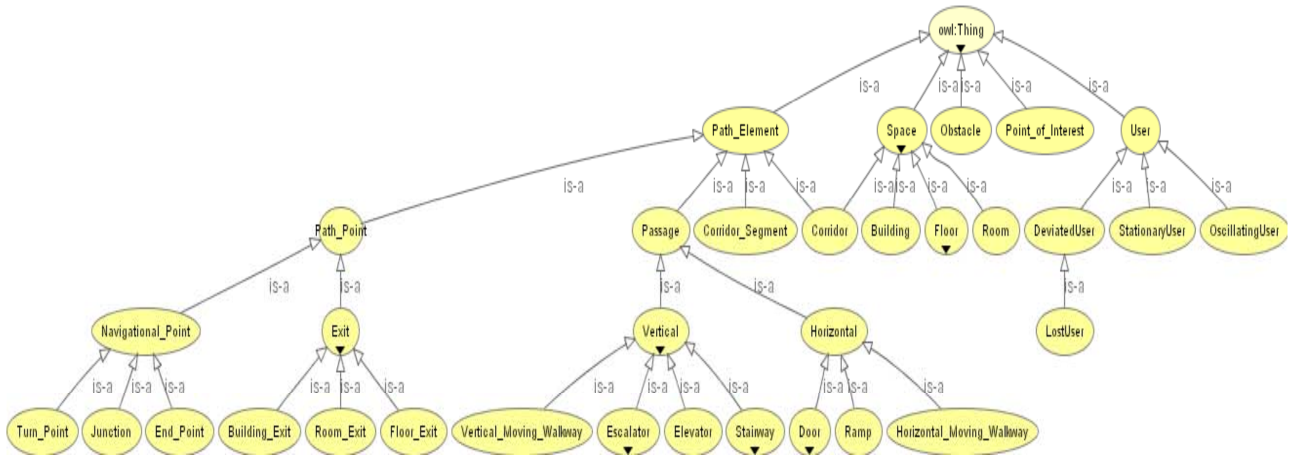

Figure 3. The Indoor Navigation Ontology

concept is borrowed from [5] and each indoor region is reduced to a set of exits.

Obstacle: anything that prevents the movement of the user. That definition includes a) physical objects whose dimensions (width and height) block a corridor or passage, b) certain properties of exits or passages (e.g., closed door, non operating elevator), and c) other non-permanent conditions which prevent the passage of the user (e.g., security policies, a deluge of people in a space that makes difficult the passage of blind people, etc). The latter type of obstacles is very important as it enables the definition of dynamic and non-physical obstacles.

Corridor_Segment: The concept of a corridor segment is a construct devised to facilitate modeling and is derived by the geometric graph of paths (see Section 5). A corridor segment belongs to only one corridor and connects exits and/or passages.

Corridor: a corridor is comprised of corridor segments, which connect two navigational points or a vertical passage with a navigational point. A corridor may also contain POIs and obstacles.

Anchor: any passage, exit, navigational point or POI included in a path that can aid the presentation of the navigation plan. Anchors cannot be movable objects. Examples of anchors are: junctions, doors, stairs and ramps. Hence, anchors are mainly structural elements of buildings. However, non-structural POIs could also be used as anchors, e.g. a coffee machine. Although we briefly describe the process of selecting suitable anchors in a later section, we do not give more details about anchor-related issues in the present paper.

Path: a sequence of interleaved path elements (e.g., corridors, path points and passages), which is capable of getting a user from its current location to a destination location. A *walkable path* is a special path, which can be used by any "normal" user. Apparently the set of walkable paths in an indoor environment is the superset of all other path-sets, which are accessible by specific user classes. The geometric model (graph) of our system represents this superset (walkable paths). A path usually contains several POIs, anchors and obstacles. The subset of them, which will be used for the final user navigation, is defined depending on the user perceptual capabilities.

The aforementioned set of concepts cannot provide all the desired model expressiveness by itself. For that purpose we had to import elements of other spatial ontologies, which define spatial concepts and topological relations between them (e.g., we need the concepts *room*, *floor* and *building* in order to completely locate the POIs and users). In the current version of the INO we have defined the concept Space along with its sub-concepts. Ideally, these should be imported from a spatial ontology. In fact we are in the course of designing such an ontology, which will enable the description of generic indoor spatial environments and reasoning functionality on their instances.

4.2 User Modeling

The main objective of our system is to provide a user-centric navigation paradigm for indoor environments based on the user's physical and perceptual capabilities or limitations. In order to achieve this objective, the system is aware of the aforementioned user capabilities, which are described by a User Profile (UP). A UP is defined as a collection of classified attributes, most of which represent specific user capabilities/limitations. Such collection of attributes may be denoted as the set:

$UP = \cup_i\{<attributeClass, attributeName, attributeValue>_i\}$

for i=1..n different classes of grouped attributes. For the purposes of OntoNav we define three different and disjoint classes of attributes:

- The class of physical capabilities (i.e., attributes related to user's physical capabilities),

- The class of perceptual capabilities (i.e., attributes related to user's understanding of navigation guidelines),
- The class of preferences (i.e., attributes related to various user preferences regarding the path selection process)

Each UP instance is uniquely associated with a user. It is important to mention that it is this instance that is used by the reasoning tasks described in Section 6. The first time a user invokes the system's interface, she creates her profile by providing all the indispensable information that can describe her physical and cognitive condition. Moreover, the UP is completely dynamic; the user may change her profile whenever necessary.

OntoNav uses the aforementioned user profiles in conjunction with various user-independent rules in order to infer which of the walkable paths are suitable for a given user and how the navigation guidelines should be presented. These two selection processes are implemented with the aid of three kinds of navigation rules - the physical navigation rules, the perceptual navigation rules, and the routing preferences - that correspond to the attribute classes of the UP set. The physical navigation rules are used for the selection of the paths that match the user's physical capabilities. The user, according to her UP profile, applies these rules to the set of all possible walkable paths in order to exclude those paths that are not traversable. The system determines that a path is traversable by a user if and only if it contains passages that can be used by her, does not contain any obstacles and matches her preferences. Some examples of the physical navigation rules are as follows (for size limitation reasons we will not refer to navigation preference rules):

- If *path p* contains an obstacle o then *path p* is excluded.
- If *user x* cannot walk and *path p* contains a *vertical passage v* of type stairs then *path p* is excluded.
- If *user u* can walk and carries an *object o* of a given width and the *path p* contains a *vertical passage v* of type elevator whose width is less than the width of *object o* then *path p* is excluded.

The perceptual navigation rules are rules that are used for the selection of the best-suited anchors across a traversable path for the best presentation of the navigation guidelines. The anchors are selected based on both the user's perceptual and physical capabilities. Some examples of such rules follow:

- If *user u* is an illiterate person and the *path p* contains an *element x* for which the system has visual/graphical descriptions then add *elements x* to the set of anchors.
- If *user u* is blind and *path p* contains an *element x* for which the system has auditory descriptions then add *element x* to the set of anchors.

5. OntoNav GEO Service

The determination of the paths between two endpoints has been thoroughly studied in the literature [15]. Most related works model the navigation problem as a graph-searching problem. We also adopt this approach and we use a graph for representing the different path elements of indoor environments. However, in such environments there is an additional issue: the existence of floors. We present here a graph model, which a) accumulates all the floor sub-graphs into one planar graph of the whole building and b) performs a clustering algorithm for more efficient path discovery. The passages connecting two or more floors (i.e., stairs, elevators, ramps) are represented as single nodes in this graph.

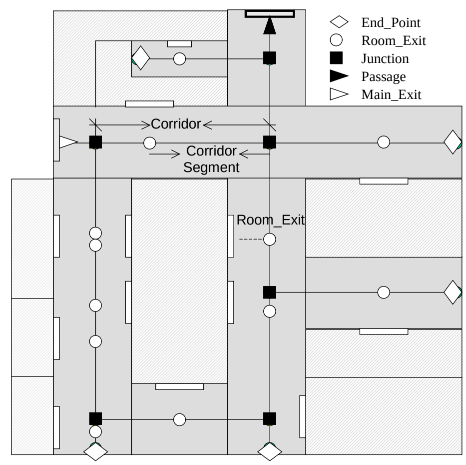

Figure 4. The path graph overlaid to the building's ground plan

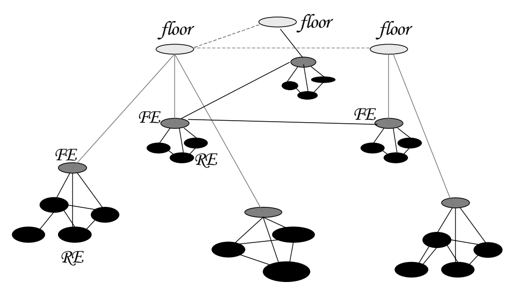

Figure 5. The Hierarchical Clustering Graph

Specifically, let us define a, possibly, not fully connected, graph G_j for the jth floor, with a set of vertices V_j and a set of edges E_j. A set of such graphs comprises the accumulated planar graph G, representing the path information of a building:

G= $\otimes_i$ (G_i) for i=1..#floors and the operator $\otimes$ acts as a special concatenation of the floor sub-graphs. Thus,

$G = (V, E) = (\bigcup_{j=1}^{n} V_j, \bigcup_{j=1}^{n} E_j)$, where n is the number of floors.

The set V_j is defined as follows: $V_j=\{N \cup X\}$, where N is the set of navigational points and X is the set of exits on the jth floor. According to the INO taxonomy (see Figure 3), we can further specialize the elements of X by defining the subsets:

- Room_Exits (RE): Such vertices are created by the vertical projection of each room exit to its adjacent corridor (see Figure 4).
- Floor_Exits (FE): Such vertices denote the exits from one floor to another.
- Main_Floor_Exits (MFE): A set with a unique element: the main exit of each floor. A FE is a MFE if it satisfies some heuristic criteria, i.e. leads to the most commonly used passage, connects the greatest number of floors. In the ground floor the MFE is the main entrance of the building.

Furthermore, we can categorize the navigational points in N to the sets:

- End_Points: These vertices denote the end of a corridor.
- Junctions: The set of locations, which connect three or more corridors.
- Turn_Points: The vertices of this set just change the direction/orientation of a path.

The set E_j defines the edges (i.e., corridors) that connect vertices from the corresponding V_j set. The *path* graph G can be created by an appropriate algorithm, which takes as input the geometry of the building's floor plans. This geometry can be built and stored in a spatial database (e.g., PostGIS [10]). The graph creation algorithm first creates a skeleton of the corridors (the edges of the graph) and then creates the vertices on this skeleton (by projecting the various spatial elements, such as exits, to the skeleton's line segments). During the graph creation we also calculate the lengths of the edges and can assign a name to every vertex and corridor. These names should be in accordance with the names of the instances of INO in order to enable further semantic reasoning on the paths.

Subsequently to the creation of the graph, we can execute a graph-traversal algorithm in order to find the walkable paths between two given vertices. The output of this algorithm is a set of one-dimensional arrays containing all the graph elements (edges and vertices) traversed by each walkable path. As the modeled buildings become bigger, their path graphs become larger and, also, the sets of walkable paths increase non-linearly. To handle such computational complexity we perform clustering and create a Hierarchical Clustering Graph [11][12]. This is a tree-like hierarchical graph with each cluster representing a floor graph (see Figure 5). The path computation algorithm (see Figure 6) first searches among the floors (the upper side of the hierarchy) and identifies which floors should be involved in the navigation. Then the algorithm is applied to the specific vertices of the graph of each selected floor (the lower level of the hierarchy). The various path segments computed between the floors and between the vertices of each floor are concatenated to form the final set of walkable paths.

```
findAllWalkablePaths (u,v)
//inputs: source location u, destination location v
//output: set of walkable paths T
T=∅
Begin
  f_u = floor(u) , f_v=floor(v)
  if (f_u=f_v) then  S = interRouting(u,v) , T = T ∪S
  else
   S_initFloor = interRouting(u,floor_exit(f_u))
   T = T ∪ S_initFloor
   S_j=∅
    for each floor j do
       Begin
          S_j = S_j  ∪ interRouting(main_floor_exit(f_j),floor_exit(f_j))
       End
       T = T ∪S_j
    endElse
    S_termFloor = interRouting(floor_exit(f_v),v)
    T = T ∪ S_termFloor
  endElse
  Return T
End
interRouting(u,v)
// u:source, v:destination
S = ∅
Begin
      S = SearchGraph(u, v)
  Return S
End

floor_exit(f) : returns the FE that is closer to f
main_floor_exit (f): returns the MFE  of floor f
```

Figure 6. The Geometric Path Computation Algorithm

6. Main reasoning tasks of OntoNav

In the previous sections we have described all the necessary modeling elements for an indoor navigation system. In this section we are discussing how these elements can be combined into a reasoning process whose final outcome will be the selection of the best-suited navigation plan for the user that requested the Navigation Service. As already mentioned, this process comprises several reasoning and computational tasks. These tasks, described in the order of their execution, are:

Task A: determination of the navigation's starting and ending points, S' and E' respectively.

We assume that S is the current location of the user, as determined by a symbolic indoor positioning system [17], and E the respective location of the requested POI. These locations are in general not represented as nodes in the graph. Thus, we need to match these locations with existing graph nodes (we should remind that the graph nodes can be either exits or navigational points of the considered environment). If we consider the problem more, we see that S and E can be rooms, corridors or vertical passages. Moreover, all these types of locations may have more than one path points or passages directly connected to them. In the first case (i.e., S and/or E are rooms) S' and/or E' are actually sets of exits. In the second case (i.e., S and/or E are corridors) S' and E' are sets of exits and navigational points. Finally, in the last case (S and/or E are passages), we can match the points S and/or E to their nearest exits. Thus, we have transformed our initial point-to-point navigation problem to a set-to-set navigation problem, between all the combinations of elements of sets S' and E'. These elements/nodes may not be the actual user or POI locations but are, generally, good approximations of them (it depends on the positioning infrastructure). Moreover, this approach enables the addition/removal of POIs without affecting the path graph topology.

Task B: discovery of all possible walkable paths leading the user from its current location S' to the target Point of Interest (location E').

This process determines (with a variant of traditional graph traversal algorithm) all the paths that a user can traverse for each combination of the S' and E' elements. The output of this iterative computational task is a (possibly empty) set of walkable paths. For each walkable path its length is computed, too.

Task C: semantic-driven selection of the Best Traversable Path (BTP).

This reasoning task is actually a two-phase procedure. During the first phase, reasoning is performed on the instances of the navigation ontology using the *physical navigation rules* and the *routing preferences*. In particular, such task uses these user-specific rules for the exclusion of the paths that are not traversable. A path is *traversable* if it supports the user's physical capabilities. For example, the paths that contain stairs are excluded if the user uses a wheelchair. Thus, the first phase ensures that only the traversable paths are selected from the superset of walkable paths. In the second phase, which selects the *best* path, additional selection criteria are applied on this set of traversable paths. Such criteria are based on user's preferences and may be that the shortest traversable path should be selected, or alternatively the path that can serve the most user tasks described in user's calendar. The output of this latter phase is a single path from the set of the traversable paths. While the first phase is default and predefined by our system, the second phase allows the adaptation of the path selection process to the actual *quality metrics* of the user. For example, the quality metric for a certain user can be the path length, while for another user, the scheduled tasks she can accomplish while traversing a navigation path.

```
Algorithm: selectBTP

Input: navigation space (INO instances, geometric graph), user
profiles repository, user ID, start and end navigation points
Output: Best Traversable Path (BTP)

UProf = getProfile(userID);
int numOfPaths = 0;
walkablePaths = findAllWalkablePaths(userLocation,
                                     POILocation);
for (i=0; i<walkablePaths.length; i++){
   for (j=0; j<walkablePaths[i].length; j++){
      pathElement = walkablePaths[i][j];
      if (pathElement.isTraversable(UProf.physicalRules)
         if (j==walkablePaths[i].length-1){
            //if all passages were traversable then
            //add the path to the selected paths matrix
            selectedPaths[numOfPaths]=i;
            numOfPaths++;
         }
      else continue;
   }
}
if (numOfPaths == 0)
  return null;
else if (numOfPaths == 1)
   BTP=selectedPaths[0];
else BTP=shortestPath(selectedPaths);
BTPAnchors=BTP.findBestAnchors(UProf.perceptualRules,
                               UProf.physicalRules);
```

Figure 7. Complete algorithm for BTP selection

Task D: selection of the anchors across the best traversable path.

Anchors are the elements of the path that are best suited for the presentation of the navigation guidelines. During this process, all the anchors of the selected path are detected and are matched against the *perceptual navigation rules* and the *physical navigation rules*. These rules define not only which anchors should be used, but also how many anchors should be used. As an example, assume that the navigation service requestor is a blind man. In that case, we should choose *many* anchors all of which having auditory descriptions. This reasoning task outputs a *sequence of navigation anchors* that are used, in their turn, as input to the navigation presentation subsystem. The specific details of this latter subsystem are out of the scope of this paper, since we focus on path modeling and path discovery/selection issues.

The complete algorithm (in Java-like syntax), for the selection of the Best Traversable Path (BTP) for a specific user and of the corresponding anchors for the navigation guidelines, is depicted in Figure 7 (the method names written in bold italics represent the different reasoning tasks described earlier in tasks A-D. The exact implementation of these methods involves the INO instances, the user profile and the geometric graph).

6.1 Optimizing the Path Computation Procedure

The major deficiency of the presented system is probably the high computational cost of the greedy graph traversal algorithm. We remind that the GEO service computes all the walkable paths, in contrast to other approaches that use shortest path algorithms (e.g., A-star). Such greedy computation of paths is necessary as the shortest paths may not be accessible by all users. Another inevitable performance handicap is the execution of the graph traversal algorithm not on a point-to-point but on a set-to-set basis (see Section 5). The reason for that "complexity explosion" is, again, the fact that for special types of users only a few traversable paths may exist and, in general, they may not be the shortest ones. While this latter computational complexity cannot be eliminated by nature, there are some approaches for the optimization of the first one (i.e., geometric path computation).

One way to overcome the first deficiency is to abandon the graph creation and the geometric path computation and merge the tasks B and C with the aid of rules applied directly on the ontology instances. Such algorithm would work only with INO instances and prune the non-traversable paths according to the UP rules. Thus, when the algorithm terminates, we will have computed all *traversable* (and *not walkable* paths). This method can, among others, exploit the *leadsTo* transitive role of the ontology and the capability of ontology reasoners to compute transitive closures of transitive roles in order to decide if a path element (exit, passage, etc.) can reach the destination point. The pruning algorithm will filter out any path elements that do not match the user capabilities. The overall algorithm can be regarded as a variant of a breadth-first search algorithm that can pass more than once from each graph vertex. As the searching proceeds, the traversed edges and vertices are pruned according to the UP rules.

An alternative method for the computation of the traversable paths, without previously computing the walkable paths, is by implementing a "semantic sieve" for the geometric graph. Such "sieve" would first receive as input all vertices of the graph and apply the UP rules to their corresponding INO instances (INO instances are created simultaneously with the graph and have the same annotations as the graph elements to enable lexicographical matching). Then the edges of the resulting (smaller) graph are, also, filtered. Eventually, we get a reduced graph with traversable elements and, thus, we can apply shortest path algorithms on it in order to determine the Best Traversable Path. This approach is considered more efficient than the previous ones, however we are currently comparing those approaches in order to quantify their performance differences.

7. User Navigation Support

The navigation systems, as described so far, compute static routes between an origin and a destination endpoint. However, some categories of users may need more "strict" surveillance and guidance due to their possible special characteristics. For example, a child or a blind man may easily lose their direction and get out of the planned navigation path. A static navigation service would not be able to detect such route deviations and redirect the users to a valid path element (e.g., corridor or passage). Therefore, the overall system should include also a navigation-aiding module (NAM), which would be able to detect deviations and help users reschedule the initially planned path. The main characteristics of such a module, which are also dictated by the pervasive computing paradigm [4], are:

- The module should aid only those users who really need support and only when this support is really needed. In other words, the module should be as unobtrusive and distraction-free as possible.
- The module should be effective. The users should be redirected to the already planned valid path with minimum effort. One could claim that instead of redirecting them to the initial path, a new and more suitable path could be computed given their new location. Such computation would not be much efficient especially when the user's capabilities are too restrictive (i.e., the new path would likely lead to the initial path).

In this section, we present some details about how such navigation aiding can be implemented on a *pure semantic*, and not geometric, level. More specifically, due to size limitations, we focus on the first characteristic described and we do not deal with the actual redirection procedure. Thus, we discuss some elements of the module's decision-making process.

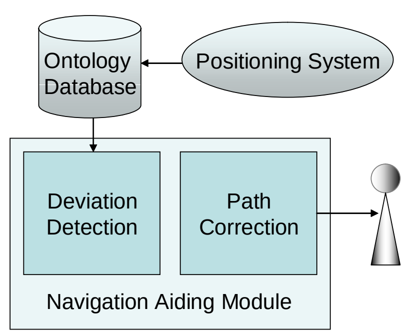

Figure 8. The Navigation Aiding Module

The general architecture of the NAM is depicted in Figure 8. We assume that the user position is periodically asserted (in symbolic form) to an Ontology Database, which stores the instances (individuals) of the INO ontology. Whenever the positioning system detects that the user has moved from her last location, it asserts this fact to the Ontology Database and deletes any deprecated facts regarding her last known location.

7.1. Important Decisions and Degree of Freedom

In order for the navigation aiding service to be as unobtrusive as possible, it should be activated only for users that may possibly lose their route or on user demand. For example, a “normal” user that deviates from the path at her will (e.g., because she met a friend of her) would perceive an alert from the service as obtrusive. On the other hand, a blind user would probably need some aiding during its navigation. Apparently, the perfect place to store information about when and how the system should aid the users is the User Profile. For that purpose we introduce a set of predefined aiding rules to the UP and categorize them based on the Degree of Freedom (DoF) they give to users. The DoF defines the following decisions:

(a) which users should the aiding module be activated for
(b) how is the deviation from a path defined and computed
(c) when should the module be activated and give feedback to a user
(d) how frequently should the positioning system retrieve a user’s location
(e) how should the system react to possible path deviations (i.e., compute a new path or redirect the user to the already computed path?)

Most of those decisions affect three metrics: the user experience, the efficiency of the aiding procedure and the system performance. We are in the course of investigating which decisions should be assigned to each DoF set so as to optimize the overall system behavior.

8. Conclusions and future work

In this paper we have described several information modeling and processing issues regarding a human-centered navigation service. This service is mainly targeted to people with navigational limitations and it adheres to the vision of intelligent location based services for ubiquitous computing environments. The overall service is decomposed to several other services (geometric path computation, semantic filtering of possible paths, reactive navigation aiding service) and utilizes a user profile structure. The main goal of this work is to creatively integrate semantic knowledge engineering technologies with traditional location-based services. In our opinion, such integration is a key enabler of pervasive services, which focus on the user experience.

OntoNav is currently in its development phase. When the implementation is complete, we aim to evaluate its effectiveness and performance through its real-world deployment in our campus facilities. Moreover, we intend to improve the path computation procedure with the aid of path caching and path prediction techniques.

9. References

[1] P. Kikiras and D. Drakoulis, “An Integrated Approach for the Estimation of Mobile subscriber Geolocation”, *Wireless Personal Communications* 30: 217–231, 2004.

[2] M. Golfarelli, D. Maio, S. Rizzi, “Correction of Dead-Reckoning Errors in Map Building for Mobile Robots”,*IEEE Trans. on Robotics and Automation,* 2001, Vol 17, No 1, pp. 37-47.

[3] C. Hofner, G. Schmidt, “Path planning and guidance techniques for an autonomous mobile cleaning robot”, *Intelligent Robots and Systems '94*. 'Advanced Robotic Systems and the Real World', Proc. of the IEEE/RSJ/GI International Conference. Sept. 1994, Vol. 1,

[4] M. Satyanarayanan, “Pervasive computing: vision and challenges”, *IEEE Personal Communications*, Aug. 2001, Vol. 8, No. 4.

[5] Haibo Hu, Dik-Lun Lee, “Semantic Location Modeling for Location Navigation in Mobile Environment”, proc. *IEEE MDM’04*. 2004.

[6] European Institute for Design and Disability, [http://www.design-for-all.org/].

[7] ISTAG, Scenarios for Ambient Intelligence in 2010; Final Report, Feb 2001, EC 2001: [http://www.cordis.lu/ist/istag.htm].

[8] H. Chen and T.Finin, “An ontology for Context-Aware Pervasive Computing Environments”, in Working Notes of the *IJCAI-2003 Workshop on Ontology and Distributed Systems*, Acapulco, Mexico.

[9] A. Ranganathan, R.E. McGrath, R.H. Campbell, M.D. Mickunas, *Ontologies in a Pervasive Computing Environment,* in Working Notes of the *IJCAI-2003 Workshop on Ontology and Distributed Systems*, Acapulco, Mexico.

[10] PostGIS, [http://postgis.refractions.net].

[11] Y.Shiloach, “Arrangements of Planar Graphs on the Planar Lattices”, PhD Thesis, Weizmann Institute of Science, Rehovot, Israel, 1976.

[12] Mirkin, B., *Mathematical Classification and Clustering*, Kluwer Academic Publishers, 1996.

[13] Tobias Höllerer, Steven Feiner, Tachio Terauchi, Gus Rashid, Drexel Hallaway, “Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System”, *Computer & Graphics* 23(6), Elsevier Science Limited, 1999, pp 779-785,

[14] Philippe Hoppenot, Gilbert Pradel, Catalin Caleanu, Nicolas Perrin, Vincent Somm, “Towards a symbolic representation of an indoor environment”, *IEEE CESA* 2003.

[15] Brian Salomon Maxim Garber Ming C. Lin Dinesh Manocha, "Interactive Navigation in Complex Environments Using Path Planning", *Symposium on Interactive 3D Graphics*, 2003, pp. 41-50.

[16] Abowd, G.D. et al., "Cyberguide: A mobile context-aware tour guide" . *Baltzer/ACM Wireless Networks*, 1997, 3 (5), pp. 421-433.

[17] Nibble Location System, [http://mmsl.cs.ucla.edu/nibble]

[18] Cristoph Stahl, Antonio Krüger,Jörg Baus, "Location Dependent Generation of Sketches for Mobile Indoor Route Descriptions", *ECAI 2002*.

[19] G. Gartner, A. Frank, G. Retscher, "Pedestrian Navigation System in Mixed Indoor/Outdoor Environment- The NAVIO Project", *9th International Symposium on Planning & IT*, Vienna, Austria, 2004.

[20] Roy Want, Andy Hopper, Veronica Falcao, Jonathan Gibbons, "The Active Badge Location System", *ACM Transactions on Information Systems*, Vol. 10, No. 1, , January 1992, Pages 91-102.

[21] M.Spanoudakis, A.Batistakis, I.Priggouris, A.Ioannidis, S.Hadjiefthymiades, and L.Merakos, "Extensible platform for Location Based Services deployment & provisioning", book chapter in Wireless Information Highways, IRM Press, ISBN: 1-59140-568-8, 2004.