An open-source software system for setting wildfire evacuation triggers

Dapeng Li
Department of Geography
University of Utah
dapeng.li@utah.edu

Thomas J. Cova
Department of Geography
University of Utah
cova@geog.utah.edu

Philip E. Dennison
Department of Geography
University of Utah
dennison@geog.utah.edu

ABSTRACT

Wildfire evacuation triggers are prominent geographic features used in wildfire evacuation practices to facilitate warning and communication during evacuations. When a fire crosses these features, emergency managers issue evacuation orders for the residents in the path of the fire. Computerized modeling of triggers uses geographic information systems (GIS) and fire-spread modeling to calculate a geographic buffer for a given locality with an input evacuation time. The existing trigger modeling method uses several software packages and programs to create trigger buffers, which makes it inconvenient and time-consuming to use in practice. In this work, an open-source software system is designed and implemented for setting wildfire evacuation triggers. The analysis, design, and implementation of the software system are presented. Moreover, a case study that employs the proposed system to create trigger buffers in Julian, California illustrates the software system. The results show that the proposed system is more user-friendly and makes it more convenient to perform trigger modeling in various applications. The open-source nature of the software system could potentially facilitate communication and collaboration between the developers, researchers, and emergency managers.

Categories and Subject Descriptors

D.2.13 [**Software Engineering**]: Reusable Software – *domain engineering*; I.6.3 [**Simulation and Modeling**]: Applications; H.4.2 [**Information Systems Applications**]: Types of Systems – *decision support*

General Terms

Algorithms, Management, Performance, Design, Experimentation, Human Factors.

Keywords

Open-source software, GIS, emergency management, wildfire evacuation, trigger modeling, fire-spread modeling

EM-GIS'15, November 03-06, 2015, Bellevue, WA, USA

DOI: http://dx.doi.org/10.1145/2835596.2835600

1. INTRODUCTION

The wildland-urban interface (WUI) refers to the areas where urban areas and wildlands meet or intermix [17]. In the American West, due to dry climate, long-term fuel accumulation, and seasonal drought, wildfires pose significant risk to residents in the WUI [22]. Wildfires cause significant loss of life and property in the western U.S. each year, and public safety in the WUI has attracted significant research attention [4; 16]. When residents are threatened by a wildfire, incident commanders (ICs) issue protective action recommendations (PARs) to the threatened population. PARs in wildfires can be categorized into evacuation and shelter-in-place (SIP), and the latter can be further classified into shelter-in-home and shelter-in-refuge [6]. Existing scholarship reveals that factors like fire hazard, community context, warning and evacuation time, and policy context impose significant influence on the ICs' protective action decision making [6]. The complexity of these factors overwhelms the ICs and makes it a challenge to choose appropriate protective actions and issue them at the right time [9].

In wildfire evacuation practices, the ICs will use prominent geographic features (e.g., rivers, roads, and ridge lines) as trigger points to facilitate communications during evacuation warnings [3]. When a fire crosses a feature, a pre-designated protective action recommendation will be issued to the threatened residents or firefighters [5]. Note that if enough time is available, the protective action should be evacuation because it generally maximizes safety. When the time available cannot ensure a safe evacuation, SIP could be issued so that residents and firefighters will not be trapped en route. Triggers in wildfire evacuation can be considered as an emergency warning to help time PARs and ensure that residents or firefighters will have enough time to evacuate to safer places. From a systems perspective, triggers take into account the hazard, the threatened population, as wells as evacuation timing and could have great potential in facilitating wildfire evacuation.

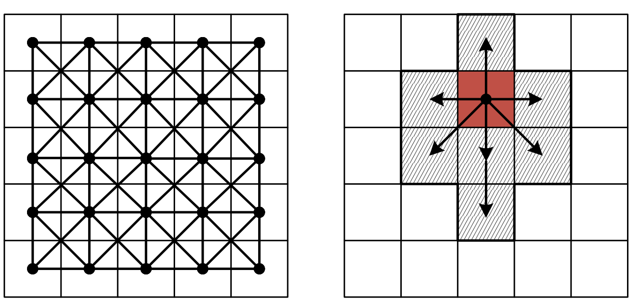
Existing trigger modeling employs the fire-spread modeling software FlamMap to calculate fire-spread rates and uses a separate C program to construct a fire travel-time graph and to create a trigger buffer around the input geographic asset [5]. The current trigger modeling procedure is time-consuming and requires proficiency in FlamMap, which limits its applicability in real-world evacuation practices. Open-source software (OSS) has enjoyed great popularity in software development during the past few years [10]. Specifically, the open-source library fireLib has been widely used for fire spread modeling in the past few years [12; 20]. Its advantage lies in that it can be integrated into software systems to perform fire-spread modeling for various applications more conveniently. Thus, fireLib could be a good alternative for FlamMap in an OSS for trigger modeling.

This work aims to design and implement an OSS system for setting triggers to further improve the applicability of trigger modeling. The remainder of the article is organized as follows. Section 2 first introduces the trigger modeling method using a coupled human-environment system framework. The analysis, design, implementation, and testing of the software system are also presented. Moreover, a case study that employs trigger modeling for strategic evacuation planning for Julian, California is given to illustrate the usage of the software. The results of the case study are given in section 3. Finally, section 4 concludes this article with a discussion of the strengths and weaknesses of the software system and future research directions.

2. METHODS

2.1 Trigger Modeling

Dennison et al. [7] formulated trigger modeling into a three-step procedure: first, the FlamMap software is used to perform fire spread modeling and derive the maximum fire spread rate as well as its direction for each raster cell, and then the fire shape model developed by Anderson [1] is used to calculate the two-dimensional spread rates; in the second step, the spread rates are used to construct a fire travel-time graph that connects all orthogonally and diagonally adjacent cells; third, all the arcs in the graph are reversed and a graph traversal starting from the input cells of threatened geographic assets is performed using the Dijkstra [8] shortest path algorithm. The spatial representation of trigger modeling is shown in Fig. 1. In Fig. 1(a), the grids represent the raster cells, nodes are the centroids of the cells, and the lines are the arcs between adjacent nodes. Note that the arcs are directional and their weights denote the travel time from one cell to the other in that specific direction. Fig. 1(b) illustrates how a trigger buffer is generated. The input geographic assets for trigger modeling are in raster format and can represent geographic features or locations at different scales. For example, the location of a fire crew in the wildland can be represented as a raster cell [5], an evacuation route can be represented as a raster polyline [11], and a threatened residential area a raster polygon [14]. Note that the geographic footprint of the geographic assets depends on its geographic scale as well as the spatial resolution of the raster data used in trigger modeling.



(a) Fire travel-time graph (b) Trigger buffer

Fig. 1 Spatial representation of trigger modeling

The graph traversal process is similar to the process of fire growth. The difference is that all the fire spread rates are reversed and the traversal starts from the cells of the threatened geographic assets instead of fire ignition. When the traversal starts, the algorithm will search the eight neighbors of each input cell and calculate the accumulated travel time for each cell. This expansion will stop until the accumulated time has reached the given input time, which usually denotes the available time for the threatened population located at the input geographic assets to travel to safer places [5]. The algorithm for creating a trigger buffer with a given raster cell and input time is as follows.

```
Variables:
    N: the total number of cells for the raster dataset
    T: input time
    S: the ID of the input raster cell
    A[N][8]: travel time of the arc in eight directions
    B[N]: trigger buffer (initialized to 0)
    C[N]: minimum accumulated travel time (initialized to MAX)
Algorithm:
v = S, B[v] = 1, C[v] = 0, n =0, t = 0 // initialize variables
AC = set( )   // a set to store adjacent cells
for i from 0 to 7                  // iterate each neighbor
    c = getNeighborID(S, i)   // get the each neighbor
    C[i] = A[S][i]                 // update accumulated cost
end for
while(n < N-1 and t <=T)    // set the constraint
    node = -1
    minCost = MAX          // initialize it to MAX
    for  i in  AC               // iterate through AC
        if B[i] == 1           // if cell I has been selected
            if C[i] < minCost
                node = i
                minCost = C[i]
            end if
        end if
    end for
    if minCost <= T      // judge whether T has been reached
        B[node] = 1      // if no add node to the buffer
    else
        break                // otherwise break out of the loop
    end if
    for i in getNeighbors(node)  // iterate through eight neighbors
        AC.add(i)                    // add i to AC
    end for
    t = minCost
    n++                              // increase n by 1
    for i in AC                      // iterate through AC
        if B[i] == 0                 // if cell i is not in the buffer
            cost = getArcCost(node, i)  // get cost from node to i
            if minCost + cost < C[i]    // if a shorter path exists
                C[i] = minCost + cost  // C[i] is updated
            end if
        end if
    end for
end while
```

2.2 System Analysis

Most software construction processes include four typical fundamental activities: specification, design and implementation, validation, and evolution [19]. The first step towards building a software system for trigger modeling is to perform a system requirements analysis so as to define the problem and delineate the scope. Trigger modeling can be considered as a process to take a variety of input data and generate a specific raster buffer around a given geographic feature for a specified time that is derived by estimating the total time needed by the threatened population to evacuate to safe places. In wildfire evacuation practices, fire-spread modeling experts could use fire simulation software like FlamMap and FarSite to simulate fire spread and examine fire behavior, spread rates, and fire perimeters, which could help improve the ICs' situational awareness and facilitate their decision making. Similarly, trigger modeling could also be used by the fire modelers to provide more information for evacuation timing and protective action selection.

Trigger modeling can be considered as a reverse fire-spread modeling process. It takes into account fire-spread rates when generating a buffer around the threatened geographic assets. Trigger modeling has been proved to have great potential in a variety of applications. It is necessary to take into account existing applications into account during requirements analysis so that the new system may improve the usability of trigger modeling in these applications. Existing applications can be categorized into two groups based on the type of the threatened population: residents and firefighters. Trigger modeling has been applied to create trigger buffers for strategic community evacuation planning [7; 14]. Anguelova et al. [2] integrated trigger modeling and pedestrian mobility models to examine wildfire risk on the pedestrians in wildland areas. A recent study by Li et al. [15] employed trigger modeling at the household level to construct evacuation warning zones and stage evacuation warnings. As for firefighter safety applications, Cova, Dennison, Kim and Moritz [5] used trigger modeling to create trigger buffers for a fire crew in an operational manner. Fryer, Dennison and Cova [11] applied trigger modeling to create trigger buffers for a fire crew as well as its evacuation route using different travel modes and weather inputs to avoid firefighter entrapment. All of these applications call for a better software system to enable the users to perform trigger modeling more efficiently and conveniently. And the system should be an OSS system with good scalability so that it can be readily reused and tailored for different specific applications.

The general workflow for the trigger modeling process is shown in Fig. 2. The data for trigger modeling can be readily compiled from the LANDFIRE project [18]. Note that the fire spread modeling step in the old implementation employs a separate software package FlamMap, which makes it inconvenient for applying trigger modeling in various applications. Given that fireLib can be used to perform fire-spread modeling and calculate fire spread rates effectively [20], it can be employed to replace FlamMap and calculate spread rates. After the fire travel-time is constructed, the user can specify the geographic asset as well as input time for trigger modeling. Finally, a trigger buffer can be derived and saved to a file.

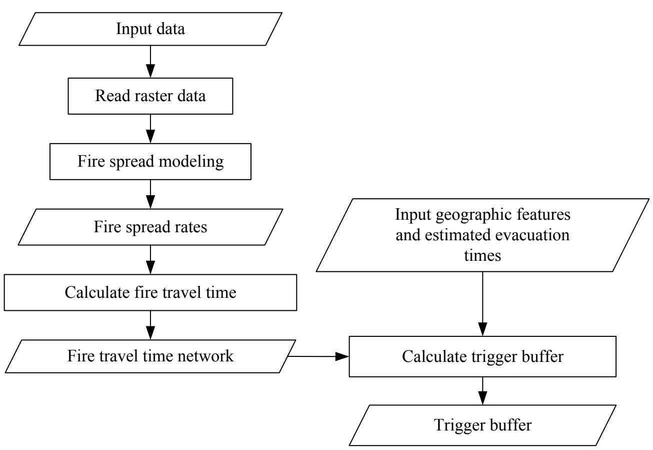


Fig. 2 Workflow of trigger modeling

In addition to the above-mentioned functionalities, there are also some other requirements for the system. First, the system should be designed and implemented as an OSS so that developers from all over the world can contribute to its development. Second, a graphic user interfaces (GUI) should be implemented to facilitate the use of the software. Third, the system should be platform independent so that users could use it on different platforms. Last but not least, the trigger modeling component should be loosely coupled with other components to improve the scalability and extensibility of the system. All the above-mentioned requirements should be taken into account during the system design and implementation phase.

2.3 System Design

Based on the above system analysis, key components of the software system can be defined and designed. As shown in Fig. 3, the system can be divided into three modules: data management, trigger modeling, and visualization. The data management module organizes necessary geospatial data for trigger modeling. Specifically, each project corresponds to a project configuration file, where the information about the raster data such as topographic and fuel model data is stored. And the generated trigger buffer can also be exported to a file. The trigger modeling module is core component of the system. The user can set the input geographic asset and time for trigger modeling. Fire-spread modeling and trigger buffer calculate are designed as two separate functionalities because both of them are computationally intensive and the user could create trigger buffers for different geographic assets and input times without performing fire-spread modeling repeatedly. The visualization module enables the user to examine the trigger buffers.

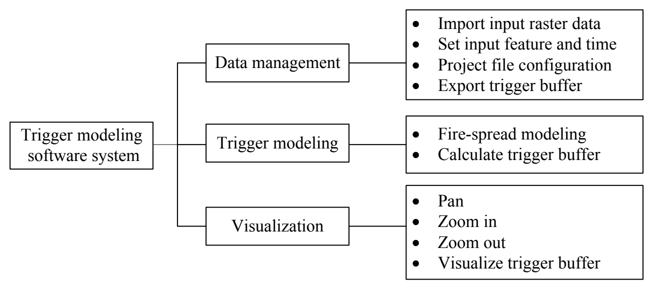


Fig. 3 Modules and functionalities of the system

2.4 System Implementation

The software system was implemented using C+ + and Qt. C++, an object-oriented programming (OOP) language characterized by encapsulation, abstraction, inheritance, and polymorphism [21], has advantages in developing large software systems. Moreover, its computational efficiency also makes it suitable for computationally intensive trigger modeling. Qt is a cross-platform application framework and has been widely used to develop GUI for various software applications. The GUI for the proposed system was implemented using Qt, which makes it more user-friendly compared with the old command-line implementation. Fig. 4 shows the GUI of the system, and the above-mentioned functionalities correspond to relevant menu and tool buttons. The dark raster polygon in Fig. 4 is a trigger buffer generated using uniform topography and south wind. Thus it is skewed towards the south. Note that standardized icons are used for the tool buttons, which could improve the interaction between the user and the system. The user can employ the software to import required raster data and save the configuration to a project configuration file. Specifically, the project configuration file includes the file name and directory information of the raster datasets as well as relevant weather inputs (wind speed, wind direction, and moistures). After the data are imported into the software, the user can perform fire-spread modeling by just clicking one button. The spread rates are calculated and a fire travel-time graph is conducted after fire-spread modeling. Then the user can specify the input geographic asset and the time to create a trigger buffer. Moreover, the generated trigger buffer can be exported to a file. The system also provides basic visualization tools that allow the user to zoom in, zoom out, or pan over the generated buffer to examine it.

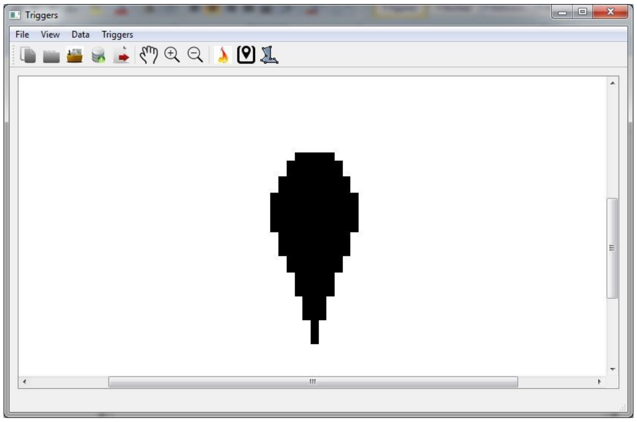


Fig. 4 The GUI of the software system

The implemented system can allow the user to perform trigger modeling using a set of simple standardized operations, which is a significant improvement over the old three-step process. Moreover, the system is cross-platform and can be deployed and used on different mainstream operating systems. Note that the core trigger modeling code is loosely coupled with the user interface, which makes the code easily be reused to build other systems. The source code of the proposed system has been published on a popular web-based code management and version control platform—GitHub, which could significantly facilitate future model development and software production.

3. CASE STUDY

Due to seasonal drought, flammable Mediterranean vegetation species, and the Santa Ana wind, Southern California is one of the areas that are most vulnerable to wildfires in the American West [13]. In order to demonstrate the effectiveness of the proposed software system, a case study using the system for strategic community-level evacuation planning in Julian, California was conducted. Julian, an isolated census-designated place (CDP) in the east of San Diego County, is surrounded by a large amount of flammable fuels, which makes it a good example of many fire-prone communities in the western U.S. All the spatial data (DEM, aspect, slope, and fuel model) needed for trigger modeling were downloaded from the LANDFIRE project. The raster cells with unburnable fuel models in Julian downtown and two communities to the east of downtown—Whispering Pines and Kentwood—were extracted as the input geographic asset for trigger modeling, as shown in Fig. 5.

According to the fuel model data, Julian is primarily surrounded by short grass, shrub, and brush. The software system was used to perform trigger modeling for the residential area in Julian. South wind with a speed of 16 Km/h was used. The 1 h, 10 h, and 100 h dead fuel moisture were set to 6%, and the live fuel moisture for wood and herbaceous fuel types were both set to 65%. 30 min, 60 min, 90 min, and 120 min trigger buffers were generated, as shown in Fig. 6. Note that the trigger buffers are skewed because of the wind as well as the variability in topography and fuel type distribution. If a fire crosses the boundary of the buffer in Fig. 6(d), the residents within the residential area will have two hours to evacuate to safe places before the fire reaches their community. If the total evacuation time for the community is small than two hours, the two-hour buffer could be associated with an evacuation order; otherwise if the total evacuation time is much larger than two hours, a buffer generated with a larger input time should be used as the evacuation trigger buffer.

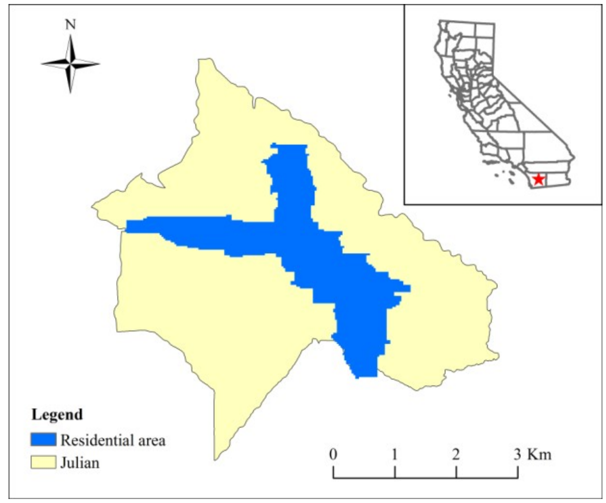


Fig. 5 The residential area of Julian

4. DISCUSSION AND CONCLUSION

This work presents an OSS system for setting triggers based on previous scholarship on trigger modeling. The case study has revealed that the proposed system can be used to calculate trigger buffers more efficiently and conveniently, which can significantly shorten the time needed to perform trigger modeling in wildfire evacuation practices and facilitate the ICs' decision-making process. In addition to the improvement of the usability of trigger modeling in real-world time-urgent circumstances, this work also demonstrates the use of OSS development strategy in professional software production. The OSS strategy can help users, researchers, and developers collaborate with each other in a more convenient and efficient manner. Future work that can help further improve trigger modeling and the proposed software system includes the following aspects.

First, existing trigger modeling methods assume that the ICs could make an estimate of the total time needed for the safe evacuation of threatened population. Given that evacuation traffic simulations have been widely used for estimating evacuation times, further work can focus on how to integrate evacuation traffic simulation into trigger modeling, which will enhance the applicability of trigger modeling in wildfire evacuation practices as well as evacuation planning. Given that real-world data for evacuation are scarce and many assumptions need to be made in evacuation traffic simulation practices, uncertainty in the estimated evacuation times will be unavoidable. Further study should be conducted to examine and model the impacts of the uncertainty of estimated evacuation times on trigger modeling as well as on the ICs' decision-making on protective action recommendations. Moreover, the uncertainty of the weather inputs also poses challenges for the strategic use of trigger modeling for community evacuation planning. Note that modeling uncertainty may need hundreds of runs of trigger modeling. The proposed system could only handle one run at a time, and more work needs to be done to implement a batch processing module to facilitate uncertainty or sensitivity analysis.

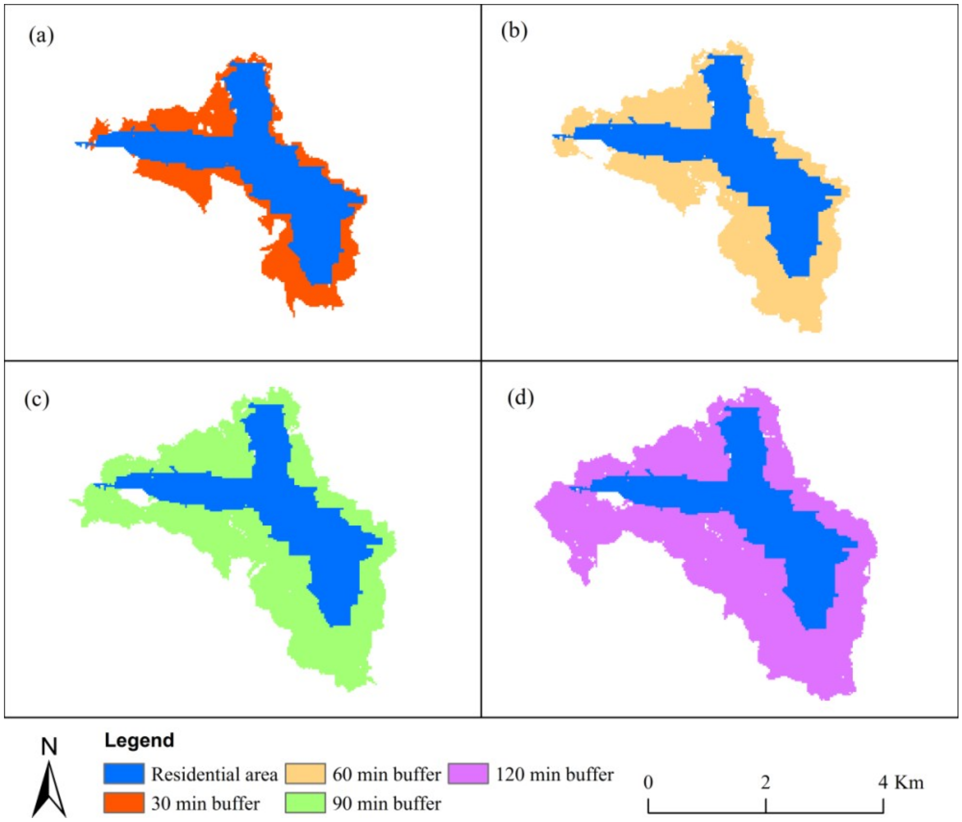


Fig. 6 Generated trigger buffers for different input times

Second, more work needs to be conducted on associating the trigger buffers generated by trigger modeling with prominent geographic features. As noted, prominent geographic features like rivers, ridges, and roads can be used to facilitate communications in wildfire evacuations. Thus, research on how to extend existing trigger modeling by bringing in geographic features will help make trigger modeling more applicable. Further work on this line of research can explore the use of reverse geocoding and digital gazetteers in setting trigger points. Research on reverse geocoding could focus on how to use the boundary of a trigger buffer to retrieve geographic features and set trigger points. Spatial accuracy will be a significant concern for this line of research. As for digital gazetteers, more efforts should be made to evaluate existing publicly available online gazetteers. For example, GeoNames is a popular gazetteer that contains millions of geographic features and could be potentially used to retrieve features to set trigger points. Metrics for evaluating the gazetteers for setting trigger points should be developed to evaluate them. Note that this line of research falls into geographic information retrieval (GIR), and thus existing theories and methods in this field could be borrowed for the evaluation of existing gazetteers. If existing gazetteers cannot satisfy the needs, more work needs be conducted to design and develop a special digital gazetteer for setting trigger points. If these functionalities are implemented and incorporated into the proposed system, the system will be become more applicable and useful in real-world evacuation applications.

Third, modern HPC techniques can be used to implement trigger modeling in the future. The proposed system is a desktop application that could be potentially used by the fire modelers or GIS specialists to support evacuation decision-making during wildfires. The graph traversal algorithm used to create trigger buffers involves heavy computation. When the input geographic asset is a large raster polygon or the raster data are of high spatial resolution, the computation will be too intensive for regular desktop computers. In this case, parallel computing should be employed to perform trigger modeling. Accordingly, relevant parallel algorithms need to be developed, and a systematic study should be done to evaluate its efficiency and effectiveness. We should note that parallel computing using modern HPC clusters could improve computational efficiency significantly but may not be readily integrated into existing software systems. Software as a Service (SaaS) has enjoyed great popularity in the era of cloud computing [23]. Thus, trigger modeling can be deployed to the cloud platform as a service so that it can be easily integrated into various systems. These HPC techniques can be used to significantly reduce the computation time and facilitate time-critical operational applications of trigger modeling.

Lastly, from the software engineering perspective, the open-source nature of the proposed system could potentially facilitate future model development and applications. Compared with traditional software production, OSS can provide a platform for modelers, developers, and users to communicate and collaborate more efficiently. With the popularity of OSS platforms such as GitHub, software development has become a globally collaborative activity. More research can be conducted to further explore how to maximize the advantage of OSS for model development and applications.

In conclusion, the proposed OSS could be used as a tool to perform trigger modeling more efficiently. Its open-source nature and characteristics such as strong scalability could significantly facilitate model development and software production, which could potentially better improve the ICs' situational awareness and support evacuation decision-making during wildfire evacuations.

5. ACKNOWLEDGMENTS

The source code is available on GitHub under the Academic Free License (AFL 3.0) (https://github.com/lidapeng/Triggers). We would like to acknowledge Dr. Robert Keane from the US Forest Service and the developer of fireLib Collin Bevins for their help during system implementation. This research was supported by National Science Foundation CMMI-IMEE grant number 1100890.

6. REFERENCES

[1] Anderson, H.E. 1983. *Predicting wind-driven wildland fire size and shape*. USDA Forest Service, Intermountain Forest & Range Experiment Station, Ogden, UT.

[2] Anguelova, Z., Stow, D.A., Kaiser, J., Dennison, P.E., and Cova, T. 2010. Integrating Fire Behavior and Pedestrian Mobility Models to Assess Potential Risk to Humans from Wildfires Within the U.S.–Mexico Border Zone. *The Professional Geographer 62*, 2 (Mar. 2010), 230-247. DOI= http://dx.doi.org/10.1080/00330120903543756.

[3] Cook, R. 2003. Show Low, Arizona, inferno: evacuation lessons learned in the Rodeo–Chedeski fire. *National Fire Protection Association Journal 97*, 2, 10-14.

[4] Cova, T.J. 2005. Public Safety in the Urban–Wildland Interface: Should Fire-Prone Communities Have a Maximum Occupancy? *Natural Hazards Review 6*, 3 (Aug. 2005), 99-108. DOI= http://dx.doi.org/10.1061/(ASCE)1527-6988(2005)6:3(99).

[5] Cova, T.J., Dennison, P.E., Kim, T.H., and Moritz, M.A. 2005. Setting Wildfire Evacuation Trigger Points Using Fire Spread Modeling and GIS. *Transactions in GIS 9*, 4 (Oct. 2005), 603-617. DOI= http://dx.doi.org/10.1111/j.1467-9671.2005.00237.x.

[6] Cova, T.J., Drews, F.A., Siebeneck, L.K., and Musters, A. 2009. Protective Actions in Wildfires: Evacuate or Shelter-in-Place? *Natural Hazards Review 10*, 4 (Nov. 2009), 151-162. DOI= http://dx.doi.org/10.1061/(ASCE)1527-6988(2009)10:4(151).

[7] Dennison, P.E., Cova, T.J., and Moritz, M.A. 2007. WUIVAC: a wildland-urban interface evacuation trigger model applied in strategic wildfire scenarios. *Natural Hazards 41*, 1 (Apr. 2007), 181-199. DOI= http://dx.doi.org/10.1007/s11069-006-9032-y.

[8] Dijkstra, E.W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik 1*, 1 (Dec. 1959), 269-271. DOI= http://dx.doi.org/10.1007/BF01386390.

[9] Drews, F.A., Musters, A., Siebeneck, L.K., and Cova, T.J. 2014. Environmental factors that influence wildfire protective-action recommendations. *International Journal of Emergency Management 10*, 2, 153-168. DOI= http://dx.doi.org/10.1504/IJEM.2014.066187.

[10] Fitzgerald, B. 2006. The Transformation of Open Source Software. *MIS Quarterly 30*, 3, 587-598.

[11] Fryer, G.K., Dennison, P.E., and Cova, T.J. 2013. Wildland firefighter entrapment avoidance: modelling evacuation triggers. *International Journal of Wildland Fire 22*, 7 (July 2013), 883-893. DOI= http://dx.doi.org/10.1071/WF12160.

[12] Keane, R.E., Drury, S.A., Karau, E.C., Hessburg, P.F., and Reynolds, K.M. 2010. A method for mapping fire hazard and risk across multiple scales and its application in fire management. *Ecological Modelling 221*, 1 (Jan. 2010), 2-18. DOI= http://dx.doi.org/10.1016/j.ecolmodel.2008.10.022.

[13] Keeley, J.E., Safford, H., Fotheringham, C.J., Franklin, J., and Moritz, M. 2009. The 2007 Southern California Wildfires: Lessons in Complexity. *Journal of Forestry 107*, 6 (Sept. 2009), 287-296.

[14] Larsen, J.C., Dennison, P.E., Cova, T.J., and Jones, C. 2011. Evaluating dynamic wildfire evacuation trigger buffers using the 2003 Cedar Fire. *Applied Geography 31*, 1 (Jan. 2011), 12-19. DOI= http://dx.doi.org/10.1016/j.apgeog.2010.05.003.

[15] Li, D., Cova, T.J., and Dennison, P.E. 2015. A household-level approach to staging wildfire evacuation warnings using trigger modeling. *Computers, Environment and Urban Systems 54* (Nov. 2015), 56-67. DOI= http://dx.doi.org/10.1016/j.compenvurbsys.2015.05.008.

[16] Paveglio, T., Carroll, M.S., and Jakes, P.J. 2008. Alternatives to evacuation—protecting Public Safety during Wildland Fire. *Journal of Forestry 106*, 2 (Mar. 2008), 65-70.

[17] Radeloff, V.C., Hammer, R.B., Stewart, S.I., Fried, J.S., Holcomb, S.S., and McKeefry, J.F. 2005. The wildland-urban interface in the United States. *Ecological Applications 15*, 3 (June 2005), 799-805. DOI= http://dx.doi.org/10.1890/04-1413.

[18] Rollins, M.G. 2009. LANDFIRE: a nationally consistent vegetation, wildland fire, and fuel assessment. *International Journal of Wildland Fire 18*, 3, 235-249. DOI= http://dx.doi.org/10.1071/WF08088.

[19] Sommerville, I. 2011. *Software Engineering*. Addison-Wesley, Boston, MA.

[20] Sousa, F.A., dos Reis, R.J.N., and Pereira, J.C.F. 2012. Simulation of surface fire fronts using fireLib and GPUs. *Environmental Modelling & Software 38* (Dec. 2012), 167-177. DOI= http://dx.doi.org/10.1016/j.envsoft.2012.06.006.

[21] Stroustrup, B. 1988. What is object-oriented programming? *Software, IEEE 5*, 3, 10-20.

[22] Theobald, D.M. and Romme, W.H. 2007. Expansion of the US wildland–urban interface. *Landscape and Urban Planning 83*, 4 (Dec. 2007), 340-354. DOI= http://dx.doi.org/10.1016/j.landurbplan.2007.06.002.

[23] Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., Bambacus, M., and Fay, D. 2011. Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing? *International Journal of Digital Earth 4*, 4 (July 2011), 305-329. DOI= http://dx.doi.org/10.1080/17538947.2011.587547.