Indoor Navigation for Motion Disabled Persons in Medical Facilities

Rita Fleiner[1], Barnabás Szász[2], Gabriella Simon-Nagy[1], András Micsik[3]

[1]Department of Applied Informatics, John von Neumann Faculty of Informatics, Óbuda University, Bécsi út 96/b, 1034 Budapest, Hungary
{fleiner.rita; nagy.gabriella}@nik.uni-obuda.hu

[2]Faculty of Informatics, University of Debrecen, Kassai út 26, 4028 Debrecen, Hungary

[3]Department of Distributed Systems, Institute for Computer Science and Control, Hungarian Academy of Sciences, Lágymányosi u. 11, 1111 Budapest, Hungary, andras.micsik@sztaki.mta.hu

Abstract: A data model is presented in the form of ontology which includes the indoor location description of hospitals, the indoor navigation features and the accessibility attributes for people with motion disabilities. The possible use of the ontology is demonstrated by outlining some RDF data excerpt, OWL definitions and SPARQL queries for the navigation features of future applications.

Keywords: Linked Open Data; ontology; indoor navigation; medical facility; accessibility

1 Introduction

Linked Open Data (LOD) [1] is a well-known method of publishing and interlinking open structured data on the Web so that computers can read it automatically. This method enables data from different sources to be connected and queried; connections can be created by utilizing links contained in datasets that refer to other datasets. The standard data model for Linked Open Data is Resource Description Framework (RDF). In RDF, data is structured as triples in the form of subject, predicate and object, which is called a statement. SPARQL is an RDF query language, designed to retrieve and manipulate data stored in RDF format.

In our previous works we developed general methods for indoor wayfinding. In this paper we address the special case of navigating inside the building of a medical facility. This means that the navigation task is defined between points of interest (POIs), rooms or departments of a hospital building as starting and end points. We also take notice of the accessibility features of building parts to support navigation for the motion disabled visitors and patients. To accomplish this, we allow description of several features of corridors and rooms that are significant in terms of accessibility, such as distance, number of stairs, and the presence or absence of barriers and assistance features in the building.

Outdoor navigation is widely available nowadays, and helps people to find a place while driving or walking or using public transport. This type of navigation is usually based on a map and coordinates provided by GPS. Inside buildings, however, a navigation system has to cope with more complex routes and lacks the use of GPS signals. Most of the current solutions require special and expensive hardware for indoor positioning. Instead of relying on an indoor coordinate system, our approach follows the natural way, how people give route advice to each other, telling which landmarks to pass in sequence to reach the destination. Requisites for our navigation method are kept as inexpensive and simple as possible. The solution should scale up to large buildings, like hospitals. In our work we exploit Linked Data and SPARQL for building flexible APIs providing location and routing information.

The complexities of the requirements demanded a data model developed as a formal ontology, which also helps to apply the Linked Data principles to the published data. During the ontology development our aim was to design a 4-star vocabulary described in the guidelines [2].

The aim of the paper is to provide a data model in the form of two ontologies: iLOC [3] and hLOC [4]. The iLOC ontology describes the indoor navigation features and the accessibility attributes for people with motion disabilities, the hLOC contains the indoor location description of hospitals, including supportive services typically found in medical facilities.

The rest of the paper is organized as follows. In Section 2, we describe research fields that are related to the topic and in Section 3, usage examples are presented that the suggested model should satisfy. In Section 4, the ontology achieving accessible indoor navigation is explained. In Section 5, the ontology of indoor location description of hospitals is introduced. Section 6, shows examples for SPARQL queries and OWL definitions using the developed ontologies.

2 Related Work

Linked datasets rely on vocabularies, schemas or ontologies. There were several attempts to provide indoor navigation ontologies. Worboys [5] provided a general overview of the state of the art, and defines a top level taxonomy to classify indoor models into semantic and spatial categories. Semantic indoor space models represent entity types, their properties and relationships. Topological models are concerned with the connectivity within a space. Geometrical models add quantification of distance and finally hybrid or multilayered models provide combined features of all the above.

OntoNav [6] is a semantic indoor navigation system and an ontological framework of handling routing requests. OntoNav navigates the users inside floors and buildings, but it does not provide navigation instructions within rooms, while in case of a large hall with several entrances it is useful to have routes inside the hall as well. The underlying INO ontology is unavailable, furthermore, it has most of the required implicit knowledge about selecting best paths and avoiding obstacles built into the path-finding algorithms, not into the ontology.

ONALIN [7] provides routing for individuals with various needs and preferences; it takes the ADA (American Disability Act) standards, among other requirements, into consideration. Buildings are modeled as hallway networks, and feasible routes can be identified for users having specific constraints. ONALIN uses pre-computed variants of hallway networks for each possible set of disabilities, and then runs path search algorithms on the selected hallway network.

Scholz and Schabus [8] created an indoor navigation ontology for the movement of production assets in a production environment, to support autonomous navigation in the indoor space. Production assets pass through steps of a workflow, and for each step the suitable equipment is found and the best route to the equipment is calculated. Details of route calculation are not given, but it is done with a tailor-made algorithm considering asset properties such as size or special handling instructions.

Geodint [9] uses standard shortest path algorithm in a derived graph model for navigation. None of the above ontologies is accessible at the moment of writing this paper. However, some parts of the conceptual semantic model were reused from these earlier work inspiring the hierarchy of classes. The main difference in our approach compared to the mentioned related work is the use of standard (or de facto standard) query languages and reasoning for the computation of routes. In this way we can keep the computation generic and use the model (the ontology) to describe all necessary knowledge for various types of route calculations. Thus, the need to design and implement a specialized route search algorithm for each domain and scenario can be avoided.

SNOMED CT [10] is one of the most comprehensive medical ontologies, consisting of more than 316.000 classes organized in a hierarchical structure. It is maintained and regularly updated by IHTSDO[1], the latest version was released in

[1] International Health Terminology Standards Development Organisation, http://www.ihtsdo.org/

September 2015. SNOMED CT contains terminology for the human body, medical procedures, pharmaceutical products, and also the logical and physical organization of healthcare facilities. The structured collection of hospital departments and room types has proven to be very useful in the process of constructing indoor navigation for medical buildings. SNOMED CT does not exist in RDF format, therefore we could not utilize it directly.

Several single-city navigation applications can be found that enable wayfinding for wheelchair users. Also, there are maps with accessibility information on public buildings. However, few applications provide a comprehensive solution for the differently abled to navigate in most cities. Perhaps the best-known international project is Wheelmap.org, an OpenStreetMap-based solution. Wheelmap concentrates on the needs of wheelchair users. Therefore, the color signs for locations are as follows: green color shows buildings that are fully wheelchair-accessible; yellow means partial accessibility; red signs show places that are not accessible for wheelchair users. The map also shows accessible toilets, based on the following criteria: the doorway's inner width is at least 90 cm, clear inner space is at least 150 x 150 cm, it has a wheelchair-height toilet seat, folding grab rails and accessible hand basin. [11]

Benner and Karimi [12] examined available ontologies (including INO and ONALIN) for pedestrian wayfinding and navigation, and found that there is a lack of generic approach for disability, existing solutions cover only parts of the whole spectrum. They suggest to focus more on the semantics of accessibility of the built environment.

Several countries have laws to ensure equal access to public services by the means of enforcing accessible building structures. These rules can be included in civil rights laws, or in architectural regulations. An example for the former case is the Americans with Disabilities Act of 1990 (ADA) in the USA, that regulates public buildings including medical facilities to be accessible for persons with motion (and other types of) disabilities. [13] In Hungary, the government regulation about national settlements planning and building requirements (OTÉK) orders the newly built and renovated public buildings to be accessible for wheelchair users by providing alternatives to stairs (slopes or stairlifts or elevators), the doorways with inner width of at least 90 cm, and accessible toilets. [14]

3 Usage Examples

Medical services are typically organized in several departments, each department located at different parts or on different floors of the hospital building. Departments usually include multiple medical-purposed rooms, such as examination and treatment, operating, or diagnostic imaging rooms, and also some non-medical rooms and services. Visitors, patients and medical staff may look for an actual room, a category of rooms (e.g. a toilet nearby), a department without the aim for a specific room, or just a location offering some services like an ATM or a vending machine.

The following use-case examples demonstrate the requirements for navigation in a medical facility:

The simplest case is when a person wants to get from one point of interest to another, for example finding a way from the building entrance to the room number 312. The navigation instructions should be simple and easy to follow: take the elevator to the third floor, go to the left in the corridor until you get to the vending machine, then look for the room number 312 nearby.

A somewhat more complicated situation arises when the visitor has incomplete information about their goal. A good example for this is a patient with an appointment for a heart checkup. He knows the name of the doctor (let us say Dr. Heart) and the department, which is outpatient adult cardiology in this case, but not the exact room number. A useful navigation application could offer two different options at this point: either navigate the patient to the information desk or the nurse station of the cardiology department, or offer a list of the rooms in that department with detailed information, so that the patient can find Dr. Heart’s office in the list.

Another example can be a visitor looking for a toilet. In this case, the exact room number is irrelevant; the navigation should provide directions toward the nearest toilet available for visitors (as opposed to toilets of ensuites belonging to inpatients rooms, or toilets reserved for the personnel).

The user of the navigation application initializes the search based on the names of services or departments. However, the naming conventions differ from country to country, and sometimes even among hospitals of the same area. To support hospital navigation successfully, the ontology must accommodate these differences. The following examples provide some insight into the problematic cases.

The co-location of different services can result in compound names for departments: in Hungary, gynecology departments are mostly co-located with obstetrics and neonatal care, therefore they have a common name for the three services (another example is the frequent co-location of dermatology and genitourinary care). On the other hand, there are departments with generalized names and area of care such as Internal medicine, but some hospitals provide separate departments for different specialties like cardiology or gastroenterology.

The route search should consider various parameters of the corridors, doors and other building parts to accommodate the special needs of the differently abled, e.g. wheelchair users, or the elderly with motion difficulties. When initializing the query, the user gives the starting point and the goal as well as the accessibility preferences for the route. These preferences should be finely tuned to the person's needs, including the maximum distance they have to walk without a resting opportunity, the number of stairs, or the angle of slopes.

For example, a person using a wheelchair has to avoid stairs, and can only travel between floors by elevator. They also need wide-enough doors to cross, and a route without high door thresholds, and steep slopes. But different kinds of wheelchairs (e.g. an electric wheelchair) can travel through passages of different steepness, and even a standard wheelchair can travel different steepness routes depending on the direction (e.g. rolling up or down the slope). A wheelchair user can overcome some smaller barriers with help.

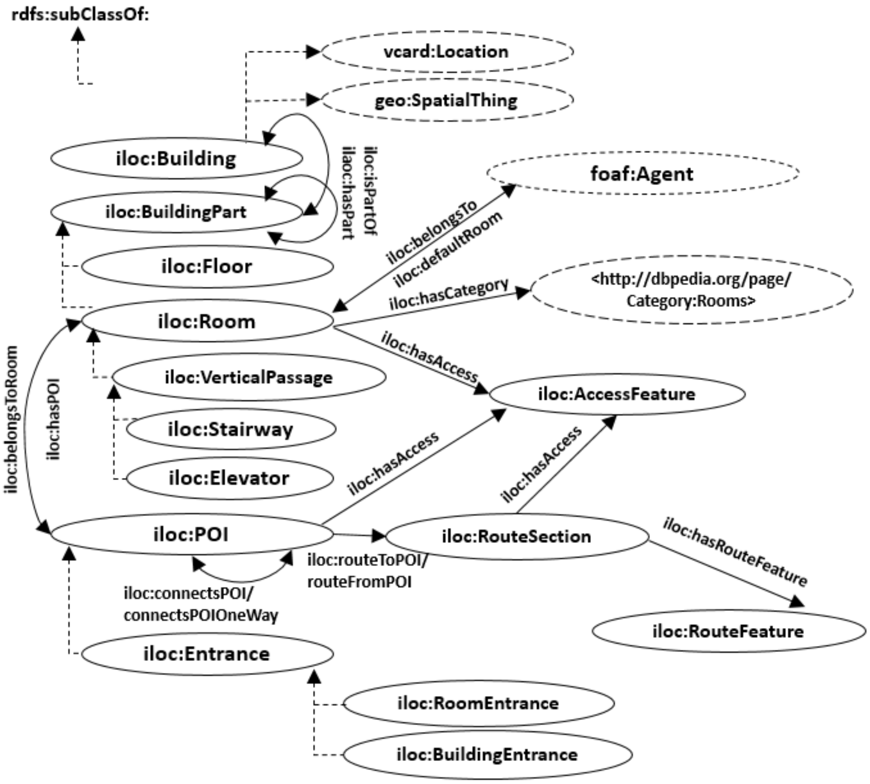
An elderly person using a walking cane may only want to walk less or climb as few stairs as possible, but it is not impossible for them to get through obstacles on the way.

The support of the staff of the medical facility (e.g. students, medical residents, ambulance personnel) is also among our aims. Therefore, navigation has to extend to staff-only rooms such as laboratories. However, the restricted access to these locations (e.g. door only passable using an RFID card) has to be described in the ontology and considered in the route search as a parameter.

4 iLOC Ontology

In this section the iLOC ontology is presented, which provides indoor location description of a general building, navigation method inside a building and accessibility attributes for people with disabilities. The iLOC ontology was designed in such a way that it can be extended easily by additional ontologies to serve environment specific use cases.

Figures 1 and 2 show the main classes as ovals and the most significant object properties as arrows between the classes. Classes defined in this ontology are using the *iloc* prefix, classes and properties used from other ontologies are prefixed with their own and such classes are marked with dashed line. Dotted arrows mark the subclass relationships between the classes.



iloc: http://lod.nik.uni-obuda.hu/iloc#
geo: http://www.w3.org/2003/01/geo/wgs84_pos#
vcard: http://www.w3.org/2006/vcard/ns#
foaf: http://xmlns.com/foaf/0.1/
qudt: http://qudt.org/schema/qudt#

Figure 1
iLOC ontology

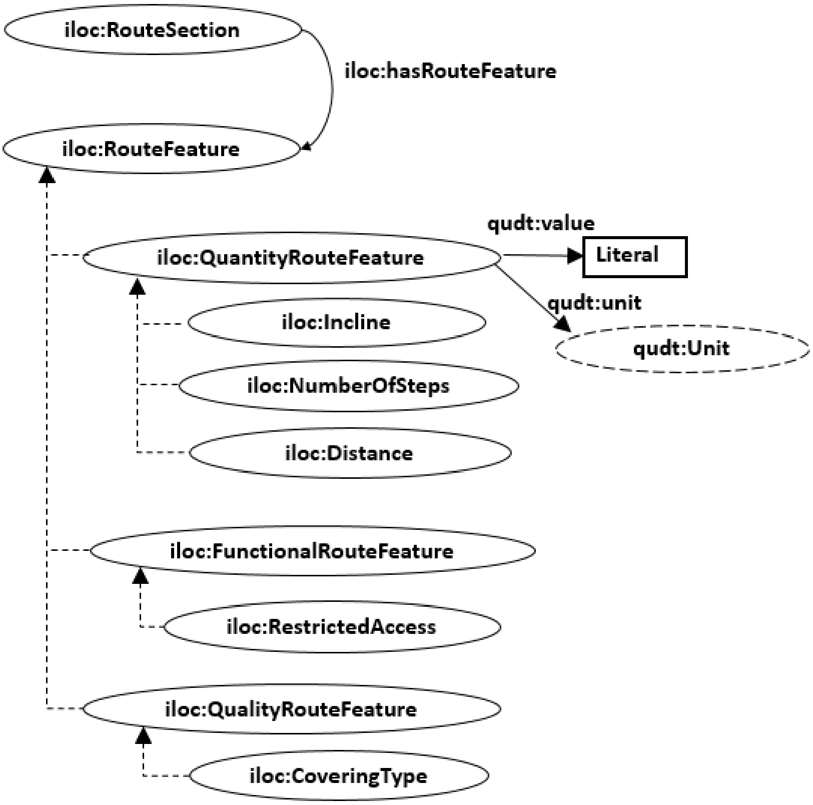


Figure 2
iLOC:RouteFeature and its subclasses

The iLOC ontology has three main classes: *Building*, *BuildingPart* and *POI*. *Building* class is a subclass of vcard:Location and geo:SpatialThing, in this way the address and the latitude, longitude coordinates can be assigned to its instances. *Building* entities have internal structure that the ontology aims to describe. *BuildingPart* provides an abstract concept to the different parts of the internal structure of a building. It has two subclasses: *Floor* and *Room*. A *Floor* entity represents an actual floor of a building. The *Room* class has a subclass *VerticalPassage* with special meaning for indoor navigation, its instances connect different *Floor* entities. *VerticalPassage* has two further subclasses: *Elevator* and *Stairway*.

The *isPartOf* object property, and its inverse property *hasPart* express hierarchical structural relationships within the building, e.g. a specific *Room* entity can be in *isPartOf* relation with a specific *Floor* entity. A *Room* instance is not tied to a specific floor directly, as there are examples when room height and floor height do not match and the room has entrances on multiple floors. Another example is an

Elevator or *Stairway* instance that can belong also to multiple floors. The solution is that a specific *Room* instance can be assigned to more floors with the *isPartOf* property.

Additional environment specific ontologies can define further *Room* subclasses for shopping malls, universities and other use cases; hLOC is such an ontology for hospital environments. The environment specific subclasses do not play a special role in the navigation process; they can act as custom filters to select specific type of rooms. In most cases, at the beginning of the navigation the exact starting point or target point is not exactly known by the user, a potential list of rooms can support the user’s choice. So the classification of the rooms can help to narrow down the potential list of rooms to a convenient length.

Room entities might belong to *foaf:Agent* entities, represented with the *belongsTo* property. With this property, *Room* instances can belong to specific organizations, and also to certain persons. In the hospital environment for example, in this way we can represent the location of a room in a department, or the linkage of a room to a certain person. Domain ontologies should define the necessary organization-type subclasses. A default *Room* instance might be assigned to a specific *foaf:Organization* entity with the *defaultRoomOf* property. This may be useful, if one has limited knowledge about the actual room he/she is looking for, only the specific organization is known (e.g. the department name in a hospital). Entities of the *Room* class can be classified into further external categories by the *hasCategory* property which can point to room categories defined in DBpedia[2].

The *POI* (Point of Interest) class plays an important role in the navigation process supported by iLOC. An indoor route is built up by a consecutive sequence of *POI* instances, where adjacent *POI* entities are connected with the *connectsPOI* property. In iLOC the *POI* class has one subclass: *Entrance*, which is further specified as *RoomEntrance* and *BuildingEntrance*. The *Entrance* instances have the special meaning of defining the connections between rooms or the entry point to buildings. Constraints in the ontology require that each building and room should have at least one entrance and a room entrance should belong to exactly two rooms. Similarly, to the class *Room*, additional *POI* subclasses can be defined in environment specific ontologies. Examples for *POI* class instances are statues, bank automats, display boards or vending machines.

The *connectsPOI* describes a direct route between two *POI* instances, the navigation route between two connected POIs is taken for granted. The *connectsPOIOneWay* is the asymmetric parent property of the *connectsPOI* property. It describes one-way routes between points. A navigation route is defined by a series of named and connected POIs. This approach supports

[2] http://wiki.dbpedia.org/

instructions like: "Cross the building entrance. Pass by the display board. Go to the stairs. Go to the 4th floor. Pass by the bank automat. Look for Room 407." The *hasPOI* property and its inverse property (*belongsToRoom)* express *POI* and *Room* relationships, a specific *Room* entity contains a given *POI* entity.

The *RouteSection* class represents a traversable path between two connected POIs. When defining a *RouteSection* instance its endpoint POIs must be specified. Route sections can be described by various attributes, among these there exist qualitative (e.g. covering type), quantitative (e.g. length, width, number of steps and incline) and functional descriptors or constraints (e.g. restricted access). It follows that certain accessibility constraints (e.g. usable for wheelchairs) can be inferred from certain route properties, but they can also be specified by manual entries on the basis of human decisions.

Extra information about route and disability profiles can be added to a RouteSection instance with the help of the *RouteFeature* and *AccessFeature* classes. The role of the *RouteFeature* class is to add extra descriptions to *RouteSection* instances that can be used in patient customized wayfinding queries. As shown in Figure 2, *RouteFeature* class has three direct subclasses *QuantityRouteFeature*, *QualityRouteFeature* and *FunctionalRouteFeature*, which can have further subclasses. *QuantityRouteFeature* subclasses (e.g. *Distance, Incline* or *NumberOfSteps*) contain instances having unit properties and numeric values as well. The QUDT ontology[3] can be used in providing generic measures and units to reuse. *QualityRouteFeature* subclasses (e.g. *CoveringType*) contain instances that describe specific qualities of the route sections. *FunctionalRouteFeature* subclasses (e.g. *RestrictesAccess*) contain instances adding extra information about the funcionality of the route. The *hasRouteFeature* property establishes the connection between the *RouteSection* and the *RouteFeature* classes. With the above property class hierarchy our aim is to present an extensible property framework, and not to give a complete and final solution for route descriptors.

The *AccessFeature* class represents different disabilities that require special features to traverse a *RouteSection* entity or to use a *Room* or *POI* instance. As we focused on supporting people with motion disabilities, the following instances were defined for the AccessFeature class for representing different accessibility needs: *Wheelchair, EWheelchair* (for electronic wheelchair), *WheelchairWHelp* (for wheelchair with help), *Stretcher* and *Stroller*. *RestrictedAccess* instance was introduced to represent areas where permission is required for the access. In the future additional instances can be added to the list in order to widen the accessibility features. The *hasAccess* property can be used to associate a specific disability constraint to a *RouteSection*, *Room* or *POI* instance.

[3] http://qudt.org/schema/qudt#

5 hLOC Ontology

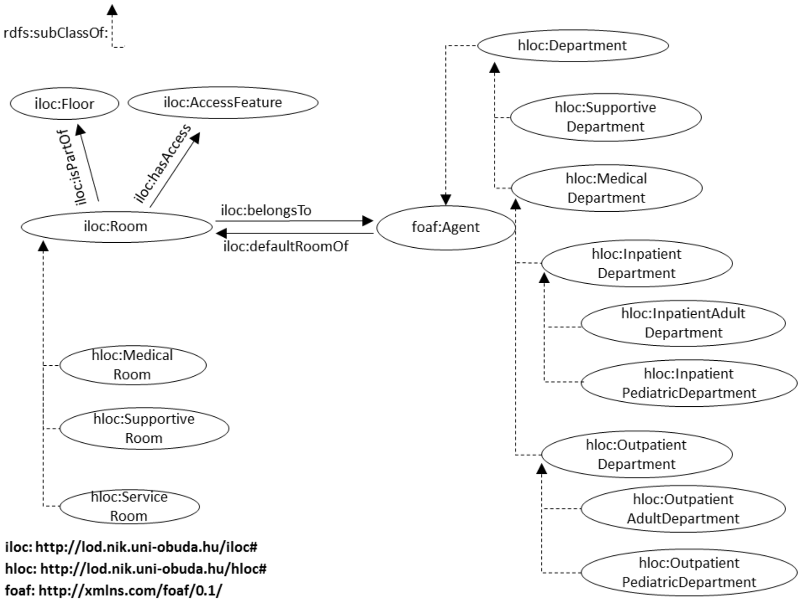

rdfs:subClassOf:
iloc:Floor
iloc:AccessFeature
iloc:isPartOf
iloc:hasAccess
iloc:Room
iloc:belongsTo
iloc:defaultRoomOf
foaf:Agent
hloc:Department
hloc:Supportive Department
hloc:Medical Department
hloc:Inpatient Department
hloc:InpatientAdult Department
hloc:Inpatient PediatricDepartment
hloc:Outpatient Department
hloc:Outpatient AdultDepartment
hloc:Outpatient PediatricDepartment
hloc:Medical Room
hloc:Supportive Room
hloc:Service Room
iloc: http://lod.nik.uni-obuda.hu/iloc#
hloc: http://lod.nik.uni-obuda.hu/hloc#
foaf: http://xmlns.com/foaf/0.1/

Figure 3
hLOC ontology

Hospitals have a number of room types and other building features that are typical of medical service facilities. For an ontology to support indoor hospital navigation, it has to contain descriptions of medical purpose locations.

iLOC does not have these specifics, however, it can be extended to provide support for indoor environments with special requirements. hLOC is an extension over iLOC that enables its usage in hospitals and to provide navigation support for people with motion disabilities. This ontology uses the prefix *hloc*.

The overview of the hLOC ontology is shown in Figure 3. The hLOC ontology extends iLOC with semantic classifications for hospital indoor structure. One of these is the definition of three iloc:Room subclasses in compliance with the room categories in the hospital environment. These are the classes: *MedicalRoom*, *SupportiveRoom* and *ServiceRoom*.

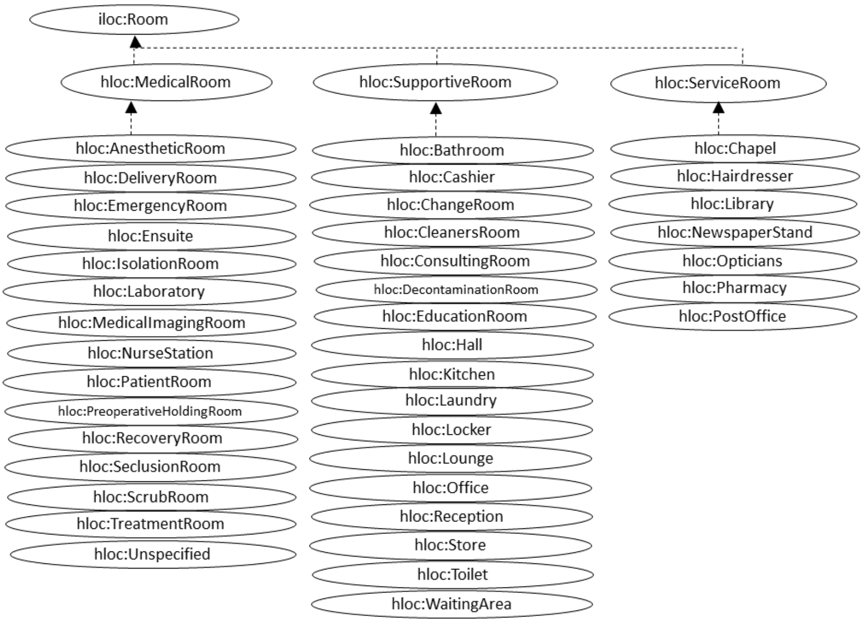


Figure 4
iLOC Room subclasses in hLOC

The subclasses of the MedicalRoom class cover medicine-specific room types like medical imaging rooms, operating and emergency rooms or ensuites for patients staying permanently in the hospital. Supportive rooms are also necessary for the daily operation of the hospital but not direct scenes of medical procedures. Examples of SupportiveRoom subclasses can be: DecontaminationRoom, Kitchen, Toilet, Office and so on. A service room can be any optional, convenience type service provided in the building of the hospital, such as a chapel, shop, pharmacy or post office.

In Figure 4 we give a recommendation for the list of iloc:Room subclasses (falling under *MedicalRoom*, *SupportiveRoom* and *ServiceRoom*) based on [10, 15].

The hLOC ontology also contains the class *hloc:Department* as the subclass of foaf:Organization. The hloc:Department has two direct subclasses: *SupportiveDepartment* and *MedicalDepartment*. MedicalDepartment can be further classified according to the following features: inpatient or outpatient, and adult or pediatric. Inpatient departments provide care for people in need of long-term medical assistance. Therefore, they contain ensuite type medical rooms to accommodate patients. Outpatient departments are specialized in shorter medical procedures, where the patient is able to go home after treatment.

Both inpatient and outpatient departments can be either for adults or children. In this way we get the following subclasses: *InpatientPediatricDepartment*, *InpatientAdultDepartment*, *OutpatientPediatricDepartment* and *OutpatientAdultDepartment*.

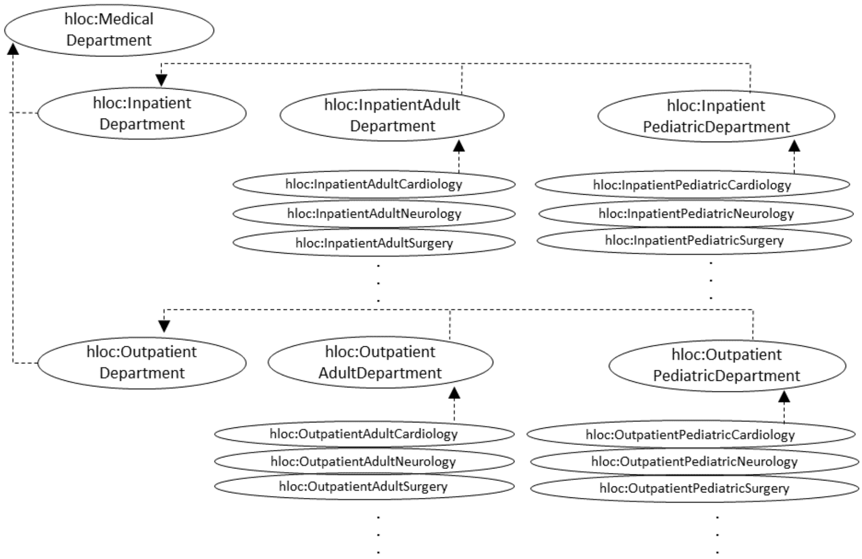

hloc:Medical Department
hloc:Inpatient Department
hloc:InpatientAdult Department
hloc:Inpatient PediatricDepartment
hloc:InpatientAdultCardiology
hloc:InpatientAdultNeurology
hloc:InpatientAdultSurgery
hloc:InpatientPediatricCardiology
hloc:InpatientPediatricNeurology
hloc:InpatientPediatricSurgery
hloc:Outpatient Department
hloc:Outpatient AdultDepartment
hloc:Outpatient PediatricDepartment
hloc:OutpatientAdultCardiology
hloc:OutpatientAdultNeurology
hloc:OutpatientAdultSurgery
hloc:OutpatientPediatricCardiology
hloc:OutpatientPediatricNeurology
hloc:OutpatientPediatricSurgery

Figure 5

Examples for Medical Department subclasses

The recommended complete list (made by studying the SNOMED CT taxonomy and the structure of several prestigious hospitals in Hungary and abroad) for Medical Department names is the following:

Addiction Services, Anesthesiology, Andrology, Cardiology, Clinical Laboratory, Critical Care, Dentistry, Dermatology, Diagnostic Imaging, Dietetics, Emergency, Endocrinology, Gastroenterology, Genetics, Genitourinary Medicine, Gynecology, Hematology, Hepatology, Immunology, Internal Medicine, Microbiology, Nephrology, Neurology, Neonatology, Obstetrics, Oncology, Ophthalmology, Orthopedics, Otolaryngology, Palliative Care, Pathology, Physiotherapy, Plastic Surgery, Psychiatry, Psychology, Respirology, Radiology, Rehabilitation, Rheumatology, Sports Medicine, Surgery, Toxicology, Trauma, Urology.

The exact name for a certain MedicalDepartment subclass arises as follows: [Inpatient | Outpatient] || [Adult | Pediatric] || [MedicalDepartment_name]. For example, for the term *Neurology*, *InpatientPediatricNeurology* is generated as an *InpatientPediatricDepartment* subclass. Figure 5 shows some examples for Medical Department subclasses following the structure of the hLOC ontology.

6 Evaluation

In this section data excerpts, OWL definitions and SPARQL queries are presented to demonstrate the possibilities of the developed ontologies. The following excerpt describes RouteSection, POI and Room instances:

```
@prefix iloc: <http://lod.nik.uni-obuda.hu/iloc/iloc#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

:RS014 a iloc:RouteSection;
iloc:hasAccess iloc:Stretcher;
iloc:hasRouteFeature [
        a iloc:NumberOfSteps;
        qudt:value 1.]
iloc:hasRouteFeature [
        a iloc:Distance;
        qudt:value 21;
        qudt:unit qudt:meter .]
iloc:routeFromPOI :CoffeMachine002;
iloc:routeToPOI :EntranceOfRoom203.


:CoffeMachine002 a iloc:POI;
rdf:label "Coffe machine on the 2nd floor"@en;
iloc:belongsToRoom :Hallway2ndFloor ;
iloc:connectsPOI :EntranceOfRoom203,
    :EntranceOfRoom202,
    :EntranceOfToilet2ndFloorLadies.

:EntranceOfRoom203 a iloc:Entrance;
rdf:label "Entrance of Room 203"@en;
iloc:belongsToRoom :Hallway2ndFloor ;
iloc:belongsToRoom :Room203 ;
iloc:connectsPOI :CoffeMachine002,
    :EntranceOfRoom202,
    :EntranceOfToilet2ndFloorLadies.
```

```
:Room203 a iloc:Room;
rdf:label "Room 203"@en;
iloc:belongsTo hloc:InpatientAdultDepartment ;
iloc:defaultRoomOf ex:InstitueOfCardiology ;
iloc:hasAccess iloc:WheelChair;
iloc:hasPOI :EntranceOfRoom203.
```

We can define the list of tolerable route features for wheelchair users and for visitors as RouteFeature subclasses: WheelChairAccessibleFeature and VisitorAccessibleFeature. Using the Protégé syntax for OWL restrictions, we can then write a definition for Wheelchair Accessible RouteSection individuals as follows:

```
WheelChairAccessibleRouteSection ≡ RouteSection and ( hasRouteFeature only
WheelChairAccessibleFeature)

VisitorAccessibleRouteSection ≡ RouteSection and ( hasRouteFeature only
VisitorAccessibleFeature)
```

Finally, if we have calculated a Route individual referring to all included RouteSections with the object property sections, the reasoner can automatically classify a route into a visitor accessible route using this class definition:

```
VisitorAccessibleRoute ≡ sections only VisitorAccessibleRouteSection
```

SPARQL 1.1 supports property path queries that can be used in wayfinding queries. It does not return what the path is nor the length of the shortest path - only whether there is such a path. By probing against different path lengths, this limitation can be overcome by a query similar to the following, which returns the shortest route (routes with the least steps) between <room1> and <room2> (with maximum length of three steps for brevity):

```
SELECT ?distance ?start ?p1 ?p2 ?p3 ?end WHERE {
  BIND (<room1> AS ?start ).
  BIND (<room2> AS ?end).
  ?p1 iloc:belongsToRoom ?start.
  ?p1 iloc:connectsPOI ?p2.
  ?p2 iloc:connectsPOI ?p3.
  ?plast iloc:belongsToRoom ?end.
  FILTER (?p3 = ?plast || ?p2 = ?plast || ?p1 = ?plast )
  BIND (if( ?p3 = ?plast , 3, if( ?p2 = ?plast , 2, if( ?p1 = ?plast , 1, -
1))) AS ?distance)
} ORDER BY ?distance LIMIT 1
```

Future SPARQL versions might better support such path queries by enabling access to the length of the path or to the specific elements of a route. OpenLink Virtuoso[4] has an extension for SPARQL with a transitive closure operator. The following SPARQL query using this extension provides the same result as the previous example:

```
SELECT ?step ?link WHERE {
  BIND (<poi1> AS ?start ).
  BIND (<poi2> AS ?end).
  ?start iloc:connectsPOI ?end OPTION(TRANSITIVE, t_no_cycles,
t_shortest_only, t_in(?start), t_out(?end), t_step (?start) as ?link,
t_step('step_no') as ?step, t_direction 3 ).
}
```

Gremlin [16], as a graph traversal language, is a functional language. The purpose of the language is to enable a human user to easily define a traversal, which is a tree of functions called steps, and thus, program a Gremlin machine. The following Gremlin code fragment provides the same result as the previous examples:

```
start = g.v(<room1>)

end = g.v(<room2>)

start.as('x').dedup().out('iloc:connectsPOI').loop('x')

  { it.loops < 3 && !it.path.contains(it.object) &&

    it.object != end }

.path.filter{it.last()==end}[0]
```

Although SPARQL is able to query longer path, not all implementation is capable to deliver results. Experiments were carried out with Apache Marmotta[5] and Virtuoso triplestores and also with Gremlin traversing engine. In some cases, Marmotta produced long response times in finding routes, queries were returning results in a wide range of 1-10 s. Virtuoso was significantly faster with its non-standard extension. The best performance was measured with the Gremlin traversing engine, queries were returning results in the 150-250 ms range.

Conclusions

In this paper we presented two ontologies iLOC and hLOC for supporting accessible free-text type indoor navigation in hospitals. The possible use of the ontologies was demonstrated by presenting SPARQL queries that future applications can build on to provide these navigation features. According to the classification of Worboys, iLOC with the extension of hLOC represents a hybrid indoor model since they contain semantic, topological and geometrical features as well, in the form of entity information, connectivity and distance descriptions.

[4] https://virtuoso.openlinksw.com

[5] http://marmotta.apache.org

The research contributions of the paper include the new ontological model of wayfinding which enables the use of SPARQL or Gremlin and inferencing for the calculation of indoor routes. This generic approach makes it possible to compute routes of different details, include third party local data into route finding and to apply various preferential and capability-based filtering for calculated routes. While previous approaches used specially constructed algorithms for wayfinding and disability specific concepts were built into the navigation ontologies, in our case it was possible to separate the generic task of navigation from concrete constraints on disabilities and building specific features. Furthermore, the domain of navigation in hospitals was investigated, and iLOC was extended with the novel hLOC ontology providing a wide range of way-finding functionality dedicated for hospitals.

References

[1] Berners-Lee, T. (2006) Linked Data-Design Issues, http://www.w3.org/DesignIssues/LinkedData.html

[2] Janowicz, K., Hitzler, P., Adams, B., Kolas, D., and Vardeman II, C. (2014) Five Stars of Linked Data Vocabulary Use. Semantic Web, 5(3) 173-176

[3] Szász, B., Fleiner, R., Micsik, A. and Simon-Nagy, G. (2016) iLOC – An Indoor Ontology, http://lod.nik.uni-obuda.hu/iloc/iloc-20161107.owl

[4] Fleiner, R., Simon-Nagy, G. and Szász, B. (2016) hLOC – A Hospital Ontology, http://lod.nik.uni-obuda.hu/hloc/hloc.owl

[5] Worboys, M. (2011, November) Modeling Indoor Space. In Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness (pp. 1-6) ACM

[6] Anagnostopoulos, C., Tsetsos, V., and Kikiras, P. (2005) OntoNav: A Semantic Indoor Navigation System. In 1st Workshop on Semantics in Mobile Environments (SME05) Ayia

[7] Dudas, P. M., Ghafourian, M., and Karimi, H. (2009, May) ONALIN: Ontology and Algorithm for Indoor Routing. In Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on (pp. 720-725) IEEE

[8] Scholz, J., and Schabus, S. (2014, September) An Indoor Navigation Ontology for Production Assets in a Production Environment. In International Conference on Geographic Information Science (pp. 204-220) Springer International Publishing

[9] Matuszka, T., Gombos, G., and Kiss, A. (2013) A New Approach for Indoor Navigation Using Semantic Webtechnologies and Augmented Reality. In Virtual Augmented and Mixed Reality. Designing and Developing Augmented and Virtual Environments (pp. 202-210) Springer Berlin Heidelberg

[10] SNOMED CT. Systematized Nomenclature of Medicine - Clinical Terms, http://purl.bioontology.org/ontology/SNOMEDCT

[11] Wheelmap.org. How does Wheelmap work?, https://community.wheelmap.org/en/about/how-to-map/

[12] Benner, J. G., Karimi, H. A. Accessible Wayfinding Ontologies for People with Disabilities. Extended Abstract for the RDWG Online Symposium on Accessible Way-Finding Using Web Technologies, 3 December 2014

[13] The Americans with Disabilities Act of 1990 and Revised ADA Regulations Implementing Title II and Title III, http://www.ada.gov/2010_regs.htm

[14] 253/1997. (XII. 20.) Korm. rendelet az országos településrendezési és építési követelményekről - Akadálymentesítésre vonatkozó kivonat, Mozgáskorlátozottak Egyesületeinek Országos Szövetsége (MEOSZ), www.meosz.hu/doc/p26-2otek_segedlet.doc

[15] Department of Human Services, Victoria. (2004, November) Design Guidelines for Hospitals and Day Procedure Centres http://www.healthdesign.com.au/vic.dghdp/dghdp_content/guidelines/dghdp_part_b_complete_20_377.pdf

[16] Rodriguez, M. A. (2015, October) The Gremlin Graph Traversal Machine and Language (invited talk) In Proceedings of the 15th Symposium on Database Programming Languages (pp. 1-10) ACM