



# Emergency Response Applications

## *Dynamic Plume Modeling and Real-Time Routing*

In a crisis situation, toxic gas can be released into the air, blocking routes for emergency responders. Rescue workers must be able to compute the shortest and safest paths in the presence of toxic gas dispersions that move dynamically with changing wind speed and direction. To model this, the authors developed weather retriever software, which fetches weather details about a particular location from the Internet and acts as the station for atmospheric measurements for the Aloha plume-modeling software. The authors also explored options for displaying this dynamic plume on a geographic map.

**Pavan Kumar Chitumalla,  
Douglas Harris,  
Bhavani Thuraisingham,  
and Latifur Khan**

*University of Texas at Dallas*

**E**mergency response deals with the reaction to a natural or man-made disaster and involves containing the disaster situation. The process is time-critical because many lives depend on how quickly and accurately emergency responders make decisions. During an emergency response, *emergency operations centers* (EOCs) face different problems depending on the category the emergency falls under (fire, earthquake, biochemical attack, hazardous chemical release, or hurricane, for example<sup>1</sup>).

Our project at the University of Texas at Dallas targeted one class of problems – those that arise when a hazardous, airborne chemical is released into an inhabited area. Current software appli-

cations can model this gas release on a map with a *plume* – that is, a plot between the chemical concentration in the air and the distance from an accident site, which is essentially a set of polygons – but these plumes are static and don't consider the wind and its dynamic nature. With a change in wind direction and speed, a toxic plume changes shape and extent.

To address this problem, we utilized the Aerial Locations of Hazardous Atmospheres (Aloha) software to model the wind's dynamic aspects and produce a real-time plume that's dynamic in nature. Our model also shows the plume's trail as it moves with the changing wind speed and direction. By incorporating the plume into a geo-

graphic map, we can see roads inside the plume that could be blocked by gas and dangerous for traffic. To help emergency responders avoid these hazardous regions, our research leverages geospatial operations, extending a geographic information systems (GIS) application to determine the shortest and safest path between any two locations on the map.

### Aloha and Modeling Gas Dispersions

*Dispersion* is a term modelers use that includes advection (moving) and diffusion (spreading). A dispersing vapor cloud will generally move downwind and diffuse in a crosswind and vertical direction (see [www.epa.gov/OEM/cameo/pubs/ALOHAManual.pdf](http://www.epa.gov/OEM/cameo/pubs/ALOHAManual.pdf)). Dispersion occurs during various types of hazards, including overpressures (blast-force explosions), fires, and toxic gas clouds released into the atmosphere.

In modeling a gas leak, consider the following scenario. A train derailed in Allen, Texas, causing a tank car that contains 33,800 gallons of chlorine to leak from a crack roughly 20 inches long and 0.1 inches wide, which formed due to high stress and the tank car's impact with the ground. Assume the tank is 70 feet long and that the chlorine is stored in its gaseous state with a pressure of two atmospheres. When the incident occurs, the wind is blowing from the northeast at 15 miles per hour, the air temperature is 67°F, and the relative humidity is 65 percent. We must immediately assess the toxic gas dispersion.

To model this scenario, we used icons and tools from the Aloha software application, which the US Environmental Protection Agency (EPA) and the National Oceanic and Atmospheric Administration (NOAA) developed specifically for emergency responders. (You can download a free copy at [www.epa.gov/ceppo/cameo/aloha.htm](http://www.epa.gov/ceppo/cameo/aloha.htm).) Once we configured the input parameters, Aloha was ready to create the plume model, or *threat plot* – a plot between the concentration level of a toxic chemical in the air and the distance from the source. We can view the plot by selecting Threat Zone in the Display menu.

Figure 1 shows the threat plot. By default, Aloha configures three levels of concern for this case: the yellow region shows a 0.5 parts per million (ppm) or greater concentration of chlorine in the atmosphere, the orange zone 2 ppm or greater, and the red the most hazardous with a concentration of 20 ppm or greater.

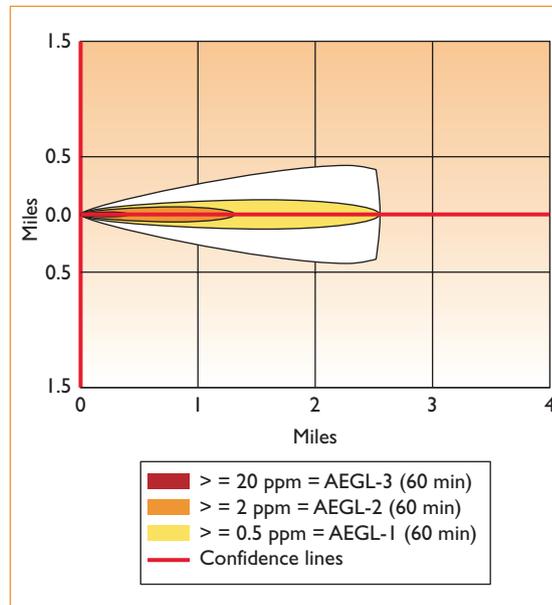


Figure 1. Threat plot of a chlorine gas leak from a derailed train car. The yellow region shows a 0.5 parts per million (ppm) or greater concentration of chlorine in the atmosphere, the orange zone 2 ppm or greater, and the red 20 ppm or greater.

### Mapping Applications and Geoprocessing

Although Aloha can generate the plume model (threat plot), to convey the exact picture of the toxic plume, we must still superimpose it onto a geographic map. Drawing plumes on a map helps us precisely gauge the toxic plume's spread in an inhabited area, so that emergency responders can determine which areas must be evacuated or which buildings will be impacted. Thus, we must use a mapping application to draw the extent of these Aloha threat plots.<sup>2</sup>

Several mapping applications are available, from Web-based ones such as Google or Yahoo maps to stand-alone applications such as the EPA's Mapping Application for Response, Planning, and Local Operational Tasks (Marplot) and the Environmental Sciences Research Institute's (ESRI's) ArcGIS application suite. The application we need must support geoprocessing, or "the process of applying geographic analysis and modeling to data to produce new information" ([www.esri.com/software/arcgis/concepts/geoprocessing.html](http://www.esri.com/software/arcgis/concepts/geoprocessing.html)).

Our goal was to create a map that "blocks" roads that are unsafe for navigation and provides the shortest alternate routes with driving directions between any two locations. We examined various map applications for use with

the Aloha software, including Marplot, Google Maps, and ArcGIS.

Marplot is a general-purpose mapping application compatible for use with Aloha. These two applications integrate seamlessly and provide integrated mapping and plume-modeling capabilities. However, Marplot lacks routing functionality, geoprocessing capability, and dynamic plume modeling (<http://archive.orr.noaa.gov/comeo/marplot.html>).

Google Maps is a Web-based, fast, easy-to-use, modern application that provides routing functionality and a JavaScript-based API. Although using Google Maps is considerably better than using Marplot when it comes to routing functionality, it lacks a geoprocessing ability. After superimposing the plume on the map, there's no means for extracting the roads that fall inside this plume and then blocking them. Thus, in our scenario, if we try to find the shortest alternate path between any two locations whose most direct path crosses the red region, the application fails and returns the same routing directions even in the presence of the plume.

Finally, ESRI's ArcGIS is a suite of software applications for building a complete GIS, which is a "system for capturing, storing, analyzing, and managing data and associated attributes which are spatially referenced to the earth" (<http://en.wikipedia.org/wiki/GIS>). The ArcGIS framework is based on Microsoft's Component Object Model (COM) and is flexible. We can develop custom components in any COM-compliant language (C#, C++, and Visual Basic, for example) and utilize them in ArcGIS. ArcMap is a comprehensive map-authoring application for desktop GIS that comes with the ArcGIS suite and performs map-based tasks such as cartography, map analysis, and editing ([www.esri.com/software/arcgis/about/arcmap.html](http://www.esri.com/software/arcgis/about/arcmap.html)). We decided to use ArcGIS – specifically, the ArcMap application – to create our map.

### System Architecture and Implementation

Before developing our own solution to the toxic gas problem, we examined various others.

The US Defense Threat Reduction Agency's (DTRA's) Hazardous Prediction and Assessment Capability (HPAC) can accurately predict the effects of hazardous chemicals in the atmosphere. Additionally, it can model attacks from nuclear, biological, and chemical weapons. The Conse-

quence Assessment Tool Set (CATS) imports HPAC's plume model into ArcGIS for further analysis. Together, CATS and HPAC constitute a comprehensive solution from the DTRA, but one that lacks dynamic plume behavior and routing functionality.

The Automated Decision Aid System for Hazardous Incidents (ADASHI) is commercial software from OptiMetrics (see [www.safetyact.gov/DHS/SActHome.nsf/Awards?ReadForm](http://www.safetyact.gov/DHS/SActHome.nsf/Awards?ReadForm)). Its plume-modeling functionality is based on Aloha, and it has its own mapping application. ADASHI is easy to use and provides a dynamic plume model, but it doesn't have routing capability.

Due to these deficiencies in existing solutions, we developed our own system. We wanted to test this system in the Collin County region of Texas, so we first downloaded the necessary files to start our work from the North Central Texas Council of Governments (NCTCOG) GIS data clearinghouse, which contains high-quality digital geographic data for the North Central Texas region. (It's available for free from [www.dfwmaps.com/clearinghouse/](http://www.dfwmaps.com/clearinghouse/)). We carried out the entire project using the North American Datum 1983 (NAD83) coordinate system, which lets us identify a point on the Earth on a planimetric map ([www.towermaps.com/nad.htm](http://www.towermaps.com/nad.htm)). We downloaded the shapefiles that contain Collin County's highways, public roads, and county boundaries. These shape files would only enable ArcMap to display a map of Collin County just like any other mapping application, so to enable the routing functionality, we created a network data set from the files.

To set the stage for dynamic plume models and emergency routing, we developed a three-step process:

- First, semiautomate Aloha so that it can read live weather data from an incident site and generate new threat plots rather than requiring users to enter atmospheric details manually. This is crucial because in high-pressure situations, emergency responders won't have enough time to repeatedly enter wind speed, direction, and air temperature.
- Next, extend ArcGIS to read Aloha's footprint file, which contains the threat plots, and display those plots on the map.
- Finally, perform a spatial query that extracts all the roads that were inside the plume,

blocks these roads to traffic, and leaves a plume trail when the system draws an updated one.

Figure 2 shows the system architecture.

### The Live Weather Retriever

To complete the first step, we developed a *live weather retriever*, a software tool that can obtain a list of all surrounding live weather stations given a zip code. Here, we address how we developed this tool.

**Requirements.** The weather retriever's first requirement is to fetch the real-time weather data given the incident site's zip code. The second requirement is that weather details must be formatted as Station for Atmospheric Measurements (SAM) messages before the retriever sends them to Aloha. This partially automates one of Aloha's required inputs. Finally, the weather retriever should measure conditions every two seconds, compute the average of these measurements, and send it to Aloha approximately every 30 seconds.

**Design.** We chose the WeatherBug weather service as our weather data provider because it manages and operates the largest network of weather stations throughout the world (roughly 8,000 stations; [www.weatherbug.com/aws/aboutus.asp](http://www.weatherbug.com/aws/aboutus.asp)). These weather stations generate live weather reports about their surroundings every second, unlike the NOAA weather service and other Web services that update every hour. Because Aloha imposes the restriction that the time interval between any two SAM messages it receives should not exceed 30 seconds, we configured the weather retriever to send weather data messages every 25 seconds to accommodate Internet delays and other computational latencies that might occur. The weather retriever is also very flexible and can be configured to work with any Web-based weather service.

**Implementation.** We developed the weather retriever in C#; it consists of two components, one based on the WeatherBug API, which retrieves weather details from the Web, and the other based on the EPA's dynamic-link library, NOAA\_32.dll, which communicates the weather data to the Aloha software. The WeatherBug API lets users build custom as well as customize weather

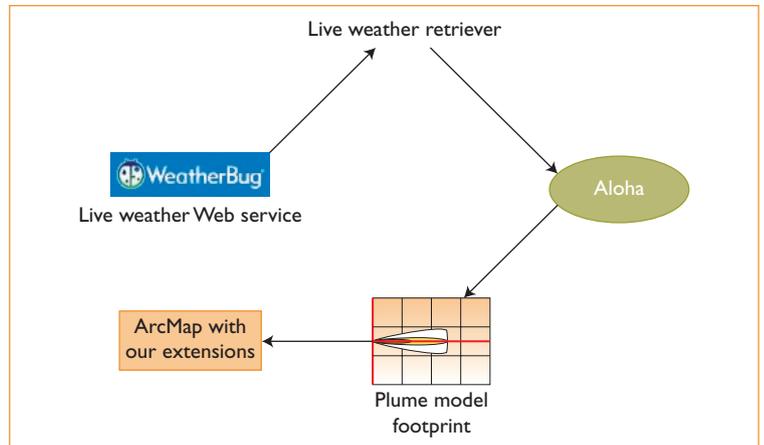


Figure 2. System architecture. The weather retriever retrieves weather data from the weather service. Aloha then processes this data to create the plume model, which we input into Arcmap together with our system. We display the routes as output.

applications and provides various programming functions that are required to establish communication with Aloha. We installed NOAA\_32.dll together with the Aloha installation.

To retrieve the list of weather stations in the vicinity of a given zip code, the weather retriever invokes the WeatherBug API call; for an application to communicate the weather data with Aloha, it must identify itself with a unique four-character signature and perform a series of steps, including registering the application with NOAA\_32, registering with Aloha by sending a register application message, receiving messages from Aloha, sending a "bye" message once communication with Aloha is complete, and, finally, unregistering the application from NOAA\_32.<sup>3</sup>

In our system, the weather retriever first registers with Aloha by sending a message informing Aloha that an external application is ready to communicate atmospheric data. Aloha then sends a message with START as the data parameter to indicate that it's ready to accept the weather data. Then, the weather retriever starts sending messages with weather data about the incident site. Once Aloha is done modeling or about to exit, it again sends a message, this time with STOP as the data parameter. When the weather retriever receives this message, it stops sending weather data.

### Extending ArcGIS to Process Plume Models

ArcGIS is built from a library of scalable, modular, and cross-platform software components called ArcObjects.<sup>4</sup> The ArcObjects framework

is based on Microsoft's COM technology,<sup>5</sup> so we can use ArcObjects with any COM-compliant language. To extend ArcMap's functionality, we used its embedded Visual Basic for Applications (VBA) editor and developed a toolbar in ArcMap that users can click to process Aloha's plume model every 25 seconds, draw it on the map, extract the roads that are inside the plume, and place barriers on those roads. We can then use the ArcMap Network Analyst extension, which analyzes transportation layers in a map, to route a path that excludes roads with these barriers. (The Network Analyst is also based on the ArcObjects framework, so extending it to our needs is easy.) Hence, given any two locations on the map, Network Analyst always computes a route that avoids the threat zone.

**Understanding Aloha's threat zone files.** Aloha assumes that the incident site is the origin of the threat plot; the gas dispersion polygon is represented by a set of points in the Aloha footprint file called ALO\_FTP.PAS (a text file). This file stores each point representing the polygon by its distance in meters east and north relative to the origin. A point is represented as a triple: `character number1 number2`. If the character is `t` or `T`, then it's a comment; if it's an `M` or `L`, then it denotes a point located at `number1` meters north and `number2` meters east of the origin, where origin is the source of the leak. For example, assume `p1`, `p2` are one meter east and 0.5 meters north and south of the origin, respectively. In the footprint file, `p1` would be stored as `L 1 0.5`, and `p2` as `L 1 -0.5`.

**Drawing the plume on ArcMap.** When the user clicks on the ArcMap map with the mouse, the coordinates at that location are returned as a point *P*. ArcMap transforms these coordinates at *P* into map units using ArcObject's transformation function. Once the application obtains the incident site's location in map units, it can use Thaddeus Vincenty's<sup>2</sup> inverse equations to convert all the points into map units and subsequently draw the polyline representing the plume on the map.

**Geoprocessing the plume to extract the roads inside it.** The biggest advantage to using ArcMap is that it supports geoprocessing – with other mapping software, the map is just an im-

age file (a JPEG or GIF), so after superimposing a plume, extracting the area features (a feature is any geographic entity, such as a school or road) inside this polygon isn't possible. However, given a map and a polygon from software that supports geoprocessing, we can extract features that lie within or surround this polygon. Given a point on a map, for example, we can use geoprocessing to extract all the schools that lie within a five-mile radius of this point. To extract roads that are inside the toxic plume, we use a spatial query that returns a cursor (a pointer to a collection of records) to the set of roads that are necessary for responding to the emergency. Note that in addition to ArcGIS, products such as Oracle Spatial can also perform spatial queries. We chose ArcGIS for the reasons mentioned in the "Mapping Applications and Geoprocessing" section.

**Adding barriers to roads inside the plume.**

After identifying all the roads inside the plume model, the application adds them to the Barriers feature layer; the Network Analyst extension then uses this layer to compute the route between any two locations. Network Analyst computations are all based on an underlying feature layer called a network data set (we explained how to create this data set at the beginning of the "Architecture and Implementation" section).

Given any two locations on the map, the Network Analyst can find the route between them along with driving directions. We can specify a set of prohibited roads using the Barriers feature class; the Network Analyst avoids those roads while computing the route, letting responders find the safest and shortest paths in the presence of a plume. Note that these paths are truly alternative paths and, furthermore, are safer for the given 25-second time period.

### Simulation and Results

To test our system, we simulated an incident in which a chlorine-carrying truck has an accident at the Plano Road and Coit intersection in Collin County, Texas. The gas tank is cracked due to the accident's impact, and toxic gas is escaping into the air. We perform the following steps to set up modeling our scenario:

- Start the weather retriever and enter the incident location's zip code.

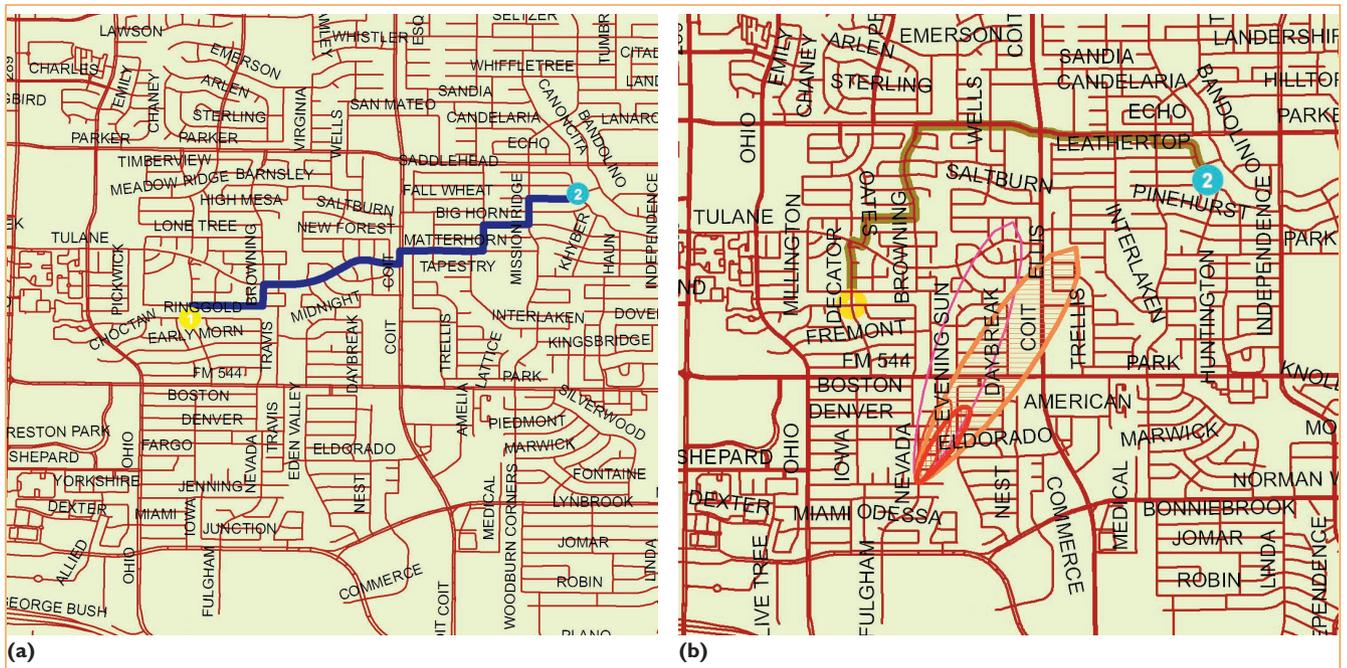


Figure 3. The Plano Road and Coit intersection in Collin County, Texas. We can see (a) the shortest path between points 1 and 2 under normal conditions. After an accident in which a chlorine-carrying truck crashes at the intersection, toxic gas begins escaping into the air. In modeling this scenario, we can see (b) the plume model originating from the accident. The polygons in bold represent the toxic plume’s current position, whereas the thinner polygons represent its earlier positions. The system (or application) automatically computes the alternative shortest and safest path between points 1 and 2.

- Start ArcMap, enabled with the Network Analyst extension, load the network data set created, and then pan and zoom to the incident location.
- Start Aloha and input relevant parameters (as explained in a previous section).

Choosing any two locations and solving the routing problem using the Network Analyst gives us the shortest and safest path, avoiding those roads that go through the toxic gas. Thus, we achieve a dynamic plume model as well as emergency routing functionality. Figure 3a shows the shortest path between points 1 and 2 under normal conditions. Figure 3b shows the plume model originating from the accident at the intersection. We run the system for two minutes and can see the effects of the weather on the plume. The polygons in bold represent the toxic plume’s current position, whereas the thinner polygons represent its earlier positions. The application automatically computes the alternative shortest and safest path between points 1 and 2, which also appears on the map in Figure 3b.

A discussion of the distributed tools we used

in our modeling as well as details from our experiment are available elsewhere.<sup>1</sup>

### System Limitations

We’ve integrated several modules to develop an innovative system for emergency response and are currently integrating our system with operational systems; however, our system has several limitations that we will address in the future:

- Given that some roads are closed off in an emergency situation, traffic congestion will occur far above normal levels. We must model this congestion from regional origin-destination models if we want to use a shortest-path algorithm.
- We must thoroughly analyze various routing algorithms and select one for the problem we outlined in the previous section.
- As depicted, our plume model doesn’t consider terrain. When tall buildings are involved, especially, terrain can drastically affect a plume’s dispersion and direction. This is particularly important if the results obtained for alternate paths are incorporated

- into a GIS and used for decision making.
- We must study the information flow between the system's various components and model it so that we can improve system performance.
- We must also incorporate a buffer around the plume to account for uncertainty in the model. This is critical for a practical system.
- The plume model generates regular elliptical plumes – we must further investigate why this phenomenon occurs.
- We must conduct usability studies of the algorithms and the use of ArcMap.

We are making system enhancements based on these limitations, while also integrating the system into operational environments.

**F**or this project, we integrated four different technologies – plume modeling, geoprocessing, routing, and mapping – to achieve a novel product that can draw dynamic plume models and find the shortest and safest emergency routes under time-constrained situations. Many components we built for this project are flexible and customizable to accommodate emergency responders' demands. For example, if the WeatherBug weather Web service isn't sufficient, then users can easily change the source for the weather retriever. Because we based our work on ArcGIS, which is flexible, users can further enhance maps by adding more layers – adding a hospital layer, for instance, will let emergency responders determine which hospitals are in close proximity to the incident site and evacuate the victims. Users can also add real-time traffic data, if available, to make the model compute emergency routes with less traffic.

In addition to addressing the limitations we just mentioned, we're currently integrating our prototype with a fusion center being built by the Collin County Homeland Security Office. We're also developing an E-Plan system for the EPA that we expect to transition into the emergency preparedness office at the US Department of Homeland Security, and we plan to integrate our prototype into that system. E-Plan is a Web-based system that provides first responders and other emergency preparedness personnel with on-site hazardous chemical information for facili-

ties around the US (<http://eplannews.utdallas.edu/FAQs.htm>). □

### Acknowledgments

We thank the anonymous reviewers for many insightful comments. Some of our current enhancements are based on their suggestions.

### References

1. P.K. Chitumalla, *Dynamic Plume Modeling and Real-Time Routing for Emergency Response Applications*, master's thesis, Dept. of Computer Science, Univ. of Texas at Dallas, May 2007.
2. T. Vincenty, "Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations," *Survey Review XXII*, vol. 176, 1975, pp. 88–93.
3. J. Muhasky, "Communicating with Aloha," Nat'l Oceanic & Atmospheric Administration white paper; 2007; [http://response.restoration.noaa.gov/book\\_shelf/1417\\_Communicating%20With%20ALOHA.pdf](http://response.restoration.noaa.gov/book_shelf/1417_Communicating%20With%20ALOHA.pdf).
4. *Exploring ArcObjects*, vol. 1, ESRI, 2001.
5. "Designing a Portable Weather Station for Use with Aloha," Nat'l Oceanic & Atmospheric Administration document, 2007; [http://response.restoration.noaa.gov/book\\_shelf/982\\_Metdesig.pdf](http://response.restoration.noaa.gov/book_shelf/982_Metdesig.pdf).

---

**Pavan Kumar Chitumalla** works at Microsoft, where his research interests are in data security and geospatial data management. Chitumalla has an MS in computer science from the University of Texas at Dallas. Contact him at [pavanc@winse.microsoft.com](mailto:pavanc@winse.microsoft.com).

---

**Douglas Harris** is a research professor and executive director of the Cyber Security and Emergency Preparedness Institute at the University of Texas at Dallas. He has a PhD in engineering management from the Southern Methodist University. Contact him at [edh@utdallas.edu](mailto:edh@utdallas.edu).

---

**Bhavani Thuraisingham** is a professor of computer science and the director of the Cyber Security Research Center at the University of Texas at Dallas. She has an MS in mathematics from the University of Bristol and a PhD in computer science from the University of Wales. Thuraisingham is an IEEE fellow and received the 1997 IEEE Technical Achievement Award for her research in data security. Contact her at [bhavani.thuraisingham@utdallas.edu](mailto:bhavani.thuraisingham@utdallas.edu).

---

**Latifur Khan** is an associated professor of computer science and the director of the Data Mining Laboratory at the University of Texas at Dallas. He has a PhD from the University of Southern California. Contact him at [lkhan@utdallas.edu](mailto:lkhan@utdallas.edu).