Ontology based Indoor Navigation Service for the ILONA System

Dániel Péter Kun, Erika Baksáné Varga and Zsolt Tóth

***Abstract*—An ontology based way finding algorithm is presented in this paper that allows route generation between two separate parts of an indoor environment. The presented ontology provides a flexible way to describe and model the indoor environment, in addition it fits and extends the existing model of the ILONA System. Ontology reasoners provide an efficient way to perform complex queries over the knowledge base. The instances, that are queried by the reasoner, are used to initialize the graph which represents an indoor environment. Due to parameterization of the reasoner, different graphs can be generated from the ontology which makes the way finding algorithm flexible. Thus, the task of indoor way finding was converted into a well-known graph search problem. Dijkstra's shortest path algorithm is used for route generation in the graph yielded. The algorithm was implemented and tested in the ILONA System and its functioning is demonstrated by real-life scenarios.**

I. Introduction

Indoor positioning systems aim to find people or objects inside a building. The interest in this topic grows with the widespread use of smartphones in modern society. Multiple solutions use the sensors commonly found in an average smartphone. The existing solutions differ in the technology and heuristics used and in the costs. Although the first indoor localization system was developed in the early 1990's, there is no common solution for indoor positioning unlike the Global Navigation Satellite Systems for outdoor environment.

Availability, cost and accuracy are the key criteria during the development of an indoor positioning system. The most recent solutions consider smart phones as client device due to their low cost and wide range of sensors. Thus, Bluetooth, ultrasound and WiFi RSSI are all popular technologies for indoor positioning. Every technology differs in cost and precision. WiFi RSSI based solutions have very low installation cost because they use the communication network that is already established, but these systems usually achieve only about 3 meter accuracy. Apple's iBeacon and Estimote beacons are based on Bluetooth technology and an accuracy of 1-4 meters could be achieved with this technology depending on the location's size. Finding a precise and cost-efficient, while widespread technology is a challenge for the developers even nowadays.

Indoor navigation systems are built upon indoor positioning systems and extend their functionality in the same way as outdoor navigation systems make use of the service provided by global positioning systems. For example, Global Navigation Satellite Systems use the Global Positioning System to determine and track the users' location and a navigation software such as iGo or Waze for way finding and navigation. Malls, airports, hospitals often place floor plans to facilitate the indoor way finding for their visitors. Despite the popularity of this technique, it has numerous drawbacks. Firstly, it is an offline solution and it has to be replaced when the environment changes. Secondly, the poster size grows with the size of the building so its applicability is limited. Thirdly, finding of a specific location in a huge floor plan could be difficult and may take a long time. On the other hand, online indoor navigation systems can deal with the above challenges. Although updating of the map would require some modifications on the server, the zooming in the floor plan is a simple task because there are software libraries to solve this problem. Finally, searching with an online application is quite simple and it would increase the availability of the indoor navigation system.

Daniel Péter Kun, Erika Baksáné Varga and Zsolt Tóth are with the Institute of Information Science, University of Miskolc, Hungary
{kun3,vargae,tothzs}@iit.uni-miskolc.hu

DOI: 10.36244/ICJ.2018.3.4

TABLE I: Comparison of ontology-based semantic location models

Name	Usage of Ontology	Application Rules
LOC8	space model ontology, context model ontology, sensing model ontology	spatial relationship rules
OntoNav	user navigation ontology, indoor navigation ontology	path selection rules
Onalin	user navigation ontology, indoor navigation ontology	path selection rules, user preferences
Smart Hospital Project	entities ontology, semantic locations ontology, physical locations ontology	sensing areas rules
ILONA	indoor navigation ontology	path selection rules, restrictions

The ILONA (Indoor Localization and Navigation) System [16] is a web-based indoor navigation system developed at the Institute of Information Science, University of Miskolc, Hungary. The aim of the system is to provide a common research environment for evaluation and testing of positioning and way-finding methods. ILONA System can be used to provide way-finding services for students, employees and visitors between and inside the buildings located in the university campus which occupies a continuous territory of approximately $350.000m^2$ (85 acres) including 25 buildings.

The paper presents a case study that demonstrates the applicability of the ontology created for storing semantic information for indoor navigation.

A. Related Works

Indoor navigation has shown numerous applications. One of the best examples is robotics, where the system relies on the sensors found in the machine. Other well-known applications are smart hospitals, exhibitions and e-commerce. All of these systems perform functions specific to their own field. Since navigation information in indoor environments is more than geo-information, researchers have been proposing the addition of semantic models to offer suitable representations and applications. Map-based indoor navigation systems are straightforward implementations of global navigation applications in indoor environment. Map–based systems can use their own indoor representation of the building [13] or standard modeling tools. IndoorGML provides a standard format for modeling indoor environments, for example it was used to model educational buildings in [7]. One group of semantic models is based on ontology. This section introduces the existing systems [1], [2], [14], [18] using ontologies. Table I summarizes the ontology-based semantic location models based on [17] extended with the ILONA ontology.

1) LOC8: LOC8 [14] is a location model and extensible framework for programming with location. The whole framework contains three models that are called context model, sensing model and space model. These models are expressed by ontologies, which provide API to describe and apply context information into the location-based services. The space model represents the locations with a relative location, a symbolic representation and a geometric region. The context model represents the additional layers of information that belong to the locations that can be used for further reasoning. The sensing model maps information about the traversing entities of the model (e.g. people). LOC8 is based on the OWL API and the Protege ontology editor.

2) OntoNav and ONALIN: OntoNav [1] is a semantic indoor navigation system and an ontology framework for handling routing requests. User Navigation Ontology (UNO) and Indoor Navigation Ontology (INO) are Knowledge Models in the OntoNav System. UNO reuses and extends some concepts of existing ontologies, like GUMO [5]. INO serves as the model for both the path searching and the presentation tasks of the system. Based on UNO and INO, path selection rules described by SWRL (Semantic Web Rule Language) could discard some paths which are physically not accessible for users, and identify paths which match users' preferences.

The system consists of a navigation service, a geometric path computation service and a semantic path selection service. The navigation service serves as the main connection between the user and the system, receiving the requests of the user and returning the optimal path for him. The geometric path computation service is responsible for the calculation of the physical paths for the user. This is handled with classic graph theory. The semantic path selection service handles the decision about the best available path. This module compares the available path with the profile and the requests of the user then the module returns the optimal solution.

ONALIN [2] is the extension of the OntoNav system with the American Disability Act. Good examples for this extension includes height of stair and handrail availability.

3) Smart Hospital Project: The smart hospital project [18] was created to represent mobile entities in a navigational model within medical facilities. Its model consists of atomic location, semantic location and physical location. Physical locations detail the current whereabouts of the mobile entity, while semantic locations describe the environment through identifiers like door, room, floor, building, etc. The atomic location connects the two previous models through detailed information gathered by positioning techniques.

B. ILONA System

Our global motivation is to launch a Smart University Project (similar to the Smart Hospital Project) that takes into consideration the preferences of university employees, students and visitors. The test site of the initial project is the University of Miskolc (Hungary) with 1000 employees, 15 thousand students (approximately 1% disabled, 1% foreign language speaker) and thousands of visitors.

This project needs a system that integrates location and navigation services. Map-based systems use only geo-information, while ontology-based systems can also make use of semantic information related to indoor spaces. The design and implementation of the ILONA System [16] was performed in view of these requirements. As a consequence, the navigation component can use an ontology-based model, similar to ONALIN.

The main difference between this model and the one that we propose in the present paper is the approach to user preferences in path selection. ONALIN lets users add their preferences concerning the route (e.g. I want to use the elevator). Our inverse approach does not presume apriori knowledge about the indoor facilities. Users can declare their restrictions by specifying what they do not want to or cannot use (e.g. I cannot use the stairs) and the system provides all alternative ways to reach the destination. In this way, for example, the hanging wheelchair lift instead of the handrail stairs will not be disregarded. In other words, the semantic model of the ILONA system applies restrictions, rather than preferences, when making decisions. Consequently, we do not need to have any apriori knowledge about the building's facilities, therefore the ontology does not include special annotations. Users can give their "negative" preferences and the system will automatically exclude these gateways from the route. Any other alternatives will be displayed. If an invalid restriction is given, for example the user states that he does not want to use elevator and there is no elevator in the building, this request is ignored.

The ILONA System [16] was designed and developed to provide a flexible framework to test and compare various indoor positioning and navigation methods. The component based architecture and the loose coupling of services allows the extension of the system with various algorithms. The main subsystems are the `measurement`, `positioning`, `navigation` and `tracking` modules that are shown in Figure 1. These components are developed individually. The `measurement` subsystem defines the data model [15] of the fingerprinting database and provides web services for the managing of measurements. The `positioning` subsystem is a collection of various indoor positioning methods and it is continuously expanding with novel implementations. The `tracking` subsystem provides a service for observation of the users' movements. This paper focuses on the `navigation` subsystem of the ILONA System.

While the ILONA System defines the component structure and the web interfaces that invoke services whose expected behavior is also defined, the concrete implementation of these services is delegated. In this way the system can be extended with a novel way finding algorithm in three steps. Firstly, the novel algorithm has to be defined in a class that implements the corresponding interface. Secondly, the library that contains the novel method has to be added to the web component. Finally, the web component has to be configured to use the new method. Because of its flexibility and the services already provided, the ILONA System is chosen to implement and test the way finding method presented here.

The extension of the ILONA System with an ontology based navigation service allows the comparison and analysis of indoor navigation and way finding algorithms. Due to the component based architecture of the ILONA System, the navigation and positioning algorithms can be analyzed, tested and evaluated independently. In addition, the design of the ILONA System facilitates its extension and reconfiguration. Thus, ILONA System provides a common environment for the development and comparison of indoor positioning and navigation methods. Consequently, we can consider the way finding algorithm presented in this paper as a reference for other way finding methods.

C. Goals

The goal of the present research was the extension of the ILONA System with an ontology-based way finding method. Three major challenges occurred during the design of the module. Firstly, the ontology should fit to the existing data model of the ILONA System that is used and defined by its other components. Secondly, the way finding algorithm had to be designed. Finally, the implementation of the navigation module should not affect the other existing components. In addition, the suggested way finding algorithm and its implementation demonstrates the flexibility and extendability of the ILONA System.

The way finding algorithm is the major contribution of the paper. The applied ontology model allows the adaptation of the navigation

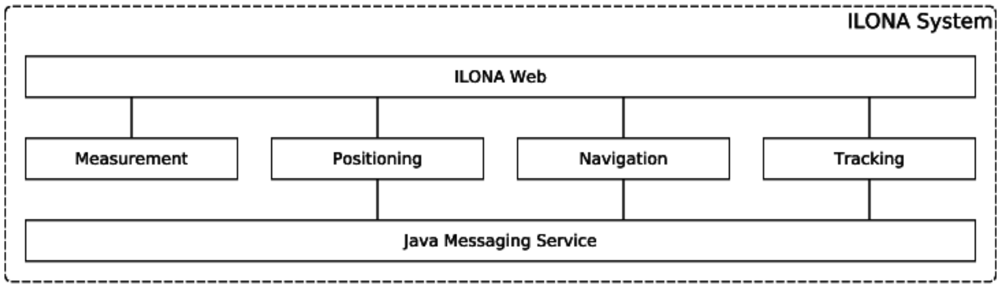


Fig. 1: Subsystems of the ILONA System [16]

system to arbitrary indoor spaces such as transport stations, shopping malls, educational buildings and offices. In addition, this algorithm can be used as a reference during the development of other indoor navigation algorithms. Because simplicity was kept in mind during the design, the way finding method can be improved with the application of different graph search algorithms.

II. METHODS

The hiding of technical details, the extension of the existing model, parameterizability and support of the administrator were set as the main criteria about the ontology-based navigation service. The ILONA System was designed to be as general as possible so it provides only an abstract navigation service to the end users. Hence the system can work with both ontology- and map-based way finding algorithms. Due to this abstraction, hiding the ontology and other technical details is necessary. The ILONA System provides data types for modeling measurements and position information. The data model of the navigation module and its ontology [8] should fit to the existing model [15]. In addition, the navigation module requires additional information to model the indoor environment properly. As a consequence, the data model of the `navigation` subsystem should be extended. The presented ontology introduces the concept of gateways that represent the connection between two zones. The ontology-based way finding algorithm should be configurable and able to generate different routes based on various restrictions. The navigation module defines a general set of restrictions that can be chosen by the users in order to omit certain types of gateways. For example, a disabled person could generate a route that avoids stairs. Although the ontology-based navigation service should be transparent for end users, administrators should be able to manage the ontology and other parameters of the way finding algorithm. In order to facilitate the administrator's work the ontology-based navigation component should provide some services intended to be used by administrators only.

Design and development of the navigation module of the ILONA System has three major challenges. Firstly, the navigation module has to extend the existing data model of the ILONA System without its modification. Since a well-defined function set is assigned to each module, this can only be modified via extension based on the Open-Closed Principle. This requirement makes the design of the navigation component challenging. Secondly, the navigation subsystem should use the positioning service of the ILONA System which is assumed to be provided. Although the behavior of the positioning service is defined by its interface, neither the performance nor the accuracy of the positioning service is given. Because the navigation service relies on the positioning service, its performance and accuracy may depend on the actual positioning algorithm. This dependency could make the testing and evaluation of the navigation service difficult. Finally, the design of the navigation module has to fit well to the existing modules. The designed ontology-based way finding method deals with the above challenges and can be integrated to the ILONA System. In addition, the ILONA System can provide services for route generation in arbitrary indoor areas such as shopping centers, transport stations or office buildings. The next sections provide detailed information about the design and development of the navigation service in the course of which all three challenges were taken into consideration.

A. Ontology Model

Navigation models in ubiquitous computing should involve context information reflecting the semantics of the actual application. Navigation context expresses the state transitions of entities in the navigation route. The ontology-based modeling approach is a semantic way of organizing and sharing context knowledge augmented with advanced reasoning capabilities for processing context information [17].

The ontology model of the Navigation component of the ILONA System is based on the concept of `Zone` which is a closed, disjunct section of the indoor space. Each zone is named and uniquely identified, and denotes an arbitrary area with no specific dimensional restrictions. The exact spatial and dimensional definition of each zone comes from the measurement component of the ILONA System. Zone entities are connected by `Gateways` – like elevators or stairs – which are also modeled as entities. By definition, gateways are named and uniquely identified, closed and disjunct indoor areas having the role of connecting zones. In this way the graph representing a navigation route consists of nodes denoting Zones and Gateways. This simple representation allows for flexible modeling of indoor environments. The routing algorithm should find a way between two Zones by determining the sequence of Zone and Gateway nodes to go through.

Gateways are modeled as entities because in this way attributes can be assigned to them (e.g. permission of use). Consequently, they are represented as nodes in the navigation graph where residence can be registered (e.g. getting stuck in an elevator). A gateway may connect more than two zones. This many-to-many relationship between the zones through the gateways is modeled by two has_a relations: a zone may have multiple exit gateways (from) and it may also have multiple entrance gateways (to). Either of these relationships can be omitted to represent a one-way gateway (e.g. chashier's desk as exit from supermarket). Following the VOWL notation [9], Figure 2 illustrates the possible navigation routes between two zones (Ground floor lobby and Room 109) in the building of the Information Technology Institute at the University of Miskolc, Hungary. The figure shows gateways as hatched nodes and extends the navigation path with the one-way emergency exit.

There are five gateway types defined in our model at present: doors, elevators, escalators, stairs and virtual gateways. This list can be extended with other facilities on demand. Doors are separators between

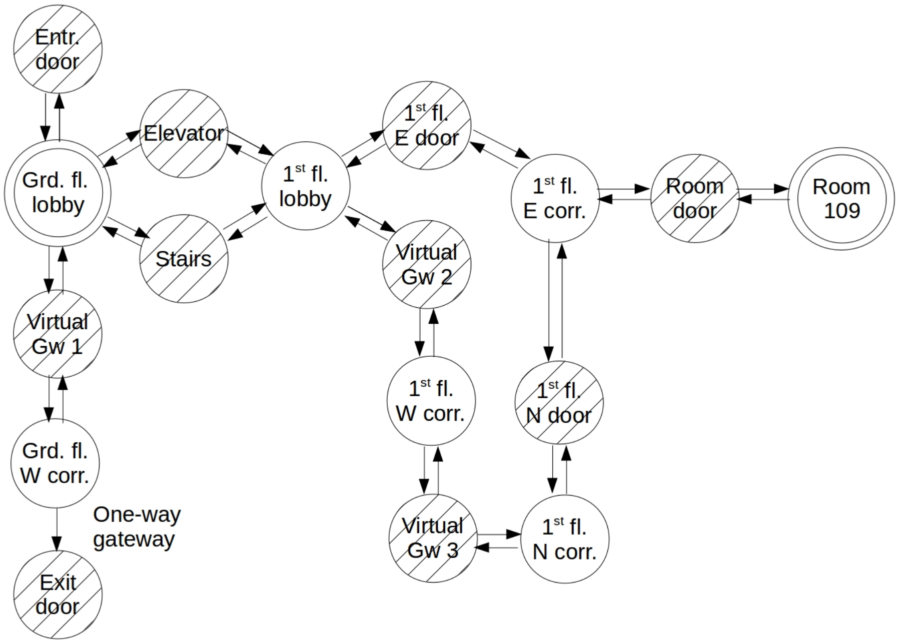


Fig. 2: Navigation route model between zones

zones with small physical extension, i.e. their dimensions are not considered in the computations. Elevators, escalators and stairs have considerable dimensions and they are differentiated according to their attributes (e.g. accessibility). In the case of virtual gateways, the connection between the zones is established through non-physical objects (e.g. joint corridors not separated by doors).

For modeling complex indoor spaces the possibility of grouping zones is necessary. The greatest zone group defined in our model is "building”. This is comprised of several other zone groups which can be established on the basis of various criteria, e.g. location, functionality or ownership. For example, the floors of the building are zone groups according to location containing neighboring zones. The laboratories of the Automation Department reside in the western corridors of the ground and 2nd floors. Although these are not neighboring zones, they belong to the same zone group when dividing the zones according to ownership. Generally speaking, the grouping of zones must comply with the following rules:

1) Each zone is a member of one or more zone groups.
2) A zone group may have any number of other zone group members.

This model was inspired by the Composite design pattern [3], where zone groups are composites of zones representing recursive part-whole hierarchies (see Figure 3).

B. Ontology Definition

In computer science, the term "ontology" was introduced by Gruber [4] to mean "the specification of a conceptualization". This is a formal description of the concepts (entities with their attributes) and the relationships that exist in a given domain. The ILONA navigation domain ontology can be accessed via Protege [12], the web-based ontology editor that assists the creation and manipulation of ontologies. The knowledge representation language used for ontology construction is the Web Ontology Language (OWL 2) [10]. It provides classes, properties, individuals, and data values on the syntax level. OWL ontologies are mapped to RDF graphs and stored as RDF documents.

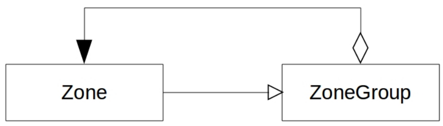


Fig. 3: The modeling of zone groups

The first task in ontology creation is to define the taxonomy of classes described by various attributes. In the ILONA ontology the top-most classes are Gateway and Zone with two attributes: name and id, where id refers to a value coming from the measurement component of the ILONA system. The five gateway types are implemented as subclasses of the Gateway class. The ZoneGroup class is derived from the Zone class and in order to represent zone groups as composites of zones an auxiliary class should be introduced to denote their relationship. This ZoneGroupMembership class connects each zone to a zone group. Since ZoneGroup is the derived class of the Zone class, this implies that zone groups can be members of other zone groups. Our model specifically defines two subclasses of the ZoneGroup class: Building and Floor to demonstrate the recursive part-whole hierarchy of floors in buildings. The proposed ontology model has not been fully implemented yet. As mentioned above, the grouping of Zones is possible according to several criteria apart from location (e.g. functionality: offices, laboratories, corridors, etc.; ownership: departments, faculties).

The second task is to define the relationships between the classes other than specialization. In the ILONA ontology the most important object properties are those that define the entrances and exits of zones via gateways. By default, each zone may have zero or more entry and exit gateways, so the from_zone and to_zone object properties are not restricted. For modeling zone groups as composites of zones through the ZoneGroupMembership class, two additional membership relations are defined: has_zone and has_zonegroup. Technically, these object properties implement the many-to-many relationship between zones and zone groups.

Figure 4 shows the ILONA ontology drawn by OntoGraf which is a visualization tool embedded in Protege-5.0.0 desktop edition [12]. The presented ontology contains only physical zones, zone groups and gateways, and the figure shows only the taxonomy of classes without attributes in order to emphasize only the major concepts. Continuous line blue arrows represent has subclass relations in the taxonomy. Dashed arrows are for those object properties that are explicitly defined. Between the `Zone` and `Gateway` objects the yellow arrow denotes the `to_zone`, while the red arrow denotes the `from_zone` property. The `to_zone` property is used to express that 'a specific gateway leads to a given zone'. On the other hand, the `from_zone` property states that 'an actual gateway is an exit from a given zone'. Since OWL ontologies allow only one-to-many (parent-child) relations, the many-to-many relationship between the `Zone` and `ZoneGroup` objects should be converted to two has type object properties (denoted by dashed arrows) the domain of which is the `ZoneGroupMembership` technical entity.

C. Ontology Generation

The presented navigation module provides some functions for the administrators in order to facilitate the configuration of the ontology used. The structure of the indoor environment is stored in a knowledge base whose model is defined by the ontology [8]. Although the data model is given, the knowledge base has to be populated with instances that is time consuming. Moreover, the `Zone` object stored in the ILONA System should be mapped to a corresponding instance of the ontology. Manual insertion of these objects is not just time consuming, but may cause errors and mismatches. To reduce the probability of mismatches and to facilitate the population of the knowledge base, an ontology generation service is defined that initializes the structure of the knowledge base and inserts the records from the database of the ILONA System.

The ontology generation process is shown in Figure 5. The administrator sends an HTTP request to the server to ask for the initial knowledge base. The HTTP request is processed by the controller layer of the navigation module that parses and validates the request. Valid requests are forwarded to the `OntologyGenerationService` that implements the business logic. The initialization of the knowledge base consists of two major steps that are the querying of the Zone objects and the population of the knowledge base.

In the first step, the `Zone` objects are queried via HTTP from the controller of the `measurement` module of the ILONA System. During the query, the request is also parsed and validated then forwarded to the corresponding service that uses Data Access Objects to manipulate the database. Data Access Objects hide the technical details of the storage from the service. The current version of the ILONA System supports only MySQL database that is manipulated via JDBC. The objects queried from the database are converted by the Data Access Objects and forwarded to the service. The service returns with a collection of the objects that are marshalled to JSON format and sent back to the `navigation` module as a HTTP response.

In the second step, the queried `Zone` objects are used to populate the knowledge base. The `persist` component of the `navigation` module is used to initialize an empty knowledge base from the ontology template that is stored on the server. After that the Zone objects are inserted into the temporary knowledge base and then the temporary file is sent back to the administrator via HTTP by the `controller`. At the end of the process, the initialized knowledge base containing ontology instances is stored on the administrator's computer and it is ready for further modification.

D. Way Finding Algorithm

The `navigation` module of the ILONA System defines services for way finding but its implementation is delegated. Thus, the users and the developers have a different view of the system. From the user's point of view, the ILONA System always provides the desired service via the same path and its expected behavior is defined. From the developer's point of view, the ILONA System provides a bunch of functions and facilitates for the development, publishing and testing of different way finding algorithms. In other words the ILONA System can be extended with various way finding algorithms while its behavior remains the same.

The presented way finding algorithm traces the routing problem back to a well-known graph problem and uses the ontology to store graphs whose number grows combinatorially with the number of nodes. Figure 6 shows the sequence diagram of the route generation process. From the user's point of view, the way finding service is a black box whose input are the source, the destination and the restrictions. The way finding service returns with a sequence of zones that defines the path between source and destination. Way finding is a complex process that consists of three major steps: querying from the ontology, route finding in the graph and conversion of the path to zones.

In the first step, the ontology is used to query the navigation graph. The user sends the source and destination along with a list of restrictions to the server. The corresponding web controller processes and validates the request, then calls the way finding service. The way finding service uses the `OntologyDAO` to filter the gateways based on the restrictions. Filtering is performed by an ontology reasoner and its results are used to populate a `ZoneMap` object that encapsulates a graph. Nodes of the graph store only the zone identifier.

The second step is responsible for way finding. Because `ZoneMap` is a graph whose nodes are the zones, and the edges represent the gateway connections between the zones, the way finding problem can be solved with graph algorithms. The edges are unweighted in the graph because the navigation service does not calculate time or distance estimation for taking the route between the source and destination zones in the present stage of the research. On the other hand, finding the right gateway could be challenging. Dijkstra's Shortest Path algorithm is commonly used to find a path between two nodes in a graph whose cost is minimal. Since the weights of the edges are equal, the shortest path will be the one that has the minimum number of edges. In other words, the way finding algorithm will return with a path that contains as few gateways as possible.

In the final step, the path found is converted to a list of Zones and presented to the user. The way finding algorithm returns a sequence of Zone identifiers. The way finding service has to query the Zone objects by their identifier. To query Zone objects, the service performs HTTP requests to the corresponding web controller of the `measurement` subsystem. The way finding service returns with a list of Zone objects queried. The way finding controller marshalls these objects into JSON format and send them to the user in a HTTP response. The further processing and visualization of the generated route is not part of the current research.

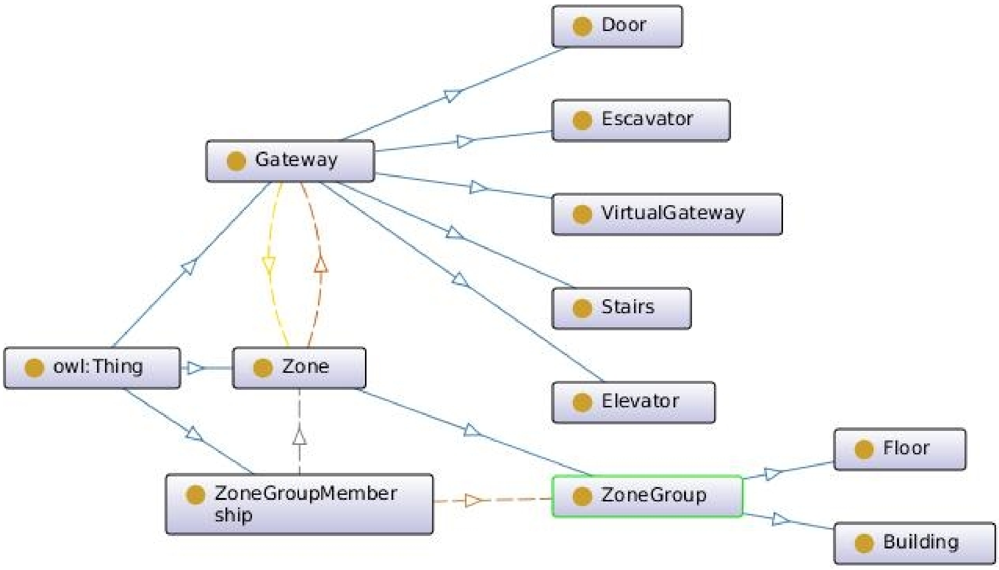


Fig. 4: Ontology structure

III. RESULTS

The ontology-based indoor way finding method was implemented as a part of the ILONA System. The implementation was tested on unit, component and integration levels. In addition, real-life case studies were used to demonstrate the applicability of the proposed way finding method.

A. Implementation of the `navigation` subsystem

The `navigation` subsystem is one of the major elements of the ILONA System. The implementation of the way finding algorithm required the modification and extension of some modules in the `navigation` subsystem, but these modifications were limited to the current subsystem. Furthermore, modifications had to fit the module structure of the ILONA System.

Figure 7 shows the module structure of the `navigation` subsystem and the modifications. The `model` module has been extended with classes for modeling the indoor environment. Modification of the `service` module was not required during the implementation. The `service-impl` module has been added and it defines the way finding algorithm. The algorithm is enclosed into a class that implements the `WayFindingService` interface so it can be easily integrated to the ILONA System. The way finding algorithm extracts data from the ontology that is stored in the file system. The storage and query functions related to the ontology are defined in the `persist`, and implemented in the `persist-ontology` module. Although the `controller` module has been extended with the ontology generation controller, the way finding service controller was not modified.

The `model` module defines the domain objects and implements a low level business logic such as value validation. The `ZoneMap` class represents a graph that is used for way finding. The graph data structure is provided by the JGraphT Java library that allows the modeling of any kind of graphs. The general graph representation of JGraphT was adapted to the ILONA System by the `ZoneMap` class. Hence the nodes of the graph adapted by `ZoneMap` store `Zone` objects. The instantiation of `ZoneMap` is performed by classes of the `persist-ontology` module.

Persistent storage of the domain object is defined in the `persist` and implemented by the `persist-ontology` module. Due to the separation of definition and implementation, the modules became loosely coupled and they could be developed independently. In addition, all ontology related program code was limited to the `persist-ontology` module. The OWL API [6] was used to represent and manage ontology entities. Querying and data extraction were performed with the HermiT Reasoner [11] that facilitates the execution of complex queries. These queries are composed on the basis of the restriction set given by the user. Based on the restrictions, the reasoner returns with the edges that are allowed.

The way finding algorithm was implemented in the `service-impl` module. The algorithm is enclosed into a class that implements the `WayFindingService` interface of the `service` module. Thus, the algorithm can be used via the interface in other modules that facilitate the configuration of the ILONA System. The algorithm queries the graph nodes and edges that fulfill the restrictions from the ontology via the `OntologyDAO` interface that is defined in the `persist` module. Query results are used to initialize a `ZoneMap` object that is used to find the shortest path between the source and destination zones. When the `ZoneMap` returns with a list of zone identifiers, the algorithm queries the corresponding `Zone` objects. So the way finding algorithm returns with a list of zones that describe the route generated.

The web services are stored in the `controller` module that is responsible for the processing and validation of HTTP requests and invocation of the service methods. The web controllers are mapped to certain URL path and wait for requests. Request parameters and the yielded response data are represented in JSON format. Consequently, the navigation component can be integrated with any kind of client devices.

Figure 8 details these components and their connections according to the N-Tier Architecture. The Presentation Tier is called Web Tier,

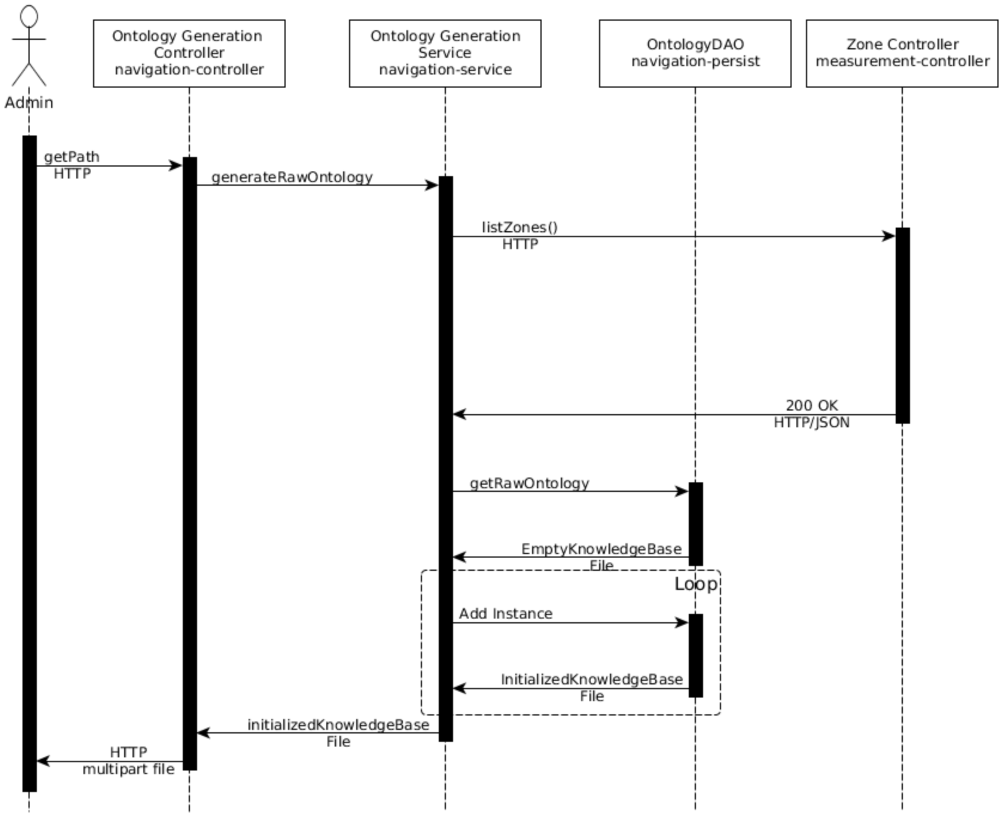


Fig. 5: Sequence diagram of the ontology generation process

the Business Logic is represented by the Service Tier and the Data Access Tier is denoted by the Persistence Tier.

The Web Tier provides an interface to the services provided and integrates the other modules of the `navigation` subsystem. Controllers are the entry points of the functions. The `Data Transfer Objects` are responsible for the conversion of domain objects into JSON format and simplify the HTTP messages. The `application-context` defines the configuration of the web application and it is used by the Spring Framework which is responsible for dependency injection.

The Service Tier contains the definition of `service` interfaces and their implementation. The ontology generation and way finding algorithms are implemented in this tier. The web controllers use these services via their interface so the concrete implementation can be changed. Hence, the extension of the ILONA System with other way finding algorithms would require only two steps. Firstly, the novel algorithm should be enclosed into a class that implements the interface of the way finding service. Then, the application should be reconfigured to use the way finding algorithm recently added.

The Persistence Tier is responsible for the storage and retrieval of domain objects. The `measurement-persist` and `navigation-persist` components are distinguished according to which subsystem they belong to. The `measurement` subsystem already implements functions to manage `Measurement`, `Zone` and `Position` objects that are the basic concepts of the ILONA System. The `measurement` subsystem stores the objects in a MySQL database that is separate from the `navigation` subsystem. The `navigation-persist` component contains the ILONA ontology and provides methods for querying. The separation of the behavior and the implementation of storage facilitates the development and maintenance of the source code.

B. *Test Results*

The `navigation` subsystem was tested automatically and case studies were performed for demonstrating its usability. Unit tests are used to verify the behavior of the software components. In addition, they facilitate the developers' work and can be considered as a helpful complementary documentation. Based on de facto industrial standards, a component is well tested if at least 80% of the source code is covered. Finally, unit testing allows the evaluation of the component separately.

The automated tests covered the `model`, `persist-ontology`, `service-impl` and `controller` modules that contains the implementation. Table II shows the code coverage of the tests. The `model` and `controller` modules were entirely covered by the tests. The code coverage of the `persist-ontology` and the `service-impl` modules exceeded 96%, so it also satisfies the common industrial requirement that is around 80%. Although the automated tests cannot prove the validity of the implementation, they can detect failures. Due to the high code coverage, the validity of the implementation is accepted.

C. *Experimental Results*

The `navigation` subsystem was tested manually with real life scenarios in order to check its functionality. The tests were performed

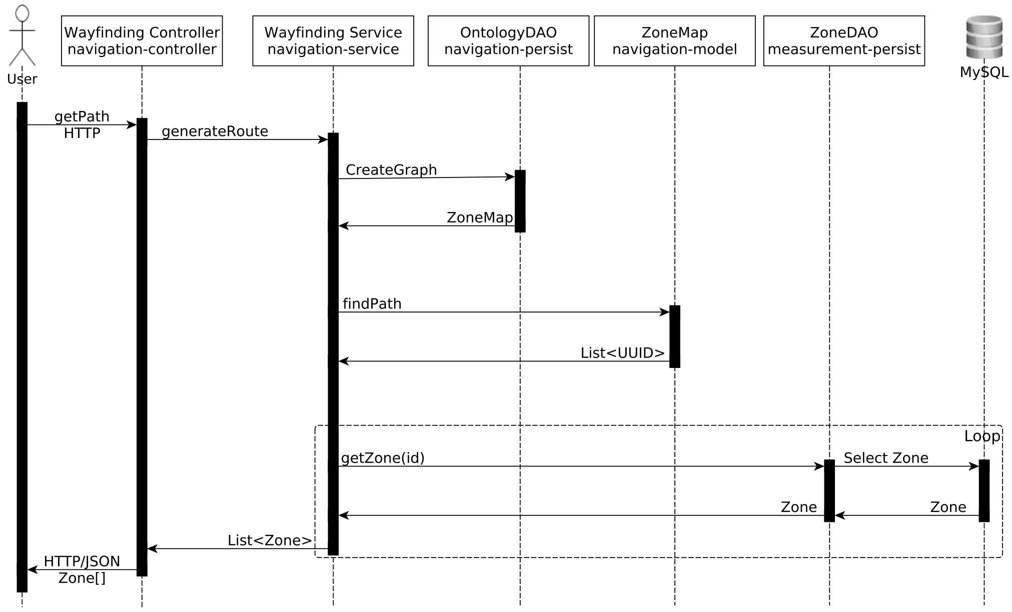


Fig. 6: Sequence diagram of the way finding process

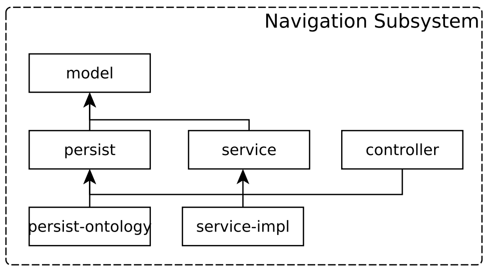


Fig. 7: Modules of the `navigation` subsystem

TABLE II: Code coverage of the `navigation` subsystem

Module	Coverage (%)
`model`	100
`persist-ontology`	97.8
`service-impl`	96.2
`controller`	100

in a three-story educational building that has both stairs and an elevator. Two case studies present the route between the entrance and Lecture Hall 205. The destination is located on the second floor in the rear of the building. The two scenarios are distinguished by the use and unuse of restrictions. During the way finding process the goal was to minimize zone transitions. Experimental results confirm the applicability of the `navigation` subsystem in real life scenarios.

Figure 9 shows the JSON messages that are sent between the server and the client during the experiments. Figure 9a shows the client request with the source and destination zones and with the given restriction. In Figure 9b we can see the server response with the suggested route, that is the list of zones leading from the start to the destination zone. The difference between the two cases is highlighted with italic text. As can be seen in Figure 9a, the `restriction` attribute of the request object is an empty array in the first case while it contains the *no_elevator* string in the second case. When the restriction is set, the response object contains the `1st Floor Lobby` zone object too.

Without restrictions the system generated a route that used the elevator in the hall. First, the route leads from *Ground Floor Lobby* to *2nd Floor Lobby*. Then, the user shall go through *2nd Floor West Corridor* where the door of *Lecture Hall 205* is placed.

In the case when the *no_elevator* restriction was given, the generated route led through the stairs without the use of the elevator. So in the response, there is an additional zone whose name is `1st Floor Lobby`. These results meet with our expectations and fit the shortest path in the building whose floor plan can be checked in Figure 10.

IV. DISCUSSION

Extension of the ILONA System with an ontology-based indoor way finding function was presented in this paper. The extension allows zone level route generation that can facilitate the navigation within indoor environments. In addition, the presented way finding algorithm is integrated to the `navigation` subsystem of the ILONA System that is an open source indoor positioning and navigation framework. Thus, the presented method can be the base of indoor navigation solutions.

The ontology was designed to match the data model of the ILONA System and it is capable of modeling an indoor environment. The ontology defines five different kinds of gateways that allow the modeling of the connections between the zones. Distinguishing the gateways is necessary for disabled people who may want to avoid certain types of gateways. ILONA Ontology is used to create the knowledge base that represents the structure of buildings. The application of ontology reasoners facilitates the performing of complex queries and allows the extraction of various views of the indoor environment. The queried model was used to convert the task of way finding into finding the shortest path in a graph.

The ILONA System was used to implement the way finding algorithm. The implementation was added to the `navigation` subsystem that is broken down to components based on their functionality.

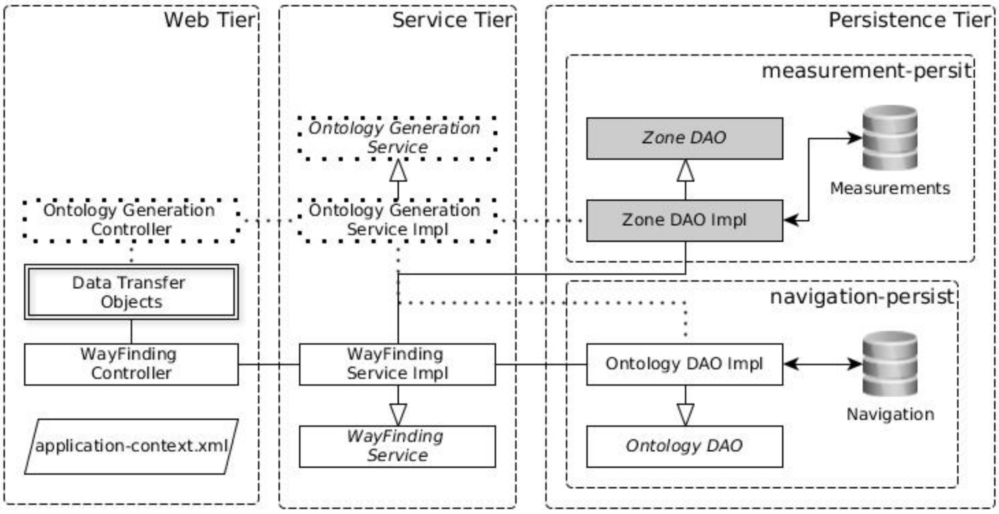


Fig. 8: Components used by the `navigation` service


```
{"sourceID ": "25f79341-7256-4871-a2d5-2008ef88fd8e",
" destID " :" fa162e02-b404-48c7-bc02-8469dbb1e052",
" startName " :" Ground Floor Lobby ",
" destinationName ": " Lecture Hall 205" ,
" restriction ": ["no_elevator"]}
```


(a) Request body


```
[{"id " : "25f79341-7256-4871-a2d5-2008ef88fd8e",
"name ":" Ground Floor Lobby "} ,
{"id " :" d93e647e-4d17-4c3b-9d4c-dd8b88e4523c",
"name": "1 st Floor Lobby "} ,
{"id " :" "d93e647e-4d17-4c3b-9d4c-dd8b88e4523c",
"name ": "2nd Floor Lobby"} ,
{"id " : "4adae318-49b6-42fb-a6a5-0eb2f0efd2f8",
" name ": "2nd Floor West Corridor "} ,
{"id " :" fa162e02-b404-48c7-bc02-8469dbb1e052",
"name ":" Lecture Hall 205"}]
```


(b) Response body

Fig. 9: JSON messages during the experiment

Both automatic and manual tests were used against the components and the `navigation` subsystem. Due to the high code coverage of the tests, the implementation of the way finding algorithm can be considered as verified. The experiments were performed in a three story building and the goal was to generate a route between the entrance and a room placed in the rear of the building. Routes were generated with and without restrictions and the yielded results matched our expectations. Hence, experimental results confirm the applicability of the way finding algorithm.

The presented ontology-based indoor way finding algorithm can be used as a reference in the ILONA System during the development of novel methods. Future investigations should focus on the time complexity of the way finding method, and the modeling of bigger environments in order to examine the limits of the presented way finding algorithm.

ACKNOWLEDGEMENTS

The described article was carried out as part of the EFOP-3.6.1-16-2016-00011 “Younger and Renewing University – Innovative Knowledge City – institutional development of the University of Miskolc aiming at intelligent specialisation” project implemented in the framework of the Szechenyi 2020 program. The realization of this project is supported by the European Union, co-financed by the European Social Fund.

This research was supported by the European Union and the Hungarian State, co-financed by the European Regional Development Fund in the framework of the GINOP-2.3.4-15-2016-00004 project, aimed to promote the cooperation between higher education and industry.

REFERENCES

[1] Christos Anagnostopoulos, Vassileios Tsetsos, Panayotis Kikiras, et al. Ontonav: A semantic indoor navigation system. In *1st Workshop on Semantics in Mobile Environments (SME05), Ayia*. Citeseer, 2005.

[2] Patrick M Dudas, Mahsa Ghafourian, and Hassan A Karimi. Onalin: Ontology and algorithm for indoor routing. In *2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 720–725. IEEE, 2009.

[3] Erich Gamma. *Design patterns: elements of reusable object-oriented software*. Pearson Education India, 1995.

[4] Tom Gruber, In Ling Liu Ontology, and M Tamer Özsu. Encyclopedia of database systems. *Ontology*, 2009.

[5] Dominik Heckmann, Tim Schwartz, Boris Brandherm, Michael Schmitz, and Margeritta von Wilamowitz-Moellendorff. Gumo–the general user model ontology. In *International Conference on User Modeling*, pages 428–432. Springer, 2005.

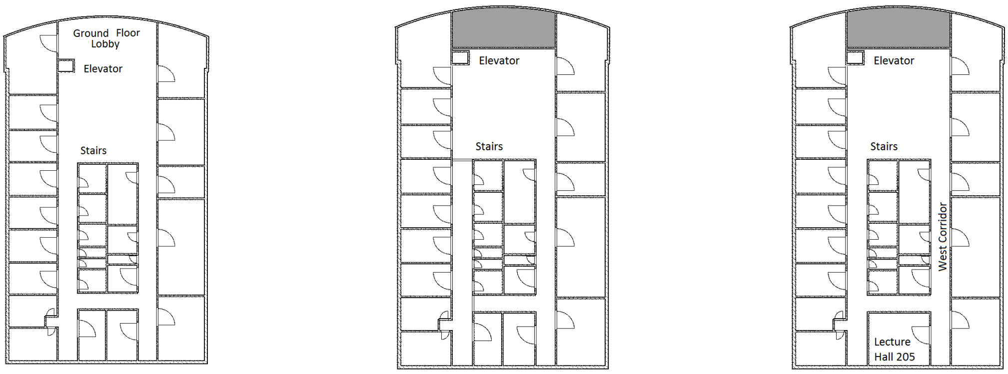
[6] Matthew Horridge and Sean Bechhofer. The owl api: A java api for owl ontologies. *Semantic Web*, 2(1):11–21, 2011.

[7] K. Ilku and J. Tamas. Indoorgml modeling: A case study. In *2018 19th International Carpathian Control Conference (ICCC)*, pages 633–638, May 2018.

[8] Daniel P Kun, Erika Baksane Varga, and Zsolt Toth. Ontology based navigation model of the ilona system. In *Applied Machine Intelligence and Informatics (SAMI), 2017 IEEE 15th International Symposium on*, pages 479–484. IEEE, 2017.

[9] Steffen Lohmann, Stefan Negru, Florian Haag, and Thomas Ertl. Vowl 2: user-oriented visualization of ontologies. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 266–281. Springer, 2014.

[10] Boris Motik, Peter F Patel-Schneider, Bijan Parsia, Conrad Bock, Achille Fokoue, Peter Haase, Rinke Hoekstra, Ian Horrocks, Alan Ruttenberg, Uli Sattler, et al. Owl 2 web ontology language: Structural specification and functional-style syntax. *W3C recommendation*, 27(65):159, 2009.



(a) Ground floor (b) First floor (c) Second floor

Fig. 10: Building floor plans

[11] Boris Motik, Rob Shearer, and Ian Horrocks. Hypertableau Reasoning for Description Logics. *Journal of Artificial Intelligence Research*, 36:165–228, 2009.

[12] Mark A Musen. The protégé project: A look back and a look forward. *AI matters*, 1(4):4–12, 2015.

[13] A. Satan. Bluetooth-based indoor navigation mobile system. In *2018 19th International Carpathian Control Conference (ICCC)*, pages 332–337, May 2018.

[14] Graeme Turnbull Stevenson, Juan Ye, Simon Andrew Dobson, and Paddy Nixon. Loc8: a location model and extensible framework for programming with location. *IEEE Pervasive Computing*, 2010.

[15] Zsolt Tóth, Péter Magnucz, Richárd Németh, and Judit Tamás. Data model for hybrid indoor positioning systems. *PRODUCTION SYSTEMS AND INFORMATION ENGINEERING*, 7:67–80, 2015.

[16] Zsolt Tóth. Ilona: indoor localization and navigation system. *Journal of Location Based Services*, 10(4):285–302, 2016.

[17] Xin Wang, Jianga Shang, Fangwen Yu, and Jinjin Yan. Indoor semantic location models for location-based services. *Int. J. Smart Home*, 7(4):127–136, 2013.

[18] Lei Yu, Yang Lu, and XiaoJuan Zhu. Smart hospital based on internet of things. *Journal of Networks*, 7(10):1654–1661, 2012.

Dániel Péter Kun has a Bachelor's Degree in Business Informatics and a Master's Degree in Software Engineering specialized in Application Development. For two years, he had been working on the ILONA System as a developer. His contribution to the navigation module of the ILONA System was valuable.

Erika Baksáné Varga is an associate professor at the Department of Information Technology in the University of Miskolc. She has MSc in Information Technology and Engineering Economics. Her research areas: ontology, natural language processing, data modeling, management and analysis, data- and knowledge intensive systems. Title of her PhD dissertation: Ontologybased semantic annotation and knowledge representation in a grammar induction system (2011).

Zsolt Tóth is a Lecturer in the Institute of Information Sciences at the University of Miskolc. He is also a member of the IEEE Computer Society.
He received his PhD in Computer Science from the University of Miskolc. His research interests include indoor positioning and navigation techniques, applications of data mining, neural network, software development, web development and distributed systems. He has designed and supervises the development of the ILONA System and leads a research group focused on indoor positioning and navigation. He has over 30 publications.