# IBM Research Report

# Depiction of Physical Space Using Node Relationship Graphs and Subsequent Way-Finding Over the Internet

**William S. Carter**
IBM Research Division
Austin Research Laboratory
11501 Burnet Road
Austin, TX  78758
USA

# DEPICTION OF PHYSICAL SPACE USING NODE RELATIONSHIP GRAPHS AND SUBSEQUENT WAY-FINDING OVER THE INTERNET

## *William S. Carter*
wscarter@us.ibm.com

**IBM**

Accessibility Center
11501 Burnet Road
Austin, TX 78759

## Problem Addressed

Many attempts to represent physical spaces as data focus on capturing information that can be used to generate pictorial maps. This data may not be sufficiently comprehensive to allow for algorithmic way-finding, for the construction of representations that work well on small or tiny screens, or for convenient delivery of information in alternative formats that are accessible for people with disabilities. Here we propose that an abstraction of physical space can be created using Node Relationship graphs which mimic natural hierarchy and structure. A web implementation was created for testing the hypothesis, and a turn-by-turn way-finding algorithm was successfully written in JavaScript which recursively submitted web queries to a server containing the modeled data.

## Background

In the present era there is quite a bit of activity under way to provide for indoor location-finding technologies that rival the accuracy and ubiquity of outdoor GPS services. Ideally the map presentation and way-finding capability would mesh into a seamless transition as one moves from the outdoors into a building or other interior venue where GPS does not function. Knowing your current interior location is useful in itself, but you will also want information about other nearby interior places and how to walk to them. This requires a data representation of the space and the inherent connectivities, and ideally the assembling of the data and the subsequent consumption of it will be convenient for web applications. Several efforts in this direction are shown in the references. One notable example is IndoorGML, which makes use of Node Relation graphs and has developed a complex XML specification.

## Strategy

Many proposed industry standards have failed to take root in the past due to an overly complex specification. We have attempted to confine our data representation to a minimal subset of

features that would be required in order to capture a sufficient number of ordinary physical scenarios and still be understandable for the typical web application developer.

Graphs are a staple of computer science and the processing of them is well understood. Fortunately, physical spaces usually take the form of hierarchical structures that map well to a graph representation all the way from the macro level to the immediate personal vicinity. The Earth contains continents, the continents have countries, countries have states and provinces which contain counties and cities. Buildings have floors connected by stairs and elevators, the floors contain rooms which are connected by hallways. Each of these represents a "level of granularity" whereby there is a set of entities enclosed all in the same container, have spacial relationships with each other, and potentially have interconnecting navigable routes between them. Our strategy has been to model these features as a node relationship graph where "Points of Interest" are the nodes and the interconnections are graph edges which are called "Connectors" in our lexicon.

A **Point of Interest** is an arbitrarily defined volume of space which may or may not be delineated by physical barriers such as walls. A **Connector** is a Point of Interest that has a 2-dimensional axis on a directional line (such as east-west), and to which Points of Interest and other Connectors can be sequentially attached like beads on a string. Points of Interest and Connectors can be designated as members of one or more arbitrary **Categories** (office, restroom, hallway, elevator, etc). Points of Interest and Connectors can be **related** in specific ways. In our system there is the **Name** of the relationship, and **Source** and **Target** designations to specify which has the named relationship with the other. For example the 5th floor of a building could be a Point of Interest which has the "**Contains**" relationship with all the rooms on the floor. The floor is the Source in this relationship, and each room is a Target. Additional relationship types can easily be created and assigned as needed.

We implemented these data entities as Java objects in an application running on a web server, and designed and implemented an API consisting of a set of REST HTTP queries with JSON responses. A JavaScript browser application was written using the Dojo Toolkit that allowed for constructing the physical model as a tree of Points of Interest 'containers' with drag-and-drop nodes consisting of child Points of Interest and Connectors. Categories can be created at will and assigned to the POI's and Connectors by clicking on a list. Connectors are populated by displaying a list of POI's and Connectors within the same container, clicking on items to 'connect' to them, and dragging the items to the appropriate position in the sequential connection order if needed.

A route-finding algorithm was written in JavaScript whereby a start point and an end point could be designated in the GUI and all of the possible success paths between them are recursively investigated via web queries and discovered. Provision is made for assigning distances between POI's and Connectors if known so that a shortest path can potentially be identified.

## Major Difficulties

The amount of information that can be acquired and stored using this technique can quickly become very large, and it soon becomes a challenge to present to the user of a browser application only the appropriate subset of the universe that will fit onto the screen. The entering of the Point of Interest and Connector details and their relationships is a somewhat tedious data entry task, and ideally there would be a way to automate this to some extent using existing maps or blueprints.

## Outcomes

In our experience the depiction of physical space as a set of node-related entities is surprisingly intuitive and useful. Most people grasp the concept easily, and the underlying body of data lends itself well to a variety of presentation forms ranging from textual descriptions to graphical

embodiments. The technique also appears to capture the key elements required for way-finding such that turn-by-turn route solutions can quickly be found automatically using ordinary graph-traversal algorithms familiar to Computer Science college students. While our technique may not easily yield pictorial maps due to lack of specifics as to the placement of walls and doors, it does allow for easy creation of "schematic" representations of spaces and routes that can be rendered on small screens and comprehended at a glance while walking.

## Open Research Avenues

Continued refinement is needed in order to more completely identify the most appropriate minimal set of information that should be acquired and recorded for depicting physical spaces while retaining the feasibility of entering the information by hand if necessary. For example, in our current schema there is no depiction of which side of a hallway a room will appear.

As one attempts to determine the names of interior connections such as corridors and stairways it quickly becomes apparent that many of these do not actually have assigned names. In these cases some appropriate convention needs to be developed that makes sense. Will "interior westernmost north-south main corridor" be appropriate?

It is likely that crowd sourcing can be an important mapping resource, but there may be problems with data reliability or even data vandalism if precautions are not taken.

## Acknowledgements

## References

**1.** Open Geospatial Consortium, http://www.opengeospatial.org/projects/groups/poiswg
**2.** IndoorOSM, http://wiki.openstreetmap.org/wiki/IndoorOSM#Relations
**3.** IndoorGML - Candidate Standard for Indoor Spatial Information, http://indoorgml.net/