U-Access:
A Web-Based Spatial Decision Support System
Routing Pedestrians of Differing Abilities

Adam D. Sobek and Harvey J. Miller

Department of Geography
University of Utah

260 S. Central Campus Dr. Room 270
Salt Lake City, UT 84112-9155

adam.sobek@geog.utah.edu
harvey.miller@geog.utah.edu

Abstract: The combination of the limited mobility and inaccessible built environments positions disabled individuals at risk of exclusion from the opportunities and activities that many take for granted. While a trip to a store maybe a routine activity for a peripatetic (traveling on foot) person, for a people who uses a wheelchair, minor features of the built environment such as a curb or a small step can prevent access. Therefore, there is a need to assist people with disabilities in navigating urban environments by providing them with personalized *accessible* routes. U-Access is a web-based decision support systems for assisting pedestrians of differing abilities to navigate through urban environments. U-Access couples the strengths of cutting edge technologies including GIS, Java2® and scalable vector graphics (SVG)®. This system uses ESRI® geodatabase technology as a distributed spatial database providing ease of access and updateability of the pedestrian network. Java2® technology is used for fast and efficient network queries on the database. Lastly, SVG provides the interface with the spatial database and maximizes the number of people who can use the application by serving it over the World Wide Web. U-Access allows users to select an origin, a destination, and to choose one of three physical ability levels, namely, peripatetic, aided mobility (using crutches, cane, or walker), or a wheelchair user. Using these inputs, the system returns the shortest accessible route with respect to the user's ability. U-Access is currently being utilized by the Center for Disability Services on University of Utah campus for identifying feasible routes for disabled people.

Keywords: Accessibility, disability, spatial decision support, routing

1. Introduction

Accessibility is an important issue in today's urban and built environments. Urban designers and architects are actively developing new spaces that are easy for people to navigate (Foster 2004). However, retrofitting existing urban environments to assist people with disabilities in compliance with legislation has proven to be difficult. For example, in attempt to facilitate ease of movement in urban areas, municipalities have installed curb cuts at major intersections. While this effort has made it easier for wheelchair users, it in turn made it more difficult for the visually impaired person to identify where the sidewalk ended and the street began. While much work is being completed in the planning realm (Franklin 2005), there is a need for furthering research to assessing the accessibility of pedestrian networks and assisting people with disabilities in navigating urban environments. This research develops a methodology and a web-based system to evaluate current discrepancies within the network and at the same time assists pedestrians of differing physical abilities by providing them with the shortest feasible route.

U-Access is a World Wide Web-based spatial decision support system that allows users to obtain optimal pedestrian routes through a built environment with respect to their individual abilities. Three abilities levels recognized by U-Access are: i) *peripatetic*, meaning walking on foot without assistance, ii) *aided mobility*, meaning requiring the assistance of a cane, walker, or crutches, iii) *wheelchair users*. Similar to other navigational tools (Map Quest® and Map Point®), U-Access allows users to select an origin and a destination. However, in addition users are also able to specify their physical ability level. U-Access uses the physical ability level of the user to distinguish which edges within the pedestrian network are traversable. This ensures that that the route provided is *feasible* and it allows for the comparison of routes of people with different levels of physical abilities.

The next section provides a general background of transportation optimal path routing applications for automobiles and pedestrians. Specifically, the needs for people with physical disabilities are addressed, as well as current practices at the study site. Section 3 provides an examination of spatial decision support systems for routing pedestrians and how they can be coupled with geographic information systems (GIS). The subsequent section outlines the system design for U-Access including a data model, algorithm and the interface design. Lastly, section 5 provides and evaluation of U-Access as both an analytical planning tool and as a routing tool.

2. Background

2.1. Transportation Routing

Transportation routing problems, in the most general terms, attempt to find solutions for optimal routing and location problems within a network (Miller and Shaw 2001). Since the shortest route is constrained by the network, the algorithms for solving for the optimal route must model actual travel conditions for realistic solutions. For example, when modeling traffic, algorithms should account for one-way streets, traffic signals, and congestion (Sheffi 1985). Commonly used web vehicle routing applications such as MapQuest® (www.mapquest.com) and MapPoint® (http://mappoint.msn.com) model some phenomena such as one-way streets, however, they are limited in that they only utilize distance or time as an impedance (or the cost to traverse an edge).

While these web-based automobile routing applications have been useful for their designed purpose, as shown by their popularity, they can not be directly translated for use with pedestrians. While automobile routing applications provide minivans with the same routes as mid-size trucks, to assume that all people can traverse all pedestrian networks is not realistic. Pedestrian environments are comprised of such built structures such as stairs, curbs, and steep slopes that can hinder or even prevent physically disabled people from participating in social or economic interactions. Therefore, it is essential to incorporate a user's physical ability when routing through built environments to ensure that an individual can successfully traverse the provided path.

2.2. Routing Pedestrians of Differing Abilities

In 1990, the United States government recognized a need to define urban environments in order to allow equal access to all people. The United States Congress took action by passing the Americans with Disabilities Act (ADA). The main goals of the ADA are to provide people with disabilities access to buildings, equal employment opportunities, equal access to public transportation, the opportunity to attend school and the chance to be eligible for social security support (Little 1995). Section 4.3.2 of the ADA states that "at least one accessible path within the boundary of the site shall be provided from the street…" This view of absolute accessibility is similar to that of automobile routing in that people with or without disabilities *can* traverse through the accessible path. Church and Marston (2003) add insight by proposing an approach which might enhance the overall accessibility of places. They suggest a relative measurement of accessibility that is sensitive to both the number of routes and the values of each accessible route in conjunction with the traditional measurement of absolute access. The point is well made by Church and Marston (2003) that society should not categorize people as merely disabled or not disabled; rather, there are degrees of abilities.

2.3. Current Practices at Application Site

In response to the ADA, the University of Utah established the Center for Disability Services (CDS). At present, if an individual has a disability, they can inquire at the CDS for information about navigating the university campus. The CDS uses a combination of paper maps and expert knowledge to individuals in finding optimal routes between campus origins and destinations. In some cases personal accompaniment is necessary to ensure that the individual arrives at their desired destination. This procedure has several drawbacks. First, it is both difficult and costly to maintain paper maps in a timely manner. Second, this method relies heavily on the availability and knowledge of experts which increases the time required to generate optimal routes and limits the ability to transfer information within the university community. A third consideration is the lack of data. The current paper maps identify stairs; however, they do not include other obstructions that may hinder or prevent a disabled person from passing, such as curbs and steep slopes. Finally, even though a paper map could be created to include all previous factors, Satalich (1995) shows that even people who have studied a map well are likely to make large errors in route estimations. The development of a route-finding application that incorporates all the factors necessary for navigating pedestrians of varying ability levels through urban environments could save a great deal of time and resources.

3. GIS and Spatial Decision Support for Routing Pedestrians of Differing Abilities

3.1. Pedestrian Routing Applications in a GIS

There are several constraining factors in determining the correct and feasible paths of travel for people with differing abilities. The first and most obvious is distance. The path returned to the user should be the shortest route with respect to network distance between origin and destination pairs. The second factor is time. Sometimes the shortest route is not always the fastest. Often times a vehicle can arrive at a destination faster by using freeways, though the overall distance may be greater. The third constraining factor to routing pedestrians is the physical ability of the user. Knowing the ability of the user is the essential consideration in determining if a route is feasible. Barriers that may be imperceptible to many people may hinder or totally restrict access to people with disabilities (Matthews and Voujakovic 1995). Examples include uneven pavement slabs, cobblestone courts and gravel. Consequently, there has been a move towards utilizing *geographic information systems* (GIS) and *global positioning systems* (GPS) to assist disabled people in navigating through urban spaces (Dewey 2001, Golledge *et al.* 1991, Golledge *et al.* 1998, Matthews *et al.* 2003).

Matthews *et al.* (2003) and Dewey (2001) developed GIS routing applications that are similar in their system design and functionality. Their applications use ESRI's® ArcView geographic information system (GIS) and customized using Avenue (ArcView's scripting language) and Java respectively. Both applications provide wheelchair users with detailed, customized information to assist them in planning and managing their mobility within urban environments. Unfortunately, these applications are not widespread due implementation difficulties.

There are two major costs involved with the implementation of Matthews *et al.* (2003) and Dewey (2001) applications. The first is found within the data acquisition. The applications are design to utilize extremely accurate data including the measurement of all bumps and abnormalities in the sidewalk. This would be extremely costly for any municipality to acquire and maintain. The second cost is found in the software. The designers the previously mentions applications deliver their solution via ArcView 3.x®. Therefore only people who can afford the commercial GIS would be able to use such and application. Costs are not the only difficulty with the current applications; there are also two fundamental implementation issues, namely, the learning curve to master the GIS software and again, the level of detail chosen for their applications. Learning the tool and terms within a GIS can be very difficult and too time intensive for most people and therefore limits who can use the created applications. Also, the issue of the volume of data necessary to run the applications is far too intensive. Again, this is not only costly, but it also requires a great deal of time from the users. The applications require a plethora of variable to be entered before find the optimal route. Such variables include things like wheel diameters, how good of shape the user is in, and the maximum height of largest obstacle the user is able to traverse. The data development for these applications is a monumental task for any municipality to undergo and therefore limits the practicality of such applications.

U-Access is a more universal tool than its predecessors. As the World Wide Web (WWW) is increasingly becoming the common medium for both the transmittal of aspatial and spatial information, it provides a degree of accessibility to the public that proprietary software packages cannot offer. U-Access capitalizes on emerging technologies that are increasing the overall accessibility of the WWW (W3C 2000a), Scalable Vector Graphics (SVG). SVG provides optimal display functionality by means of vector technology and offers a set of powerful querying tools which are based on eXtensible Markup Language (XML). As a result, U-Access is able to generate fast, correct paths of travel for all people who have access to the Web. This application also capitalizes on the speed and efficiencies of Java's object oriented technology for a 'back-end' optimal route computation. This elegant system design for a WWW-based pedestrian routing service provides an efficient, scalable, and robust application that provides both planners and users the ability to identify feasible routes through urban environments in a cost effective and timely manner.

3.2. System Design

3.2.1. Data Model

Spatial network databases (SNDB) form the kernel of many network applications such as transportation planning, air traffic control utilities, river transportation and irrigation canal management (Shekhar and Chawla 2003). Unlike traditional spatial databases that store objects based on their spatial proximity, SNDB are based on both proximity and connectivity. This shift requires new and different data storage techniques for data modeling, query languages, and indexing. This section will focus on three levels of data modeling, namely, conceptual, logical and physical.

At the conceptual level of data modeling, all the available information related to the application is organized using a high-level semantic modeling technique. The conceptual modeling process focuses on data types, their relationships and their constraints. Figure 1 shows a Universal Modeling Language (UML) diagram of the essential environmental elements and their relationships, as recommended by the Open GIS Consortium's (OGC) "Abstract Specification, Topic 5 Factors" (OGC 1999). There are three relationships worth noting in this UML diagram. The first is the "Individual's Physical Ability Level." In U-Access, an individual must define their physical ability level as a *wheelchair user*, *aided mobility*, or *peripatetic*. The pedestrian environment is then constructed by incorporating sidewalks, handicapped entrances, handicapped parking, ramps, and/or curb cuts into the data model. Lastly, in the center of Figure 1, a one-to-many relationship exists that reflects the interaction between an individual and urban structures.

At the logical level, SNDBs are expressed within the framework of graph theory. The basic operations used within graph theory are based on the three fundamental subclasses, namely, *graph*, *vertex*, and *edge*. The *graph* class must be able to add, delete, or return a vertex of an edge given two vertices, return an adjacent node of a given vertex, find all adjacent neighbors, and finally, return the parent of a given vertex. The *vertex* class necessitates four basic operations, namely, the creation of a vertex with the appropriate label, returning a label associated with a given vertex, and marking a vertex as having been visited. The *edge* class is comprised of a constructor and has three functions, namely, to return the first node of the edge, to return the end node of the edge, and to return the length of the edge. Although this is not an exhaustive list of the operations for a SNDB, the necessary elements needed by many graph algorithms are included, such as connectivity and shortest path (Shekhar and Chawla 2003).

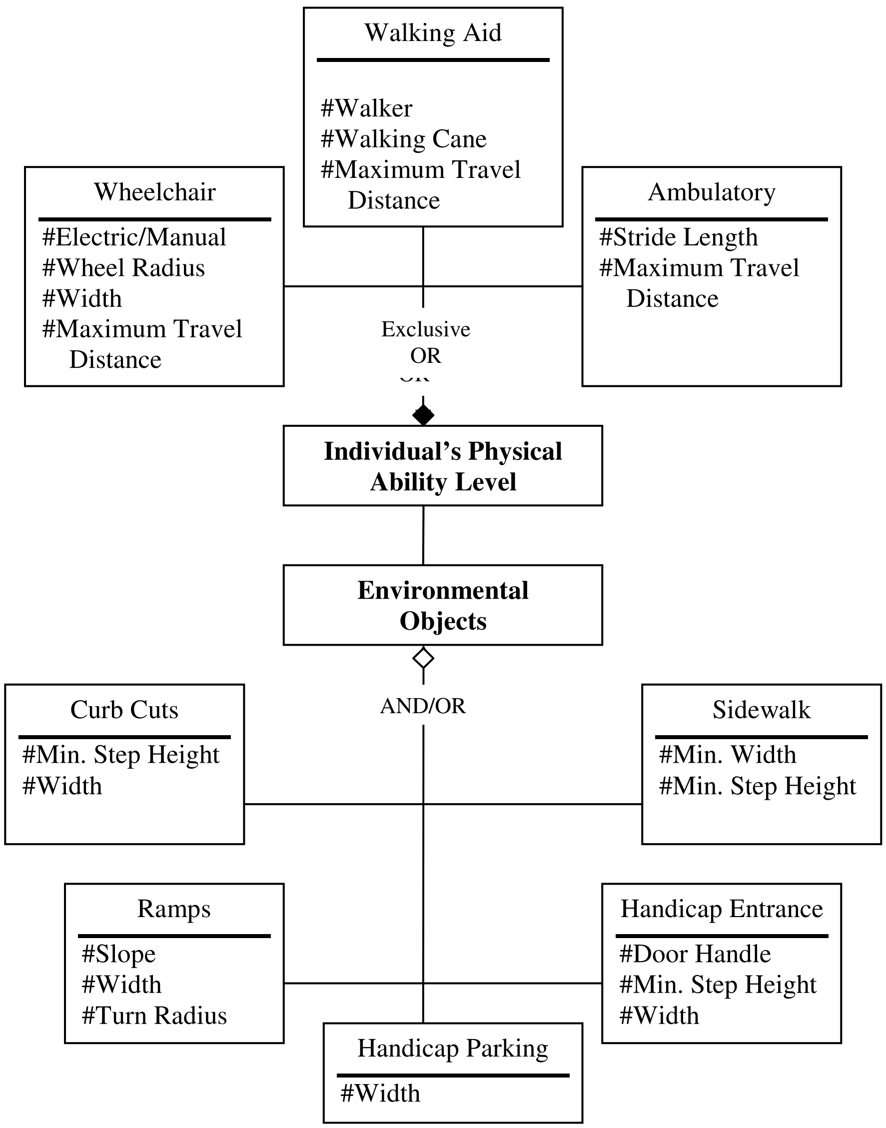
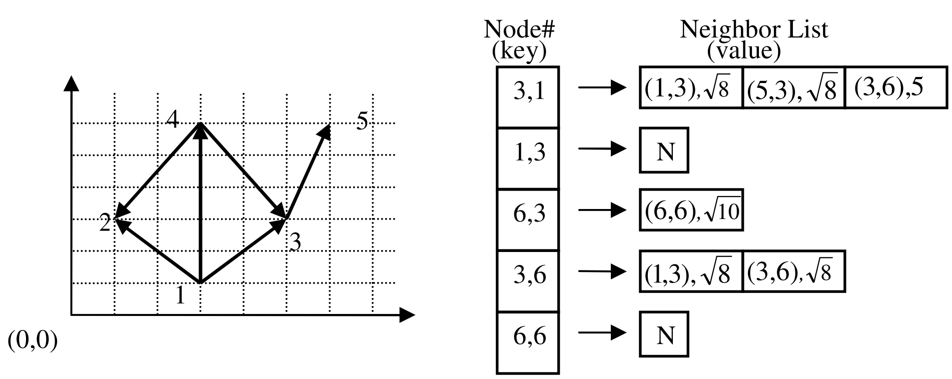


Figure 1 UML diagram of the essential model used for U-Access.

The last aspect of network modeling to discuss is the physical storage of the data. Since U-Access is a Web-based application, the speed at which solutions are generated is of critical importance. Therefore, U-Access uses the Java® hashmap data structure to represent the pedestrian network (see Figure 2). Java program parses the SVG file and creates three Java objects, namely, a list of unique nodes (nodeList.map), a list of unique edges (edge.map), and a neighbor object for each node (neighbor.map). Nodes are zero dimensional objects defined by two numbers representing the x and y coordinates. Edges are constructed using a start node, an end node, and a distance value representing the length of the edge. The neighbor objects are additional sources of information that are constructed in order to decrease the time necessary to compute the optimal path. A neighbor object is composed of a node object and an integer value representing the distance (the distance of the edge which connected the neighboring node). The neighbors are stored in a *hashmap* or aggregate categories based on some key or unique attribute. In this case, the key is a unique node, and the value is the list of neighbors for that node.



(a) Directed Network (b) HashMap Representation

Figure 2 Database representation of a pedestrian network.

3.2.2. Implementing the Data Model

The two main goals of the data model design for U-Access are that the spatial data are easily updateable and that route computation is fast and accurate. In order to accomplish these goals, there are four major steps in transforming the data from a GIS to presenting pedestrian users with optimal routes (see Figure 3). The first step includes the creation and maintenance of the spatial information within a commercial GIS. The second step is the transformation of the spatial dataset from the GIS to an SVG file format. In the third step, the SVG is parsed into three unique datasets representing three ambulatory abilities. Finally, the SVG file is embedded in an HTML file and served to the university community via the World Wide Web.

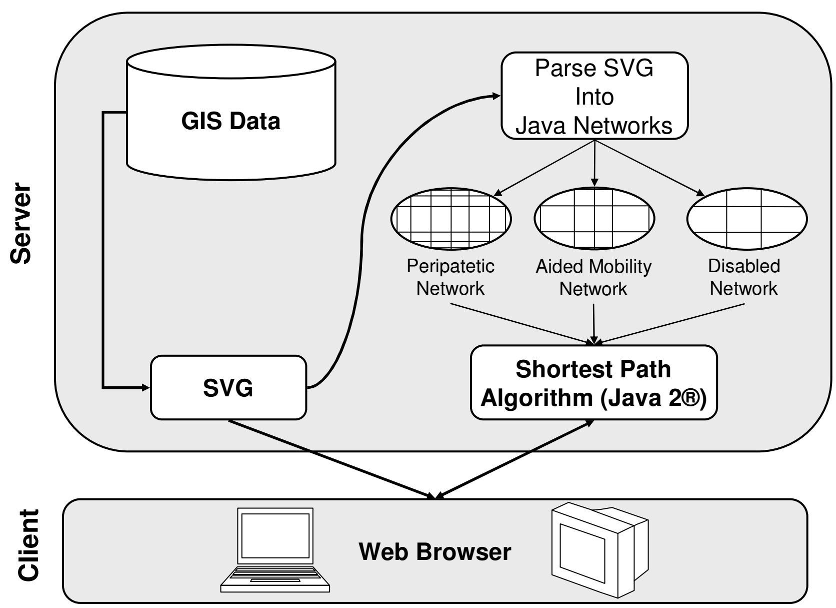


Figure 3 Data flow from commercial GIS to World Wide Web

Spatial data acquisition and creation is the most critical step in the development of a successful web mapping application. ESRI's® ArcGIS environment was chosen for several reasons. It is able to integrate and update spatial data from several different formats. It is also able to directly transform the data from a shapefile format to a SVG file format. Spatial information was obtained from several sources including global positioning system (GPS) to reference stairs and curb cuts, general campus information from the University of Utah Facilities Management department, and accessible building entries from the Center for Disability Services (CDS). The pedestrian network is represented by a network dataset with each line given a mobility index of 1, 2 or 3. A value of one is given to all edges that only peripatetic people can traverse; these include flights of stairs and non-accessible building entries. An index of two is given to urban obstacles such as curbs or locations where there may be steps in the pedestrian network. Lastly, an index of 3 is given to all edges that wheelchair users can navigate. Prior to the data being exported into SVG file format, the maps were presented to the CDS for data validation and verification.

The second step in the data model is the modification of the spatial information from ESRI's proprietary data format to the SVG Extensible Mark-up Language (XML). This was accomplished using a third-party open-source extension called GeoClient®. There are several advantages for using the SVG file format. SVG, as an XML format, incorporates both attributes and spatial data within a single file, and therefore can support both textual and geographical queries. Secondly, since SVG is a vector file format, the display properties are independent of the users' screen size and resolution (see Plewe 1997). Lastly, the seamless integration of SVG and JavaScript® creates a robust, scaleable, and effective Web environment that incorporates a full suite of tools for data querying and optimal path routing.

The third step in the data transformation process uses the geographic information within the SVG file to create three SNDBs which allows for efficient search and retrieval of locational information for the shortest path algorithm. The extraction of the pedestrian networks uses a Java® program that parses the SVG file into three pedestrian networks, namely, i) peripatetic routes, ii) aided mobility routes, and iii) wheelchair user routes. This method of storing redundant data is used in order to minimize the number of edges that the shortest path algorithm considers, and therefore minimizes the optimal route computation time.

The fourth and final step in the data transformation process is to embed the SVG file into an HTML document. This enables the integration between the SVG objects and the stored Java SNDB. JavaScript is used to communicate between the two different datasets by passing five parameters from the SVG document, namely the origin coordinates, the destination coordinates and an ability level. The JavaScript then parses in the list of edges that identifies the shortest feasible route that is returned from the Java shortest path applet and highlights them on the map.

3.2.3. Algorithm Design

At the heart of all optimal path applications are algorithms for solving the routing and location problems within a network. Although there are several versions of the basic shortest path algorithm, the two main contributors were Bellman in 1958 and Dijkstra in 1959 (Goodrich and Tamassia 1998). All shortest path algorithms use the same fundamental operations; however, they differ with respect to physical implementation, or how the data structure is designed. (see Cherkassky et. *al.* 1993, Gallo and Pallottino 1998, Goodrich and Tamassia 1998, Miller and Shaw 2001). This section examines the implementation of Dijkstra's shortest path algorithm with respect to the above mentioned data storage structure.

The high-level description of Dijkstra's shortest path algorithm is deceptively simple. There are four temporary storage elements to the algorithm. The first stores the best estimate of the shortest distance from the source to each vertex, d. The second temporary storage is P, also known as the parent or predecessor tree, which stores the predecessor of each vertex on the shortest path from the source. The third data set is S, the set of settled nodes, or those nodes whose shortest distances from the source have been identified. Lastly, Q represents the set of unsettled nodes. The three main steps in implementing Dijkstra's algorithm are (1) while Q is not empty, extract the node with the minimum distance (*u*) from Q, (2) add *u* to S, and finally (3) relax the neighbors of *u* (Waldura 2003). When the program "relaxes" a node, it finds all nodes connected to the current node (also known as the forward star structure), calculates their respective distances, and then adds the nodes to a queue of sorted nodes. This insures that Q contains a sorted list of nodes that do not already exist in data set S and the distance is defined in data set d.

While the shortest path applet utilized by U-Access follows the general steps in Dijkstra's algorithm, one enhancement offered in this research is the use of neighbor objects when relaxing neighbors of node *u*. Traditional data storage techniques using the *forward star structure* (FSS) method: this method organizes the SNDB by nodes and the set of arcs leaving each node. This requires three separate data structures: i) an arc array; ii) an arc weight array, and; iii) a pointer array for accessing the two data arrays (Miller and Shaw 2001). Thus, this data structure requires three steps to identify the adjacent nodes to node *u* and their corresponding distances. In contrast, U-Access uses the hashmap data structure to capitalize on an efficient access function within the Java environment. When the path-finding algorithm needs to identify the list of adjacent nodes and their corresponding distances, it simply uses node *u* to reference a list of neighbors. Although the hashmap data structure appears to behave much like the FSS, the number of steps needed to retrieve connected nodes is reduced from three steps to one.

3.2.4. Interface Design

U-Access is designed to provide a clear, straightforward Internet application for inexperienced Internet users. Since one cannot assume that computer literacy is adequate among all individuals (e.g., Matthews and Vujakovic 1995), all choices are made by using mouse clicks on either the 'buttons' or on the map. Key commands are kept to a minimum to ensure clarity and visibility, as well as to guard against mistakes.

U-Access begins with a user loading the Web page. The page consists of four components. The main component is the map that provides the user with a spatial reference and attributes about the data. When a user 'hovers' the mouse over a building, the name is automatically displayed in the center of the toolbar. Within the dashed line in Figure 4, the user is hovering over the "Jon M. Huntsman Center" and thus its name appears in the toolbar.

The second component of U-Access is the locator map. The locator map is a smaller scale map of the entire study area located in the upper right hand corner of the browser window. When the user zooms in on the main map, a box in the locator map highlights the viewable area since it is linked to the scale and location of the main map. This allows the user to easily identify where he/she is relative to the entire study region (see Figure 4).

The third component of U-Access is the tool bar. The tool bar is a series of buttons located at the top of the map which aid the user in navigating through it. Within the tool bar there is a 'zoom-in' function, 'zoom-out,' a search tool that enables the user to search for a building by name, and an identify tool that allows the user to 'identify' features on the map (see Figure 4).

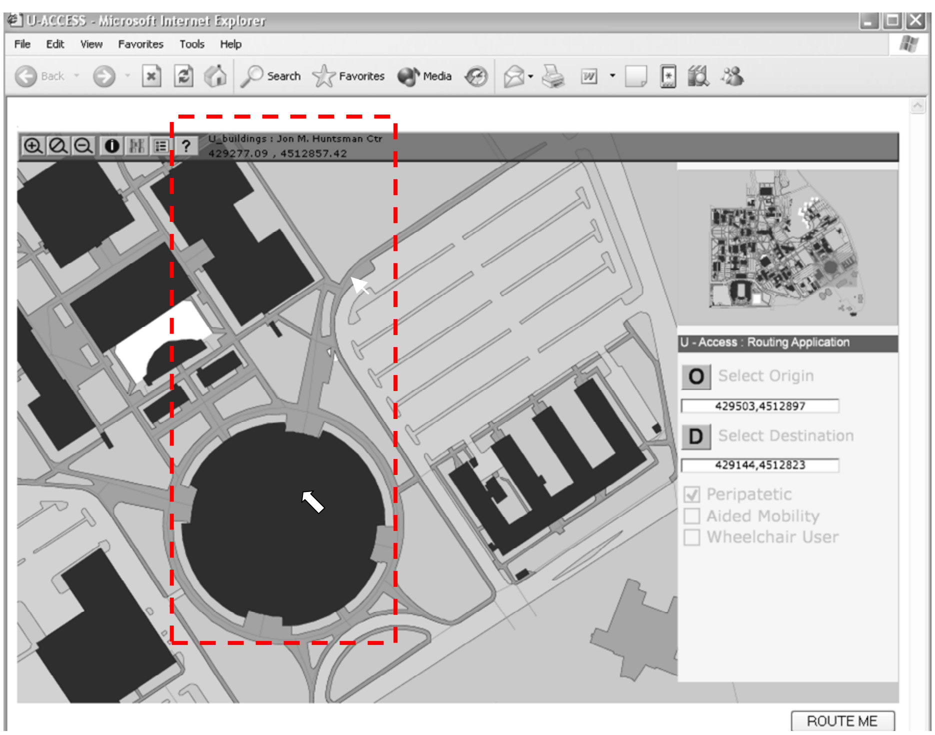


Figure 4 U-Access user interface

The last component of the U-Access interface is the routing tool box. It is a frame under the locater map that allows the user to select an origin, a destination, and an appropriate mobility level. Within the routing tool box there are three buttons as well as a “radio button” which allows the user to select one and only one ability level. The first button activates a tool that allows a user to define their origin. The second button allows the user to define their destination by clicking on the appropriate location on the map. The coordinates of the origin and destination identified by the user are recorded in the text box under the tool. Third, the user must choose an appropriate ability level. This option is selected with radio buttons to ensure that the user only selects one ability level. In this manner, the user is able to personalize their route parameters (see Figure 4). If the user has failed to enter in any of the necessary parameters, a pop-up window appears telling the user to finish supplying the necessary information. Finally, the user selects the “ROUTE ME” button located in the lower right corner of the explore window. Once a user has entered in the necessary data, the user may then select the “ROUTE ME” button to invoke the application to find and highlight the route. Figures 5 and 6 provide a sample display that illustrates the difference in routes provided to users with the same origin and destination but with different physical abilities.

Once the path-finding routine has been invoked by the user, the shortest path algorithm solves for the optimal route. A single Java program executes using the parameters defined by the user. Note that the network is not rebuilt each time the program is run since a separate network is maintained for each ability level. This minimizes computation time. The path-finding algorithm uses the physical ability level input to determine which data set to use.

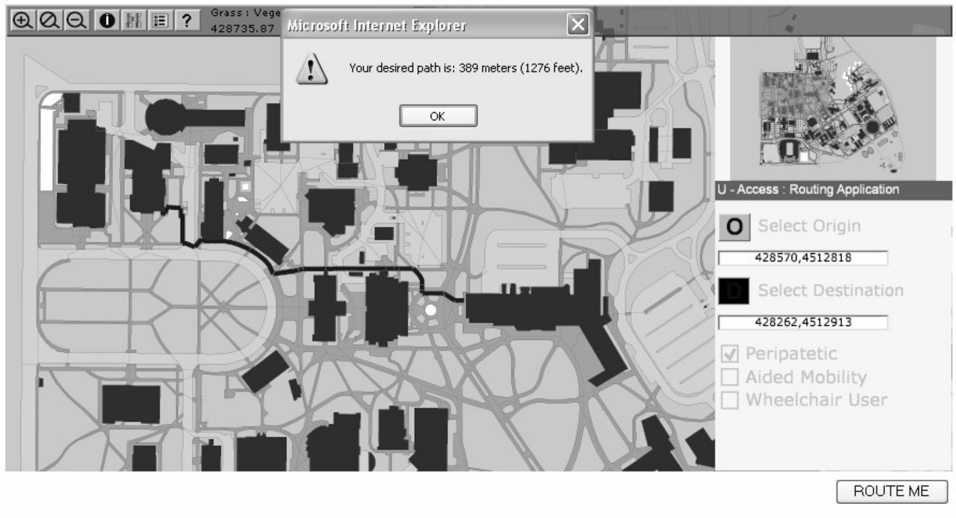


Figure 5 Peripatetic (ambulatory) route between the Union and Kingsbury Hall.

Figure 6 Disabled route between the Union and Kingsbury Hall.

Once the Java program has computed the optimal path, the result is displayed for the user. The route-finding algorithm returns a list of edge features that compose the optimal route. JavaScript parses the string of feature numbers and highlights each edge feature that is a part of the accessible route. The total distance is displayed in the bottom text box within the routing tool window. If no route is found, the distance is set to zero and the user is notified that a feasible path does not exist between the two points they defined.

4. Implementation and Expert Evaluation

In order to assess the accuracy and usability of U-Access, the application was provided to the Center for Disability Services (CDS). The CDS agreed to use the application for three weeks with their staff, as well as provided it to those students who use the computer lab within the CDS office. Each person that used the application was asked by the CDS to fill out an evaluation form. Users were asked to evaluate U-Access on four criteria web page organization/layout, web graphics, technically, and utility. The Appendix provides the evaluation instrument. We received a total of eight completed expert evaluations.

Experts were asked to evaluate the overall organization and layout of the web page. This included questions on first reactions to the initial page design and if the design had an affect on their ability to navigate the website. In general, experts felt the page needed more instructions on how to use the tools. Although U-Access utilizes common Web tool for zoom and identify, the tools for selecting an origin and a destination did not appear to be easily understood. Several experts felt that an initial dialog box with instructions would have been useful.

With respect to the graphics, experts felt the map was “readable” in that sidewalks, buildings, and vegetation were clearly delineated. They also felt that the optimal route was clearly shown. There were two common critiques in this regard. The first was that the buildings needed to be labeled. Currently, users are forced to hover their mouse over the building as opposed to being able to simple glance at a building and identify it. The second critique was that while the optimal path was clearly presented, they felt that it is still difficult to navigate the path even with the map. We believe this is a function of nature of unnamed paths. While streets have names to assist in giving directions, the sidewalks do not have such amenities.

Experts were asked to evaluate the technical performance of U-Access. This included questions on page load time, path correctness and interactions with the website. Experts felt the web page loaded quickly except if the computer had pop-up blocker software enabled. Further investigation determined that pop-up blockers prohibit the display of embed objects. Thus, users are forced to turn off their pop-up blocker software before they were able to use the web page. On the flip side, the experts were impressed with the route accuracies. One expert decided to test the accuracy and traveled the provided route and verified its correctness.

The last component of the evaluation was utility. Experts had the chance to express whether they would use the application on a regular basis, and whether they felt an application such as this one would be helpful for the general public. There was an overwhelming response that they would have loved such an application when they first came to campus. This would make sense that after a person has been on campus long enough they know which routes are feasible, and which routes are not. One particular student felt that U-Access would be helpful at the beginning of every semester.

The expert evaluations were a necessary part of this research in order to test the utility of U-Access. The evaluations reinforced the need and demand for such an application. The comments received in the evaluation forms will be considered before the application is served to the general public. As the evaluation of U-Access helped one disabled student find a shorter route from the bus stop to a frequently traveled destination, it reveals the potential that U-Access could have in better assisting all students.

5. Conclusion

The U-Access application is a tested and applied Web-based navigational system for routing people with varying ability levels in an urban environment, specifically within the University of Utah campus. U-Access is designed for both a planning purposes and to assist pedestrians of multiple mobility levels in identifying the shortest feasible routes on campus.

U-Access may not be available to all people. For example, those who lack knowledge of personal computers or whose physical or visual impairments are too severe may not benefit from it. Typically, the group of people that are most excluded from the benefit of adaptive technologies are the elderly (Matthews *et al.* 2003). However, as Internet access begins to appear in classrooms and public libraries, this barrier is being broken down. U-Access also does not claim to cure the social exclusion that many disabled people face (Gleeson 1999). However, it is hoped that this application heightens awareness for the provision of accessible tools to people with disabilities in order to assist them in gaining independence within urban environments.

As U-Access attempts to minimize the population marginalized by urban structures, issues of the visually impaired have not yet been addressed. There has been some research conducted on the matter by the W3C (W3C, 2000b) with their Web Accessibility Initiative. This initiative provides an XML based SVG-to-text converter for visually impaired Web users. Currently, when visually impaired people want to read an Internet map, they have a special printer that prints the map with raised lines for each feature on the map. The map is then placed on a tablet so that the user can interact with the SVG map by means of sound and touch. This allows them to query items on the map just like users who interact with the monitor (Campin *et al.* 2003). However, unlike automobile routing where streets are named and intersections are clearly marked in the real world, pedestrian networks seldom distinguish among network intersections. While the visually impaired can make map inquiries on the Internet, there are still issues to be resolved regarding translating the optimal route from an Internet map to a useful tool in the real world. Thus, there is still much work to do in order to better route disabled people as a whole through urban environments.

6. References Cited

Campin, B., W. McCurdy, L. Brunet, E. Siekierska. 2003. SVG Maps for People with Visual Impairment. *Proceedings from SVG Open Conference* July 2003.

Cherkassky, B.V., A.V. Goldberg, T. Radzik. 1993. Shortest Path Algorithms: Theory and Experimental Evaluation. Discussion paper, Department of Computer Science, Stanford University, Stanford, CA.

Church, R.L., J.R. Marston. 2003. Measuring Accessibility for People with a Disability. *Geographical Analysis*. 35(1): 81-96.

Dewey, C. (2001) Navigation Services for Wheelchair Users Consideration of Special Physical Demands in Computer-based Wayfinding [Unpublished Master's thesis]: Institute of Geoinformatics, University of Muenster.

Foster, L. 2004. Progressive Access: Unique Solutions for Historic Buildings. *Journal of Architectural Conservation*. 10(3):73-86

Franklin, K. 2005. A three-stage test provides some answers in disabled access cases. *Architects' Journal*. 221(5), p46.

Gallo, G., S. Pallottino. 1998. Shortest Paths Algorithms. *Annals of Operations Research*. 13:3-79.

Gleeson, B. 1999. Can Technology Overcome the Disabling City? In *Mind and Body Spaces*, by Butler, R. and Parr, H. Routledge, London, 98-118.

Goodrich, M.T., R. Tamassia. 1998. *Data Structures and Algorithms in Java*. New York, NY: Wiley & Sons, Inc.

Golledge R., J.M. Loomis, R.L. Klatzky, A. Flury, X.O. Yang. 1991. Designing a personal guidance system to aid navigation without sight: progress on the GIS component. *International Journal of Geographical Information Systems*. 5:373-95.

Golledge, R., R.L. Klatzky, J.M. Loomis, J. Speigle, J. Tietz. 1998. A geographical information system for a GPS based personal guidance system. *International Journal of Geographical Information Systems*. 12:727-49.

Little, J. 1995. What has ADA Really Done? *Accent on Living*. 39(4):28-30.

Matthews, H., L. Beale, P. Picton, D. Briggs. 2003. Modeling Access with GIS in Urban Systems (MAGUS): capturing the experiences of wheelchair users. *Royal Geographical Society*. 35(1):34-45.

Matthews, M.H., P. Vujakovic. 1995. Private Worlds and Public Places: Mapping the Environmental Values of Wheelchair Users. *Environment and Planning A*. 27:1069-1083.

Miller, H.J., S. Shaw. 2001. Geographic Information Systems for Transportation: Principles and Applications. New York, NY: Oxford University Press Inc.

OGC. 1999. The Open GIS Guide (Third Edition) Introduction to Interoperable Geoprocessing and the Open GIS Specification, available at http://www.opengis.org/techno/guide/guide980615/Guide980615.rtf.

Plewe, B. 1997. GIS Online: Information, retrieval, mapping, and the Internet. Sante Fe, NM: Onword Press.

Satalich, G. A. 1995. Navigation and Wayfinding in Virtual Reality: Finding Proper Tools and Cues to Enhance Navigation Awareness [Unpublished Master's thesis]: University of Washington, Seattle, WA.

Sheffi, Y. 1985. Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods. Englewood Cliffs, NJ: Prentice Hall.

Shekhar, S., S. Chawla. 2003. *Spatial Databases: A Tour*. Upper Saddle River, NJ: Prentice Hall.

W3C. 2000a. Accessibility Feature of SVG. http://www.w3.org/TR/SVG-access.

W3C. 2000b. "SVG Linearizer Tools." http://www.w3.org/WAI/ER/ASVG.

Waldura, R. 2003. Dijkstra's Shortest Path Algorithm in Java. Available at *http://renaud.waldura.com*.

Appendix: U-Access Evaluation Dimensions

1. ORGANIZATION/LAYOUT:
    a. Was the design of the web page helpful in understanding the intention of the software?
    b. Does the layout of the page affect your ability to navigate? Are the tools clearly labeled?
    c. Comments/Suggestions:

2. GRAPHICS:
    a. Was all the information clearly presented?
    b. Were you able to clearly delineate the route provided to you?
    c. Comments/Suggestions:

3. TECHNICAL:
    a. Does the page load reasonable quickly?
    b. Does the page respond well to your interactions (mouse clicks)?
    c. Did many events occur that were not expected?
    d. Please comment on the "correctness" of the routes?
    e. Comments/Suggestions:

4. USABILITY:
    a. If you had this application, would you use it on a regular basis (daily, weekly)?
    b. Do you feel that an application of this type would be helpful for the general public?
    c. Comments/Suggestions: