A data model for route planning in case of forest fires

Zhiyong Wang[a,*], Sisi Zlatanova[a], Aitor Moreno[b], Peter van Oosterom[a], Carlos Toro[b]

[a] *Delft University of Technology, Jaffalaan 9, 2628BX Delft, The Netherlands*
[b] *Vicomtech, Mikeletegi Pasealekua 57, 20009 San Sebastian, Spain*

Abstract

The ability to guide relief vehicles to safety and quickly pass through environments affected by fires is critical in fighting forest fires. In this paper, we focus on route determination in the case of forest fires, and propose a data model that supports finding paths among moving obstacles. This data model captures both static information, such as the type of the response team, the topology of the road network, and dynamic information, such as sensor information, changing availabilities of roads during disasters, and the position of the vehicle. We used a fire simulation model to calculate the fire evolution. The spread of the fire is represented as movements of obstacles that block the responders' path in the road network. To calculate safe and optimal routes avoiding obstacles, the A* algorithm is extended to consider the predicted availabilities of roads. We prove the optimality of the path calculated by our algorithm and then evaluate it in simulated scenarios. The results show that our model and algorithm are effective in planning routes that avoid one or more fire-affected areas and that the outlook for further investigation is promising.

Keywords: Emergency navigation, Fire simulation, Data model, Algorithm

---

1. Introduction

Natural fires have caused enormous socioeconomic losses and created many victims in the past few years. Recently, there has been growing interest in understanding and mitigating the effects of these disastrous events. In fighting forest fires, a wide range of response activities and emergency operations are involved, such as transporting injured persons, distributing supplies, and evacuating citizens, all of which require navigation aids. Because the radiant heat released during burning can be considered obstacles that might make some roads unsafe and temporarily inaccessible (Taylor and Freeman, 2010), emergency managers need a path planner that is capable of finding a safe and optimal route that avoids fire-affected areas.

Navigation has been thoroughly studied from varied theoretical perspectives and across multiple disciplines, such as robotics, geomatics and applied mathematics (Chabini and Lan, 2002; Ge and Cui, 2002; Huang et al., 2007; Delling et al., 2009). Nevertheless, very few research efforts have been devoted specifically to emergency navigation problems in the context of moving obstacles that dynamically affect the road network (Wang and Zlatanova, 2013b). Although some studies have some relevance for route planning in case of disaster events (Mioc et al., 2008; Liu et al., 2006), the issues that arise in the path planning during disasters have not yet been fully addressed. On one hand, the existing emergency support systems (Parker et al., 2008; Johnson, 2008) are capable of finding the shortest route to a certain location, taking the damages to the infrastructure into account, but do not consider the dynamics of disasters, particularly the predicted information on their developments, which limits their practical applications in disaster response. Some studies of emergency navigation used crowdsourced data regarding the state of the road to calculate the shortest path (Nedkov and Zlatanova, 2011; Neis et al., 2010). However, they can only cope with static obstacles, and do not offer the routing functionality required to avoid moving obstacles. On the other hand, most research on dynamic obstacles has been centered on robotics (Li

---

[*] Corresponding author. Tel.: +31 (0)152787934; fax: +31 (0)152784422. E-mail addresses: Z.Wang-1@tudelft.nl

et al., 2009; Masehian and Katebi, 2007; Gonzalez et al., 2012). The results from these studies could benefit the navigation of first responders in certain aspects. Nevertheless, the focus of their research is mainly on planning obstacle-avoiding paths in a given free space, without the constraints of a transportation network.

One of the most critical aspects in emergency navigation is information, most of which falls into two categories, static and dynamic. Static information is relevant to topographic and territorial data (e.g., land use, road network, buildings, and locations of fire hydrants). Most of the static data can be obtained through municipality offices and the emergency reponse (ER) sectors, as well as public resources, such as the location of fire hydrants on www.openfiremap.org and general maps from OpenStreetMap (www.openstreetmap.org). Dynamic information is more related to the incident description and its impacts, damages, and sensor measurements, etc., and has a highly temporal aspect, i.e., it changes rapidly with time. This information consists of historic information, about what has happened since the disaster occurred, and predicted information, about what may happen. Examples of historical information are the type, scale, and affected area of an incident, the number of injured and missing people, etc. This information is needed to help emergency managers identify dangerous areas that should be avoided. Examples of predicted information are the likelihood of floods in a given 2.5-dimensional terrain, areas threatened by gas plumes, and the forecasted wildfire front, etc. Such information is also needed to assist planners in adjusting original route plans in advance of developing disasters.

For the above reasons, a hazard simulation model that is capable of providing reliable predicted information about disaster changes, is a valuable framework that underlies the solutions for many problems that arise in the context of advance rescue planning. Many disaster models have emerged to encourage and facilitate emergency operations in the past few years (Hu, 2011; Moreno et al., 2012, 2011; Zelle et al., 2013; Lu et al., 2008). For example, Zelle et al. (2013) present an integrated system for smoke plume and gas cloud forecasts, combining a weather model, a smoke plume model and a crisis management system. Moreno et al. (2011) present a real-time fire simulation algorithm that can be integrated into interactive virtual simulations where fire fighters and managers can train their skills. These models make it possible for emergency workers to assess the potential impact of a hazard, identify dangerous areas that should be evacuated, and make effective plans to curb damages and protect lives.

In our research, a geo-Database Management System (geo-DBMS) is selected to manage hazard simulation results and dynamic information of geographic objects. The Geo-DBMS provides efficient management of large spatial data sets (often encountered in large scale events). In addition, it has mechanisms that enable fast update and access to geographic information, and functionality for data analysis. The geometric model, which has been used and implemented in major geo-DBMSs (e.g., Oracle Spatial, PostGIS) (Meijers et al., 2005), makes the systems capable of handling all types of spatial data related to disaster management. Some data models haven been developed in geo-DBMSs for emergency response (Dilo and Zlatanova, 2011; Kwan and Lee, 2005; Zlatanova and Baharin, 2008). However, they are not capable of dealing with predicted information from hazard simulation models and can not support routing among moving obstacles. Many researchers have been working on managing moving objects and numerous data management techniques have been developed to facilitate the collection, organization, and storage of dynamic data of moving objects (Wolfson et al., 1998; Meratnia, 2005; Güting et al., 2006). These studies provide a rich set of solutions for managing the dynamic information produced during disasters, such as the locations of the rescue unit, plume movement, and changes in the water level.

In this paper, we focus on the routing process in a real road network in the case of forest fires. We use a fire simulation model to generate datasets about the spread of the fire, and obtain information about its damage to the infrastructure through spatial data analysis. A spatio-temporal data model is proposed to structure dynamic information of transportation conditions affected by fires in the database. Using this information, we apply a modified shortest path algorithm to calculate optimal paths avoiding fire-affected areas for first responders. Such an approach is not limited to route planning during forest fires, but also can be extended to assist navigation among moving obstacles brought about by other types of disasters.

The organization of the paper is as follows. In section 2, we describe our system architecture for emergency navigation. Section 3 presents both con-

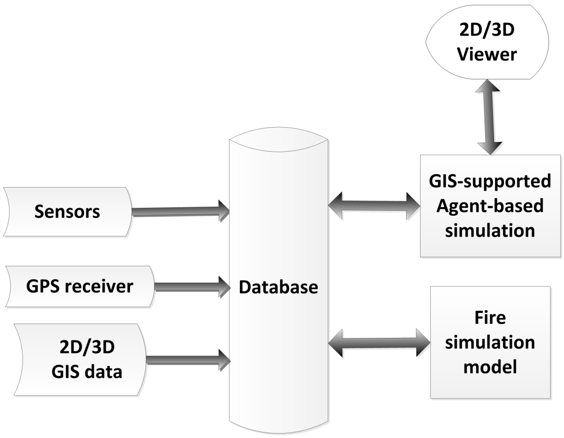

Figure 1: The overview of the proposed system architecture

ceptual and logical spatio-temporal data models of the dynamic information for routing to avoid obstacles. Section 4 illustrates the network analysis application, including the extended A* algorithm. Section 6 describes the detailed implementation of our navigation system. In section 7, we test the model and the algorithm in different scenarios, and detail our results. We draw some conclusions in section 8 and end this paper with proposed future work in section 9.

2. System architecture

To assist fire fighting in forest areas, a system architecture for routing avoiding fire-affected areas is designed. The framework of the proposed system is depicted in figure 1 and is composed of the following components: data collection, data management, fire simulation model, agent-based simulation model and visualization of simulation results. When a fire incident occurs, several measurement teams are formed and sent into the field to perform measurements. Real-time sensor information (e.g., wind speed and wind direction) is collected from the field via a communication network and incorporated into the fire simulation model (Moreno et al., 2012). The fire model produces dynamic data of spatial units about the fire state, from which the shape and direction of movement of fires are derived. This dynamic information, together with the geo-information of the network and the information regarding response units (routes, starting point, end point, status, etc.) is consistently recorded and structured in a geo-DBMS based on the data model designed for emergency response (Dilo and Zlatanova, 2011). We use an agent-based simulator with GIS functionalities to predict the availabilities of roads in a certain area at a certain time, and to display the movement of both the fire and responders. The fire simulation results are represented as one or more moving polygons crossing a certain road network. The first responder is modeled as an agent characterized by a set of attributes (e.g., speed, type of vehicle) and performs certain actions (e.g., moving, waiting). Using predicted information about the status of roads, the path planner, within the agent, applies the shortest path algorithm to calculate the safest and fastest route for responders. The calculated results are visualized to users through a 2D view as well as a navigable 3D view to enhance human situational awareness (Schurr et al., 2005).

3. Data model design

A spatial temporal data model is needed to effectively organize all required information and knowledge in the geo-DBMS. This data model should fulfill the following requirements: (1) support representation of the environment, particularly the network elements and the network topology; (2) support dynamic simulation, such as the representations of disaster developments in time, changes in the availability of roads, and the movements of relief vehicles; (3) support various analyses, including identifying the areas that are most threatened, planning paths in the context of moving obstacles, etc.; (4) support representation of the calculated results, e.g., the navigation route, estimated traveling and arrival time; and (5) should be compatible with the relevant data models for emergency response and existing standards defined by the Open Geospatial Consortium (OGC) or International Standard Organization (ISO), e.g., ISO 19107:2003 that provides a formal structure for representation of spatial objects.

Using the requirements listed above, we define a data model to capture dynamics of the environment, using Unified Modeling Language (UML) profiles for database design. The proposed model is designed adhering to the data model presented by Dilo and Zlatanova (2011) as much as possible, and is built for the following 3 groups of data: (1) data related to the road network; (2) data relevant to disasters; and (3) data on response units. We define the topology of the network by ourselves, and use the geometric data types specified by ISO

19107, e.g., GM_Point, GM_LineString, GM_Polygon, and GM_MultiSurface, to describe the spatial characteristics of geographic features. Because the data we are handling are constantly changing, new data types are created to capture this spatio-temporal nature.

3.1. Conceptual data model

Figure 2 is a UML class diagram presenting a conceptual model of the data required for navigation among moving obstacles. The yellow classes are created for handling the data related to disasters. The green classes are used to support the representation of the road network. The classes in light-gray are defined for modeling the data of response units. New datatypes are colored in purple. The class RoadNetwork is an extended graph, consisting of instances of RoadSegment that contain dynamic information produced by disaster events. To maintain the topology of the road network, an association between RoadSegment and RoadJunction is established. Both RoadSegment and RoadJunction have an attribute affected_time_list used to store temporal information regarding the availabilities of the corresponding spatial objects. A new data type called AffectedTimePeriod is created for these two classes containing the attribute of a dynamic nature. A RealIncident is used to record the information of the disaster incident. It inherits all properties of the abstract class Incident which contains static information of the incident including incidentID identifying the incident, the location of the incident, the start time, and a text description of the incident. Some additional attributes are added to store the dynamic information generated during the incident, such as the disaster type which may change in time, GRIPlevel describing the changing severity of the incident, and affected_area which stores the historic information of affected areas during the incident. The class SimulatedEvent is linked with RealIncident to describe disaster simulations that predict the effect of real incidents within a certain period of time. The class Obstacle contains predicted information about the obstacles in the form of moving polygons affecting the road network. As soon as a real incident occurs, different types of Processes are started. Several teams that are sent to address the incident are responsible for managing these processes. A team may be composed of one or more vehicles. The class Vehicle contains information related to vehicles. The association Follow is used to record the routes that drivers want to follow. These Routes are calculated based on spatio-temporal information in the geo-DBMS and proposed to the drivers. The stored route information will also be used for monitoring movement of vehicles during disasters and analysed after disaster response.

3.2. Logical data model

The proposed data model has been realized in the relational database PostGIS (www.postgis.org). PostGIS spatial data types and functions are compliant with OGC specifications and ISO 19107. Figure 3 shows the logical data model for PostGIS. Following classical approaches (Güting et al., 2000; Güting and Schneider, 2005), we create some new data types to store the spatio-temporal data, i.e., MovingPointInst to store dynamic positions of both vehicles and teams; MovingPolygonInst to record historic affected regions and identify dangerous areas in the near future. These data types are defined by adding timestamps as one of attributes to capture the temporal aspect. We use the ARRAY type, in which the new data types are used as a base type of the array elements, to record facts associated with time. For example, MovingPolygonInst[ ] is composed of a sequence of pairs of polygons and time instances. To represent many-to-many associations, an intersection table is created. For instance, a table, RoadSegment_to_Route, is introduced to hold the many-to-many relationship between RoadSegment and Route, combining the primary keys from the original tables. The logical schema is automatically transformed by a modelling tool Enterprise Architect (www.sparxsystems.com) to a collection of Structured Query Language (SQL) scripts for creating and dropping tables. These created tables are populated with spatial and spatio-temporal data that are used for analysis and visualization by our navigation application as well as traditional GIS tools.

4. Network analysis application considering the spread of the fire

In this study, we design and develop a prototype network analysis application for forest fire rescue planning. The application supports both data processing and data analysis, including fetching the fire simulation results, formatting them into a general representation, calculating the availability of road segments, and computing the shortest path while

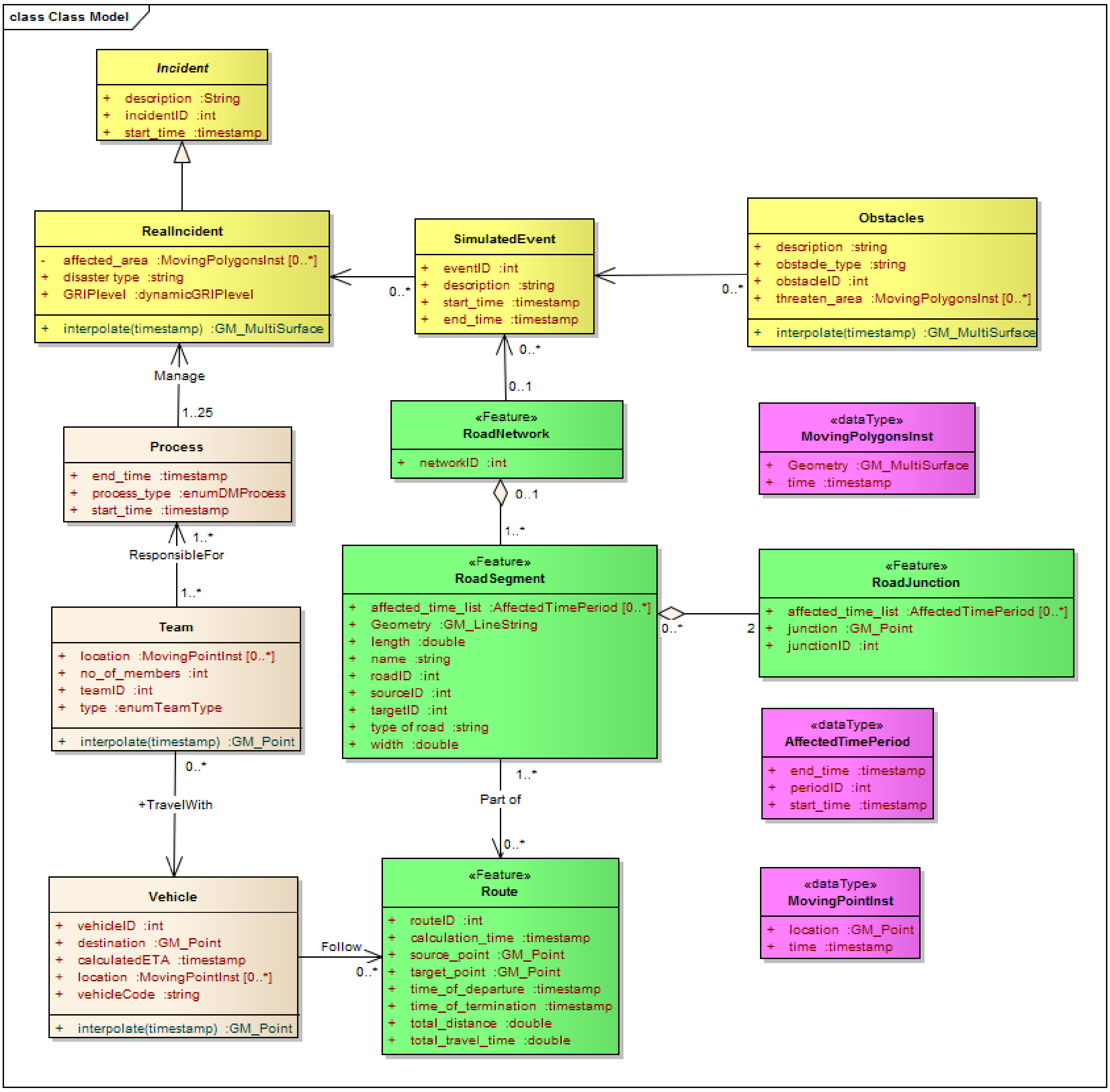

Figure 2: Conceptual data model (UML class diagram with ISO 19107 geometric data types)

avoiding predicted inaccessible roads in fire-affected areas. The shortest path algorithm is extended to consider both static information, i.e., the topological and spatial constraints of the network, and dynamic information, i.e., the predicted accessibility of roads.

4.1. Intersection of the fire-affected area with the road network

For the network analysis application, a cell-based fire simulation model developed by Moreno et al. (2011) is used to generate datasets of fire-affected areas. The fire simulation method divides the topography into a grid of square cells. Each cell contains both static information, such as position, size (i.e., 3 meters), type, and the burning rate depending on its type, and the runtime information, such

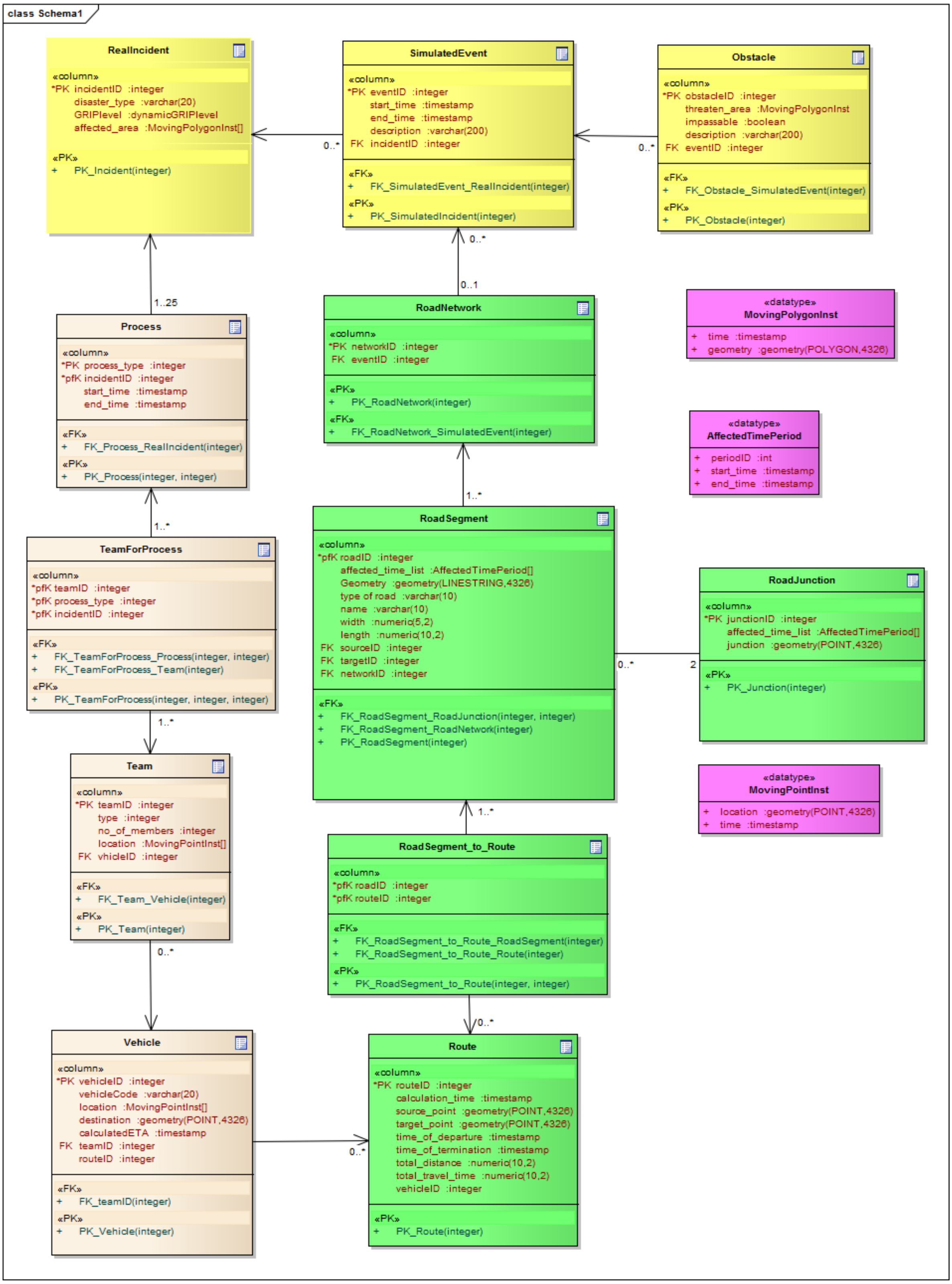

Figure 3: Logical data model (UML class diagram with PostGIS geometric data types, note that the ARRAY is used and indicated by square brackets [ ] after the datatype of the attribute)

as the quantity of combustible, the power intensity of the fire, and the state of the fire. The fire simulation system, integrated with passive data from different sources and dynamic events, including real-time changes in the weather conditions, calculates the spread of the forest fire and updates the run-time information of forest cells calculated during each simulation step. By grouping the cells according to the cell state and time step, we create a set of moving polygons that overlap a certain road network. Considering that each cell in the simulation has a certain width, we introduce a new buffer for each road-center line to represent the road network, extract all the road segments and junctions inside affected areas, and store them with their affected time periods in the database according the data model described in section 3.

4.2. Routing algorithm

Once the state of roads has been updated, the application fetches spatio-temporal data of the road network from the database and generates a graph with affected time of roads. Consider a graph $G = (N, E)$ consisting of a finite set of edges E and nodes N. Each edge $e \in E$ corresponds to an object of class RoadSegment, and each node $n \in N$ corresponds to an object of class RoadSegment. We use w to represent the length of each RoadSegment and use an interval $[t^{closed}, t^{open}]$ to denote an element of affected_time_list attached to the corresponding road segment and junction. $[t^{closed}, t^{open}]$ is an instance of data type AffectedTimeperiod, where t^{closed} is the start time of closing, and t^{open} is the end time of closing. Here we assume that once the nodes and edges are affected by the fire, they will not be available anymore. Following the above assumption, every affected edge and node has only one affected time interval, and the opening time, t^{open}, is set to inf by default. To calculate routes avoiding obstacles, a special algorithm is needed to handle the affected time of roads.

In our application, we have extended the A* methodology for shortest path planning among moving obstacles. Related research on navigation among moving obstacles have been greatly studied in the robotic field. Phillips and Likhachev (2011) introduce the concept of safe intervals to compress search space and extends the A* algorithm to generate time-minimal paths in dynamic environments with moving obstacles. Similarly, Narayanan et al. (2012) use time intervals instead of timesteps and develops a variant of A* for anytime path planning in the presence of dynamic obstacles. However, their planners do not take constrains of the real road network into consideration and can be only applied to free space. Our path planner has some similarities to the algorithms presented in Visser (2009) and Wang and Zlatanova (2013a) which also consider predicted information of the road network and introduce waiting options to avoid moving obstacles. Under the above assumptions, waiting would not be safe during fires and the vehicles need to move as fast as possible. Therefore, we remove the waiting option in the algorithm and do not consider the information on the state of nodes.

The modified A* algorithm

```
Initialize startNode $s$, goalNode $d$, moveRate, departureTime
Initialize openSet, closedSet
$g(s)$ := departureTime
Insert $s$ in openSet
while openSet is not empty do
  $n$ := the node in openSet having the lowest $f$ value
  if $n = g$ then
    return the path from $s$ to $d$
  end if
  Remove $n$ from openSet
  Insert $n$ to closedSet
  for each neighbor $n'$ of $n$ do
    if $n'$ in closedSet then
      continue
    end if
    tentative_cost := $g(n) + w_{nn'}$/moveRate
    $flag$ := false
    if $n'$ not in openSet then
      if tentative_cost $< t_{nn'}^{closed}$ then
        Insert $n'$ to openSet
        $flag$ := true
      end if
    else if (tentative_cost $< g(n')$) and (tentative_cost $< t_{nn'}^{closed}$) then
      $flag$ := true
    else
      $flag$ := false
    end if
    if $flag$ = true then
      the backpointer of $n'$ := $n$
      $g(n')$ := tentative_cost   /* the actual path cost from s to node y */
      $h(n')$ := heuristic_estimate_of_cost($n'$, $d$)
      $f(n')$ := $g(n') + h(n')$
    end if
  end for
end while
return no-path
```

Figure 4: The modified A* algorithm

A* is a well-known algorithm developed to solve the one-to-one shortest path problem (Hart et al., 1968). The A* algorithm uses a heuristic function to estimate cost from each node to the destination to guide path search. The cost associated with a node n is $f(n) = g(n) + h(n)$, where $g(n)$ is the actual cost of the path from the start to node

n, and $h(n)$ is an estimated coast from node n to the destination. The algorithm maintains two sets: *openSet* that stores nodes who are not expanded , and *closedSet* that stores nodes who have been expanded. At each iteration, the algorithm selects node m with the minimal cost from the *openSet* for expansion. All successors of node m that are unexplored will be put in the *openSet* for further expansion.

In our extension of the A*, we take into account the affected time of roads and introduce an additional parameter for the algorithm, the speed of vehicles *moveRate*, to select nodes for expansions. The value of *moveRate* can be obtained in two ways: (1) user configuration; (2) real-time calculation based on the location of vehicles recorded in the database. A new parameter *departureTime* is added to help estimation of arrival time of each node. Figure 4 shows the main structure of the modified A*. When a node n is expanded, we compute the estimated arrival time considering the cost of the edge $w_{nn'}$ and the given speed, *moveRate* (see line 15). At line 18, we use a condition to decide if the successor n' of n should be added to the *openSet*. If the object can safely pass through the edge between the expanded node n and the successor n', i.e., the estimated arrival time is earlier than the closed time of the edge $t^{closed}_{nn'}$, the successor n' will be added into the *openSet* for further expansions. If not, it remains un-explored. The same condition is also applied on line 22, which guarantees that the evaluated node n' should be updated not only with the faster arrival time but also with the safety of passing through the edge nn'.

4.3. Theoretical analysis

Here we sketch the proof of the optimality of the path calculated by our algorithm.

Theorem 1 *When the modified A* selects the goal for expansion, it has found a time-minimal and safe path to the goal node d.*

Proof Were this not the case, the optimal path, P, must have a node n that is not yet expanded (If the optimal path has been completely expanded, the goal would have been reached along the optimal path.). There are then the following two possibilities resulting in the fact that n is not expanded to generate successors: (1) $f(n) > f(d)$; (2) all successors of n cannot be safely reached, i.e. the estimated arrival time is after the closing time of the edge between n and its successor. Because f is non-decreasing along any path, n would have a lower f-cost than d and would have been selected first for expansion before the goal node, which contradicts the first possibility. We assume n' is the successor of n along the optimal path, implying that $g(n) + w_{nn'} < t^{closed}_{nn'}$, which eliminates the second possibility. In the algorithm, the cost on an edge is equal to the time it takes to execute that edge, and whenever a g-value is updated (a shorter path is found), the time value is also updated to the earlier time. Therefore, when the node d is expanded, it is the earliest time we can arrive at the goal node. This is optimal in terms of time cost. We also know that all explored nodes are safely reached, which makes the entire path safe, from the start node to the goal node.

5. Route safety

To evaluate the safety of the route, we provide a method to quantify the safety value of edges and routes. Our method is similar to the one proposed by Shastri (2006) that introduces the margin of safety of nodes, but uses the affected time of edges to evaluate the safety of routes. The safety of each edge is expressed as difference between the time when fires block the edge and the estimated time when the responder arrive at the target node of the edge. Mathematically, the safety of an edge $n_i n_{i+1}$, $S_{n_i n_{i+1}}$, is

$$S_{n_i n_{i+1}} = t^{closed}_{n_i n_{i+1}} - t_{n_{i+1}} \tag{1}$$

Here $t^{closed}_{n_i n_{i+1}}$ is the closed time of edge $n_i n_{i+1}$; $t_{n_{i+1}}$ is the estimated time of reaching node n_{i+1} though edge $n_i n_{i+1}$.

Because the safety of a route mainly depends on the most unsafe edge along the route, the minimum of safety values of edges is selected as the route safety. Let $R = \{n_0, n_1, \ldots, n_k\}$ be one of routes from s to t, where $n_0, n_1, \ldots, n_k$ are the nodes along the route, $n_0 = s$, $n_k = d$. The safety of the entire route can be computed by using the following formula (Shastri, 2006):

$$S_R = min(S_{n_0 n_1}, S_{n_1 n_2}, \ldots, S_{n_{k-1} n_k}) \tag{2}$$

If $S_R > 0$, the route is considered safe; If $S_R <= 0$, the route is considered not safe. The higher the safety value, the more safe the route is. $+\infty$ means the route is completely safe.

Using the above formulas, we can compare the routes calculated by the algorithms to evaluate the proposed algorithm.

6. Implementation

The proposed model and algorithm are realized in a multi-agent simulator, called Mason (Luke et al., 2004, 2005), and are evaluated with a real road network. The data set of the road network is extracted from OpenStreetMap and loaded into the database according to our defined schema in section 3. The fire simulation model (Moreno et al., 2011) calculates the fire spread and the results are also updated into the database and used to create the moving polygons crossing the network. GeoTools (www.geotools.org) is used to fetch the required data from the database to perform the intersection operation and route calculation. The agent simulator displays both the spread of the fire and the movements of relief vehicles. The calculated results are shown to users through both a 2D viewer, which provides an overview of the fire spread and the navigation routes, and a 3D viewer, enabling users to gain accurate impressions of the actual situation. The 3D viewer is built on top of an open source visualization tool, OSM2World (www.osm2world.org) that builds three-dimensional models of the environment from OpenStreetMap data (a snapshot can be found in figure 5). Through the construction of the 3D visualization, situational awareness is enhanced by providing information on the surroundings, such as houses, gardens, etc., that might not initially be included in the street network model.

7. Case study

The model and algorithm have been tested with the road network dataset in San Sebastián, Spain. The network is composed of 1717 edges and 1661 nodes. We simulate several scenarios in which one or more fires take place in a forest located in the eastern part of the city. The fire simulator generates the fire spread dataset within the given area in seconds, starting from time t=0 min to time t=20 min. The information regarding the status of the road network is collected and used for instantiating the model. Paths between locations are calculated by using both the modified algorithm and the classical A* algorithm.

Table 1: Calculated results

	Route ID	Distance (km)	Total travel time (mins)	Route safety (mins)
Speed =20 km/h	R0	2.56	7.7	-1.8
	R1	3.00	9.0	$+\infty$
Speed =30 km/h	R0	2.56	5.1	0.7
	R2	2.56	5.1	0.7
Speed =50 km/h	R0	2.56	3.1	2.7
	R3	2.56	3.1	2.7

Notes:

[1] The vehicles considered in this scenario departure at time t=0 min

[2] R0: The shortest route calculated by the standard A* algorithm

[3] R1: The route calculated by the modified A* algorithm at a speed of 20 km/h

[4] R2: The route calculated by the modified A* algorithm at a speed of 30 km/h (the distance of R2 equals the distance of R0)

[5] R3: The route calculated by the modified A* algorithm at a speed of 50 km/h (the distance of R3 equals the distance of R0)

[6] $+\infty$: This route is completely safe from t=0 min to t=20 min

7.1. Scenario 1: navigation for one responder avoiding one fire-affected area

Considering that different vehicle types have different maximum moving speeds, we compare relief routes for different speeds to evaluate the practical application of our route planner. Table 1 shows the results of our experiments. In the first situation, where the relief vehicle is moving at a speed of 20 km/h, our algorithm and the standard A* algorithm produce different routes, depicted in figure 5. The light blue line is the route calculated by our algorithm, and the brown line represents the shortest path without considering the fire spread. The results indicate that when fires are moving fast and affect the environment rapidly, the vehicle at a speed of 20 km/h can not safely arrive at the destination along the shortest route, because the route could be blocked by fires before the vehicle can pass through. Our algorithm finds a new route that makes the responding unit detour to avoid fires and is safer than the shortest one.

Continuing our analysis, figure 6 depicts another situation in which the shortest path and the calculated route are the same at given speeds of 30 km/h and 50 km/h. As shown in table 1, the vehicle in this situation is moving faster, which leads to a shorter path and less travelling time. The table

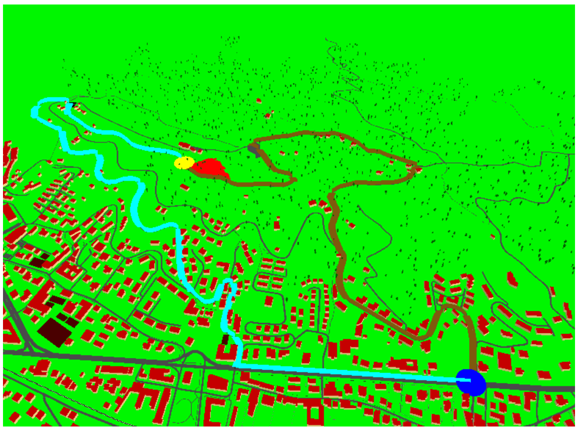

Figure 5: The calculated paths (speed=20 km/h) from the blue point to the yellow point through the environment with one fire-affected area (in red)

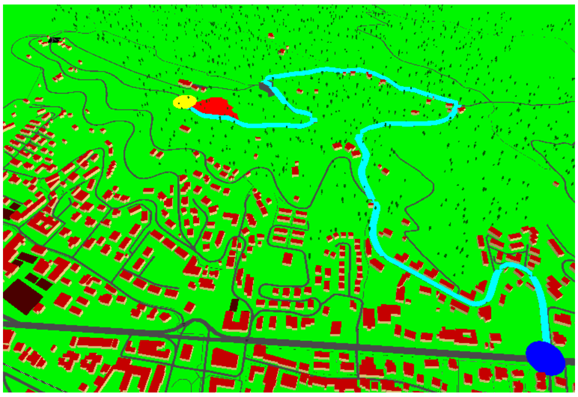

Figure 6: The calculated paths (speed=30, 50 km/h) from the blue point to the yellow point through the environment with one fire-affected area (in red)

1 also indicates the vehicle moving at a speed of 50 km/h has a higher safety value than the vehicle at a speed of 30 km/h. By testing different speeds in the application, the emergency manager can determine the minimum speed required to safely pass through the affected region or to follow a specific route.

7.2. Scenario 2: navigation for multiple responders avoiding multiple-affected areas

In this scenario, we study the navigation case that multiple rescue vehicles have to be routed to one destination avoiding multiple fire-affected areas. The considered vehicles have different maximal speeds, and start moving from different locations at different time instants. Our algorithm calculates routes avoiding fires, considering both the speed of vehicles and their departure times. The calculated results are shown in table 2. Because of the fact that the shortest routes could be blocked by the fires, emergency plans made based on estimation of arrival time of the shortest route will not be feasible due to possible delays. As we can see from the table that, although vehicle 1 can arrive at the destination on time, the time difference between arrival time of the shortest route and arrival time of obstacle avoiding route for vehicle 2 is about 3.5 min, and vehicle 3 has a time difference of 4.5 min. Because responders often work in groups, a reliable estimation of their arrival time at the field site is very important for rapid emergency operations. A lack of consideration of possible delays caused by fires could slow the response process. Figure 7 shows a snapshot of routes calculated by our algorithm. The results indicate that our algorithm can not only deal with multiple fire-affected areas, but also give a more reliable estimation of arrival time for different types of vehicles starting from different places and different time instances, which would make emergency plans more effective and contributes to an improvement of performance of the response units.

Table 2: Calculated results

	Route ID	Departure time (min)	Total travel time (mins)	Arrival time (min)
Vehicle 1	R0	2.0	6.0	8.0
(30 km/h)	R1	2.0	6.0	8.0
Vehicle 2	R2	5.0	5.3	10.3
(20 km/h)	R3	5.0	8.8	13.8
Vehicle 3	R4	8.0	6.5	14.5
(20 km/h)	R5	8.0	11.0	19.0

Notes:

[1] R0, R2, R4: The shortest routes from different sources to the same destination

[2] R1: The route calculated by the modified A* algorithm given a speed of 30 km/h and a departure time t=2.0 min (the route R1 and the shortest route R0 are the same)

[3] R3: The route calculated by the modified A* algorithm given a speed of 20 km/h and a departure time t=5.0 min

[4] R5: The route calculated by the modified A* algorithm given a speed of 20 km/h and a departure time t=8.0 min

8. Conclusions

During forest fires, transportation networks could be damaged by fires spreading and blocking roads

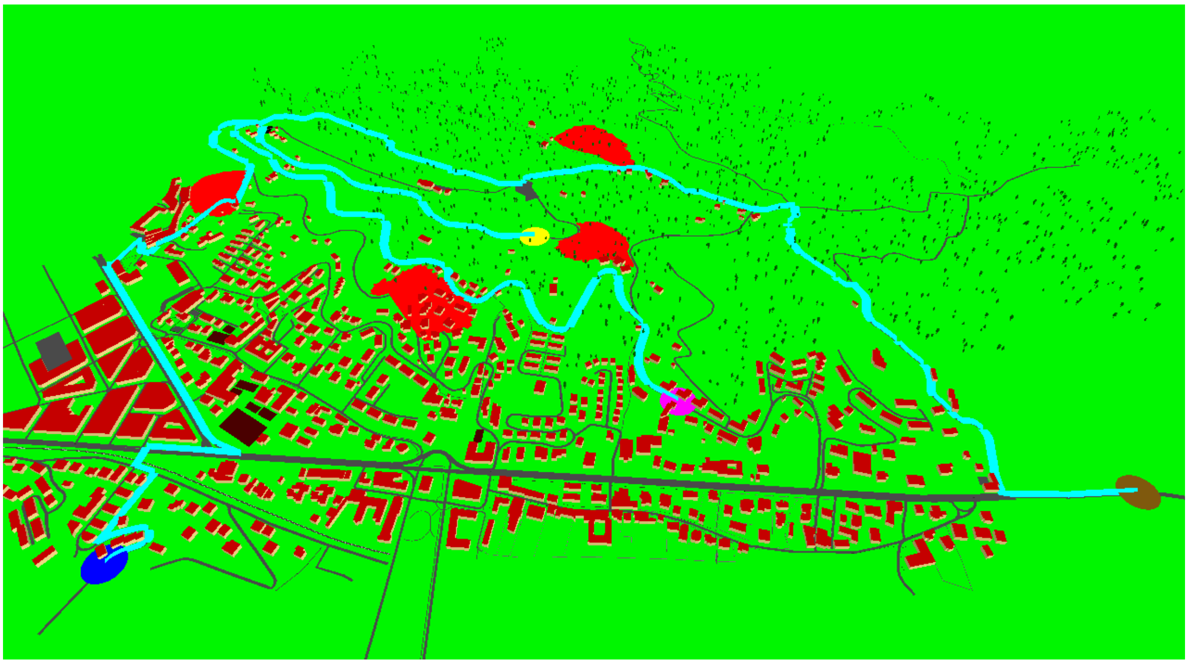

Figure 7: The calculated paths for three vehicles among multiple fire-affected areas (Vehicle 1 from the blue point for the yellow point; Vehicle 2 from the purple point to the yellow point; Vehicle 3 from the brown point to the yellow point)

(Taylor and Freeman, 2010). A system architecture, combining a fire simulation system, GIS-supported agent-based simulation system, and geo-Database Management System (geo-DBMS), is designed to assist in planning paths among moving obstacles caused by forest fires. This paper presents a spatio-temporal data model for the management of both static and dynamic disaster-related information. On the basis of our data model, the geo-DBMS, which is updated constantly, can provide latest and most consistent data required for the network analysis application. In our application described here, we extend the A* algorithm to calculate obstacle-avoiding routes, considering the predicted information regarding the state of the roads. Proof of the optimality of the path computed by our algorithm is also provided.

We apply the prototype system to the case of a simulated fire event. The experimental results indicate that our data model can manage various types of spatio-temporal data, reflect the dynamics of the road network during disasters, and allows relevant data to be appropriately organized to facilitate automated network analysis and dynamic simulation. The application also shows that the extended algorithm, incorporating the dynamic data produced by fire simulations, provides a safer route to the destination, highlighting the importance of the fire model in emergency planning. As demonstrated by our system, the integration of predicted information from the fire simulation can help to avoid one or more obstacles in the environment due to the spread of the fire, offering a promising direction for a wider range of applications.

It should be noted that, although the focus of this paper is on routing fire response units, the developed approach is not limited to fires. Our central goal here is to provide safe and optimal paths avoiding obstacles caused by different disasters. The approach introduced here can be tailored for other types of disasters, e.g., toxic plumes and floods. For example, in the designed data model, obstacles caused by other types of disasters can be also represented as moving polygons; the routing algorithm now considers the state of the edges, but the availability of nodes can also be taken into account if we introduce waiting options to avoid moving obstacles in certain situations.

Currently, the developments do not reflect all aspects of route determination during fire events. Several points should also be mentioned. First, there is not yet a direct connection between our application and the fire model. Because we need only the output data from the fire simulation, we assume that these data have been provided by external software or a simulation system and stored in the database. The integration of the fire model into the application could facilitate the computation and can be performed in later work. Second, our data model only handles data that are essential for emergency navigation. The structuring of the OSM data and the fire simulation output data used by our application is not considered in our data model and is beyond the scope of this paper. Finally, due to a lack of data on the width of roads in our test dataset, we assume all roads have the same width and use it to create road buffers. Because the affected time of roads for routing is obtained based on intersection operations between road buffers and fire affected areas, a data source that contains data on real road width is needed to make calculated route results more reliable.

9. Future work

Despite these promising results, many challenges must still be addressed. One of the most challenging problems is that the behaviors of fires are difficult to capture with the fire simulation model. The predictions, provided by the fire model, have inherent uncertainty, which decreases the effectiveness of our route planning for fire response. The next very important step will be to improve the routing algorithm to consider the accuracy of the fire model.

Because the environment could be simultaneously affected by multiple disasters and is constantly changing, we need a path planner that is capable of processing large volumes of updated data from different hazard models and able to regenerate routes as quickly as possible. Currently, we are building a multi-agent system, exploiting JADE (Java Agent DEvelopment Framework) to support automated data processing and analysis. Based on the technology of the software agents, a collaboration platform for emergency navigation is designed, enabling interoperability between the hazard simulation systems and our network analysis application.

In future work, we will also explore a variety of navigation cases involving multiple responders as well as multiple destinations. Furthermore, we will consider connecting to the simulation model to other types of disasters, e.g., the plume model, the flood model. In the case of toxic plumes, instead of being blocked or non-blocked, the affected road can have a degree of accessibility that depends on the amount of dangerous smoke along the road and also changes over time. In some situations, the responders can wait at certain places for dynamic obstacles to pass to arrive at the destination faster. Therefore, waiting could be an advantageous option for certain types of disasters and should also be considered in the routing process. Another extension of the data model is needed to meet a wider range of informational needs when multiple disasters occur simultaneously. The data model is generic and can be easily adjusted to merge and organize information from models of different types of disaster. Based on using standard Web services, we can further develop an Android navigation application that supports interoperable collaborations between the user and the machine, and apply it to real disaster situations. In this application, a user interface with various styling options will also be designed for different situations, e.g., waiting and moving, day and night, and urgent and non-urgent.

Acknowledgement

This work was supported by COST Action TU0801 “Semantic Enrichment of 3D City Models for Sustainable Urban Development”

References

Chabini, I., Lan, S., 2002. Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks. Intelligent Transportation Systems, IEEE Transactions on 3, 60–74.

Delling, D., Sanders, P., Schultes, D., Wagner, D., 2009. Engineering route planning algorithms, in: Algorithmics of large and complex networks. Springer, pp. 117–139.

Dilo, A., Zlatanova, S., 2011. A data model for operational and situational information in emergency response. Applied Geomatics 3, 207–218.

Ge, S.S., Cui, Y., 2002. Dynamic motion planning for mobile robots using potential field method. Autonomous Robots 13, 207–222.

Gonzalez, J.P., Dornbush, A., Likhachev, M., 2012. Using state dominance for path planning in dynamic environments with moving obstacles, in: Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE. pp. 4009–4015.

Güting, R., De Almeida, V., Ding, Z., 2006. Modeling and querying moving objects in networks. The VLDB Journal 15, 165–190.

Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M., 2000.

A foundation for representing and querying moving objects. ACM Transactions on Database Systems (TODS) 25, 1–42.

Güting, R.H., Schneider, M., 2005. Moving objects databases. Morgan Kaufmann.

Hart, P.E., Nilsson, N.J., Raphael, B., 1968. A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics 4, 100–107.

Hu, X., 2011. Dynamic Data Driven Simulation. SCS M&S Magazine , 662–669.

Huang, B., Wu, Q., Zhan, F., 2007. A shortest path algorithm with novel heuristics for dynamic transportation networks. International Journal of Geographical Information Science 21, 625–644.

Johnson, R., 2008. Gis technology and applications for the fire services, in: Zlatanova, S., Li, J. (Eds.), Geospatial Information Technology for Emergency Response. Taylor & Francis Ltd, London. Geospatial Information Technology for Emergency Response, pp. 351–372.

Kwan, M.P., Lee, J., 2005. Emergency response after 9/11: the potential of real-time 3D GIS for quick emergency response in micro-spatial environments. Computers, Environment and Urban Systems 29, 93–113.

Li, H., Yang, S.X., Seto, M.L., 2009. Neural-network-based path planning for a multirobot system with moving obstacles. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 39, 410–419.

Liu, Y., Hatayama, M., Okada, N., 2006. Development of an adaptive evacuation route algorithm under flood disaster. Annuals of Disaster Prevention Research Institute, Kyoto University 49, 189–195.

Lu, G., Wu, Z., Wen, L., Lin, C.A., Zhang, J., Yang, Y., 2008. Real-time flood forecast and flood alert map over the huaihe river basin in china using a coupled hydrometeorological modeling system. Science in China Series E: Technological Sciences 51, 1049–1063.

Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., 2004. Mason: A new multi-agent simulation toolkit, in: Proceedings of the 2004 SwarmFest Workshop.

Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G., 2005. Mason: A multiagent simulation environment. Simulation 81, 517–527.

Masehian, E., Katebi, Y., 2007. Robot motion planning in dynamic environments with moving obstacles and target. International Journal of Mechanical Systems Science and Engineering 1, 20–25.

Meijers, M., Zlatanova, S., Pfeifer, N., 2005. 3D geoinformation indoors: structuring for evacuation, in: Proceedings of Next generation 3D city models, pp. 21–22.

Meratnia, N., 2005. Towards database support for moving object data. Ph.D. thesis. University of Twente.

Mioc, D., Anton, F., Liang, G., 2008. On-line street network analysis for flood evacuation planning, in: Remote Sensing and GIS Technologies for Monitoring and Prediction of Disasters. Springer, pp. 219–242.

Moreno, A., Segura, Á., Korchi, A., Posada, J., Otaegui, O., 2011. Interactive urban and forest fire simulation with extinguishment support, in: Advances in 3D Geo-Information Sciences. Springer, pp. 131–148.

Moreno, A., Segura, Á., Zlatanova, S., Posada, J., García-Alonso, A., 2012. Benefit of the integration of semantic 3D models in a fire-fighting vr simulator. Applied Geomatics 4, 143–153.

Narayanan, V., Phillips, M., Likhachev, M., 2012. Anytime safe interval path planning for dynamic environments, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4708–4715.

Nedkov, S., Zlatanova, S., 2011. Enabling obstacle avoidance for Google maps' navigation service, in: Altan, Backhause, Boccardo & Zlatanova (Eds.), International Archives ISPRS XXXVIII, 7th, Gi4DM, Anltalya, Turkey.

Neis, P., Singler, P., Zipf, A., 2010. Collaborative mapping and emergency routing for disaster logistics-case studies from the haiti earthquake and the un portal for afrika. Geospatial Crossroads@ GI_Forum 10.

Parker, C.J., MacFarlane, R., Phillips, C., 2008. Integrated emergency management: Experiences and challenges of a national geospatial information provider, ordnance survey, in: Zlatanova, S., Li, J. (Eds.), Geospatial Information Technology for Emergency Response. Taylor & Francis, London. volume 6 of *Geospatial Information Technology for Emergency Response*, pp. 275–310.

Phillips, M., Likhachev, M., 2011. SIPP: Safe interval path planning for dynamic environments, in: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 5628–5635.

Schurr, N., Marecki, J., Lewis, J., Tambe, M., Scerri, P., 2005. The defacto system: Coordinating human-agent teams for the future of disaster response, in: Multi-Agent Programming. Springer US. volume 15, pp. 197–215.

Shastri, J., 2006. Safe navigation during fire hazards using Specknets. Master's thesis. The University of Edinburgh, Edinburgh, UK.

Taylor, M.A., Freeman, S.K., 2010. A review of planning and operational models used for emergency evacuation situations in australia. Procedia Engineering 3, 3 – 14.

Visser, I., 2009. Route determination in disaster areas. Master's thesis. Utrecht University, Netherlands.

Wang, Z., Zlatanova, S., 2013a. An A*-based search approach for navigation among moving obstacles, in: Intelligent Systems for Crisis Management. Springer, pp. 17–30.

Wang, Z., Zlatanova, S., 2013b. Taxonomy of navigation for first responders, in: Progress in Location-Based Services. Springer, pp. 297–315.

Wolfson, O., Xu, B., Chamberlain, S., Jiang, L., 1998. Moving objects databases: Issues and solutions, in: Proceedings of the 10th International Conference on Scientific and Statistical Database Management, IEEE Computer Society. pp. 111–122.

Zelle, H., Wisse, E., Mika, Á., van Tilburg, T., 2013. Smoke plume modeling in crisis management, in: Intelligent Systems for Crisis Management. Springer, pp. 51–66.

Zlatanova, S., Baharin, S.S.K., 2008. Optimal navigation of first responders using dbms, in: Joint conference of the 3rd international conference on information systems for crisis response and management/4th international symposium on geo-information for disaster management, pp. 541–554.