

# CSC450 Analysis 1-3

G.O.D. (aka H.I.M.)

Jody Huie, Steve Palmisano,  
Matthew Singletary, Christopher A. Watford

September 7, 2006

## 0.1 Requirement Specification

### 0.1.1 Supported Activity List

- Storing a road network in a manner consistent with easy retrieval
- Navigation of the road network
- Intelligent route planning along the road network
- Rendering the network of roads
- Traffic simulation
- Points of Interest (routing to and from)

### 0.1.2 Human Computer Interface

- **Network query tool:** What is the name of this road? What is the address here?
- **Point and Click Route Planning:** Plan a route with a few clicks of the mouse.
- **Timeframe Navigation:** Display routes over a given timeframe.
- **Road Network Renderer:** Google Maps style (or even Google Maps!)

### 0.1.3 Solved Problems List

- How do I get from A to B?
- What is the fastest route from A to B at time T?
- How do I get to A, B, and C?
- What is the fastest route from to A, B, and C at time T?

### 0.1.4 Information Source List

- A **GPS Unit** provides real time positioning information
- A **GiST** database provides information about the road networks
- A **Network Graph** database provides information for routing
- A **Traffic** database provides simulation data for routing

### **0.1.5 Information Requesting Organizations List**

- End users who are looking to plan a trip, locate an address, or who are curious about fastest routes
- Traffic planners who are looking for a tool to mark possible future roadway improvement locations
- Cab drivers who need to find the most efficient way to pick up fares
- GPS enthusiasts who just have to have the coolest in auto-navigation

### **0.1.6 Checks and Balances**

- Ensure navigation system doesn't attempt to do dangerous things (e.g. go the wrong way on a one-way street)
- Ensure the user cannot request impossible routing scenarios
- Restrict the number of waypoints to curb computation time

### **0.1.7 Security and Fault Tolerance Requirements**

- The Department of Homeland Security will want to know why we've used city data
- On the fly handling of bad or old data via user updates (avoidances)

### **0.1.8 Interoperating Systems List**

- Databases of different GiST/routing/POI/weighting information
- GUI where the user makes choices (web based? standalone?)
- Backend which does the physical routing and navigation

### **0.1.9 Estimates of Present Information Capacity and Projected Growth**

- The road network database for North America is 1.93 **GiB**. PostgreSQL can handle databases in the terabyte range.
- The point-of-interest database could possibly be unfathomably large. No current plans to support POI in full at this time.

### 0.1.10 Prioritization of Requirements

0. Storing a road network in a manner consistent with easy retrieval
0. Navigation of the road network
  1. Intelligent route planning along the road network
  1. Rendering the network of roads
  2. Traffic simulation
  9. Points of Interest

### 0.1.11 Ethical Concerns

- Software with this much infrastructure information could always be used for 'Eevil'. Restricting usage to citizens would probably be something the government would mandate, but we do not feel there are any real ethical concerns along this line.
- If this were the first or second centuries CE, the name of our software would get us smote.

## 0.2 Scenarios

### John: New to a City

#### Precondition

**John** has just moved to a new city and needs to locate certain places along with the quickest routes to them. He could learn by trial and error as to the best routes, however, this could be timeconsuming.

#### Postcondition

**John** uses **G.O.D.** to plan his different trips based on the time of day he would be making them. The data he is provided allows him to better learn the efficient corridors through the city. The end result is John is able to make more informed decisions about navigating in his new city.

### Jack: Unforeseen Traffic Impediment

#### Precondition

**Jack** has been living in the city for quite sometime. He has learned the ins and outs of traffic navigation through the crowded and often busy streets that he travels every day to work. However, when there is a wreck or construction, he is unable to adapt quickly enough while in-route.

### **Postcondition**

**Jack** is able to use the on-the-fly avoidance features of **G.O.D.** to quickly route around the traffic impediment. Jack is able to save time by not waiting in traffic.

### **Jill: Company Outing Planning**

#### **Precondition**

**Jill** is in charge in setting up a Friday Night Mixer for her company and needs to choose a venue which is conveniently accessible for her boss.

#### **Postcondition**

**Jill** uses the timeframe routing features to examine the route times from the boss's house to the different venues. Jill is able to adjust the time of the event and dynamically see the effect on the routes she has previous requested. Jill can now pick a venue and time which will not get her fired.

## **0.3 Primary Class List**

### **0.3.1 Classes**

- Database
- Interchange
- GPSSource
- Route
- Waypoint
- World
- WorldRenderer

### **0.3.2 Interfaces**

- INavigator
- INavigatable
- IPointOfInterest

### **0.3.3 Structures**

- Point
- Polygon
- Polyline

### **0.3.4 Enumerations**

- NavigatableNone
- NavigatableBoth
- NavigatableForewards
- NavigatableBackwards