Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, eds.

A DYNAMIC DATA DRIVEN APPLICATION SYSTEM FOR WILDFIRE SPREAD SIMULATION

Xuefeng Yan

College of Information and Tech., NUAA
No. 29, Yudao Street
Nanjing, Jiangsu, 210016, China

Feng Gu
Xiaolin Hu
Song Guo
Department of Computer Science, Georgia State University
34 Peachtree Street, Suit 1450
Atlanta, GA 30303, USA

ABSTRACT

Wildfire spread simulation plays important roles in wildfire management. Existing wildfire simulations are largely decoupled from real wildfires by making little usage of real time data. In this paper, a dynamic data driven application system is presented to incorporate the real time data into the simulation model, thus to improve the simulation results. The developed dynamic data driven application system is based on the DEVS-FIRE model and employs the particle filtering algorithm to estimate the state of fire spread. We describe the overall structure of the dynamic data driven application system for wildfire spread simulation. The major issues and computation models of this work are presented and experiment results are provided.

1 INTRODUCTION

Wildfires pose great threats to forests, endangered species, and people's lives annually. Some statistics of the loss caused by wildfires can be found in the corresponding scientific and official reports (see e.g., (Rey 2004, Grossi 2007)). To effectively manage wildfires and decrease the losses, many researchers from different fields, such as fire behaviors, geography information system, computer science, etc., strive to seek appropriate models to simulate the process of fire spread. Several wildfire spread models were developed, including FARSITE (Finney 1998), BehavePlus (Andrews et al. 2005), DEVS-FIRE (Natimo et al. 2008), and Hfire (Morais 2001). A common feature of these models is that they simulate the fire propagation assuming we know the landscape data, the fuel data, and the weather data. Their simulation results depend on the accuracy of the data above. Unfortunately, it is impractical to obtain the precise GIS data and the weather data for all time of the simulation period due to the following reasons. Firstly, it is difficult to precisely collect and specify all the landscape data and the fuel data of the fire area as a static input of the system. Secondly, collecting the dynamically changing weather data is challenging, and the weather data (e.g., provided by weather stations) typically do not have enough spatial resolution needed for the wildfire simulation. Thirdly, the coupled effect between wildfire and the weather makes it unrealistic to treat the impact of the weather as a single and static parameter. All of these constrains unavoidably introduce errors to the wildfire spread simulation results.

With the development of sensors technologies and the advance of computer infrastructure, dynamic data driven application systems (DDDAS) become an active research area in recent years. In a DDDAS system, the data from real systems obtained from sensors is fed into the simulation model, which is used to make predictions of the real systems. This dynamic input is treated as the measurement to evaluate the output and adjust states of the model. Based on these measurements, we can evaluate, choose, or analyze the system states utilizing statistical tools, data processing, and numeric or non-numeric techniques to improve the simulation results.

In this paper, we present a dynamic data driven application system for wildfire spread simulation. The wildfire simulation model used in this work is DEVS-FIRE, which is an integrated wildfire spread and containment model based on the Discrete Event System Specification (DEVS) formalism. We are particularly interested in estimating the dynamically evolving fire fronts for better predictions of the fire spread. The measurement data used in this work is assumed to come from ground temperature sensors deployed in the fire area. These sensors regularly provide real time sensor data to be fed into DEVS-FIRE. To deal with the highly nonlinear wildfire behavior, the particle filtering algorithm, also called sequential Monte Carlo methods, are used to support data assimilation of the real time data. Applying the particle filtering algorithm to this problem needs to deal with several challenges including uncertainty modeling, measurement evaluations, the system model and mea-

978-1-4244-5771-7/09/$26.00 ©2009 IEEE

surement model coupling, computation complexity, and computing performances. This paper describes the overall structure and its components of the dynamic data driven application system for wildfire spread simulation.

The rest of this paper is organized as follows. Section 2 briefly introduces related dynamic data driven application systems in various fields. Section 3 describes each component of the dynamic data driven system for wildfire spread model. Section 4 presents some experiments and results. Section 5 draws conclusions and points out the future work.

2 RELATED WORK

Dynamic data driven application systems are widely used in many fields, such as engineering, crisis management and environmental systems, medical, manufacturing, business, and finance (Darema 2007). This is because the new paradigm introduces dynamic observations from real systems into the system model. As said in (Derema 2000), DDDAS is possibly becoming the revolutionary concepts in science, engineering, and management systems.

In the engineering, the mechanism of dynamic data driven simulation is largely used in design and control of systems. In (Farhat et al. 2006), the authors intend to improve active health monitoring, failure prediction, aging assessment, informed crisis management, and decision support for complex and degrading structural engineering systems by using dynamic data driven methods. (Wang 2004) presents a set of data driven hardware and software techniques to explore the input space for performance/energy optimization.

Additionally, in crisis management and environmental systems, the concept of data driven simulation is utilized to incorporate the real time data into the physical models. (Fujimoto et al. 2006) applies dynamic data driven application systems to monitor and manage surface transportation systems. In the paper, the hierarchical DDDAS architecture is presented including vehicle, roadside, and traffic management center simulations. For environmental systems, such as weather, hurricanes, and fire propagation, it is very important for people to effectively predict their states, thus to forecast, control, or suppress them. (Alen 2007) tries to couple the real time sensor information with the water circulation models to forecast the emergency event of hurricane and highlight the challenges for accurate estimations of these events. As discussed in the last section, it is highly necessary for people to study the phenomenon of wildfire spread. Towards this goal, a lot of static models and DDDAS-based fire spread model were proposed. In (Douglas et al. 2006, Coen et al. 2007), the authors select Monte Carlo methods as tools to investigate fire behaviors of wildfire spread models. Although their work are based on the physical models, reaction-diffusion-convection partial differential equation-based model, and the level set method model, their final aim is to provide a framework for dynamic data driven application system of the fire models, which will consider the real time weather data, mapping images, and sensor streams as the measurements to adjust the original models so that more accurate predictions could be made. This work involves in large cooperation among researchers of different fields, and can potentially improve the fire estimation from multiple aspects. In our paper, we feed the fire sensor data to DEVS-FIRE, the integrated environment of fire spread and containment, thus to improve the fire predictions.

In the medical area, the dynamic data driven simulation could be used for illness treatment. In the laser treatment of cancer, the higher intensity heat source may be used to ablate the affected tissue. In order to precisely control the treatment process, the heat transfer computational model is developed to employ the real time data to optimize the control of the treatment. This dynamic data driven application system is capable of estimating and guiding the computer controlled temperature in the biological domain with very good accuracies (Oden et al. 2006).

Other important application domains of dynamic data driven application system are manufacturing, business, and finance. Flexible manufacturing systems, such as mass customization, require handling the product variety, uncertainty in the product demands, and reconfiguration of manufacturing resources. These requirements are just the purposes of dynamic data driven application systems. When the requirements changed, the model can adjust and reconfigure corresponding resources according to the latest analysis. Details of this kind of model can be found in (Qiao et al. 2003). With the increasing complexity of supply chain in enterprises, we need to resort to computer simulation to optimize the business process. To overcome the limitations of the conventional model, which is useful to experts, people need to simplify the model for use. Therefore, (Tannock et al. 2007) imports the data driven concept to automatically construct the model by the input data from the company. This provides multiple scenarios for employees to make their decisions.

From the recent work above, we can see that most applications with plentiful real time data are the potential fields, in which the dynamic data driven simulation would be deployed, for example, stock prediction with a huge real time data. Although there are many challenging tasks, such as the non-linearity of the data, the large computation complexity, statistical tools to process and analyze data, the dynamic data driven simulation is still a very active research topic in modeling and simulation. To improve our static fire spread model, we also seek DDDAS approach as our solution to well predict the fire states of DEVS-FIRE.

3 DYNAMIC DATA DRIVEN SIMULATION FOR DEVS-FIRE

As explained above, we also introduce the concept of dynamic data driven into the fire spread model of DEVS-FIRE to improve the estimation of the fire growth. Figure 1 shows the structure of the dynamic data driven system for DEVS-FIRE, which couples several components to work together for better predictions.

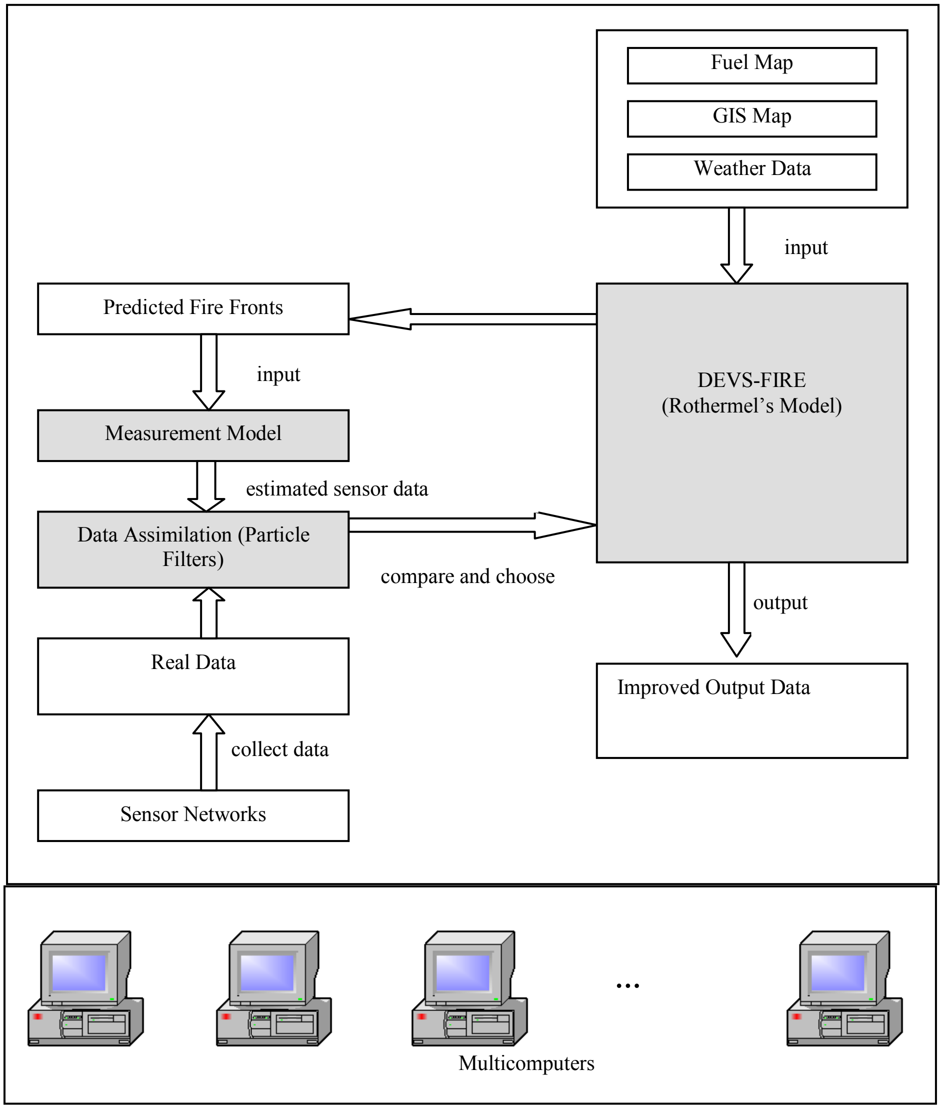


Figure 1: The structure of the dynamic data driven system for DEVS-FIRE

From Figure 1, we can see that the key components of the dynamic data driven system for DEVS-FIRE are DEVS-FIRE, the measurement model, and data assimilation. Basically, we use the physical model of DEVS-FIRE to predict the fire spread if there are the input sets including the fuel map, the GIS map, and the weather data available. This static fire spread model is the basis of fire prediction. The dynamic portion of the system comes from the measurement model, in which fire sensors are deployed in the fire space and the corresponding data is collected. Feeding these data into the measurement model, we output the data to the data assimilation system, which is compared with the real data from sensor networks. Consequently, the data assimilation system chooses the optimal fire states as the inputs of DEVS-FIRE, continuing to execute DEVS-FIRE to predict the next fire state. Because we consider the uncertainty of fire states in the system, which will introduce a lot of computations, the whole system is constructed on the multicomputer environment, in which multiple computational units run the simulations in parallel to improve the performance.

3.1 DEVS-FIRE Model

DEVS-FIRE, an integrated environment for surface wildfire spread and containment, is built on Discrete EVent Specification (DEVS) formalism. This fire area is modeled as a cellular space, in which each cell represents a sub-area of the fire. The fire spread model of DEVS-FIRE can be denoted as Y_i=DF(s, a, f, w, $Y_{i\text{-}1}$), where Y_i is the output, a set of parameters to indicate the fire states including rates of speed, the burning area, the burning perimeter, and the visual fire map at time i; s, a, f refer to the slope map, the aspect map, and the fuel map of the fire space respectively; w is the weather data during the time i-1 to i; $Y_{i\text{-}1}$ represents the fire states at time i-1; DF decides how the fire spreads in the period of time i-1 to i based on the fire behavior model and DEVS formalism.

From the description above, we know that this model is deterministic, which we obtain the unique prediction of the fire spread if any set of input parameters is given. Although the model incorporates the fire behavior models, the GIS map, the fuel map, and the weather data into the system to provide a tool to estimate the fire states, it is still hard for people to precisely forecast the fire growth. This is because the real fire is much more complicated than the abstracted one using numeric and non-numeric methods. The more practical scenario is that we could obtain a series of data from the spot, such as, aerial images, radiation, heat, or temperature information. The information is very important for us to well know the real situation of the wildfire. Therefore, how to collect useful real time data to be utilized in the static model would be a key portion of the dynamic data driven system for DEVS-FIRE.

3.2 Measurement Model

Real time data comes in the timely manner. From the growth of the fire, we can detect many kinds of data from the real fire. With the development of wireless sensor networks technologies, it provides the possibility for people to acquire real time data in time. We can deploy ground temperature sensors and radiation sensors or capture aerial images to collect the useful data used in the fire prediction. It is a crucial step to map data between the static output obtained by the original physical model and the real time data, which can be executed by the measurement model. Actually, in order to predict the fire propagation as accurately as possible, we try to use every kind of real data we can obtain. Therefore, conceptually, the measurement model should intelligently deal with all kinds of the data, implementing multiple mappings. At the current stage, we just consider the temperature mapping from the predicted fire map to the temperature map in our system.

The measurement model can be formalized as M_i =(TY_i, RY_i, AI_i)=MM(Y_i), where M_i is the measurement outputs, which are a collection of outputs mapped from all kinds of data, TY_i, RY_i, AI_i, which mean the temperature, the radiation, and the aerial image measurement outputs respectively at time i; Y_i is the predicted outputs by the original static model at time i; MM realizes the mapping from the predicted outputs to the measurements. For the specific case, we define the mapping from the predicted fire map to the temperature map as TM=$F2T$(FM, ST, δ), where TM is the temperature map, represented by a matrix, in which each element is the value of the temperature; FM is the fire map, which is obtained from DEVS-FIRE to indicate each cell is ignited or not in the whole space; ST is defined as the deployment strategy of the sensors; δ means the transition from the ignited cells to the temperature of any cell in the space; $F2T$ implements the entire mapping process. To explain this mapping, we consider the following example in Figure 2.

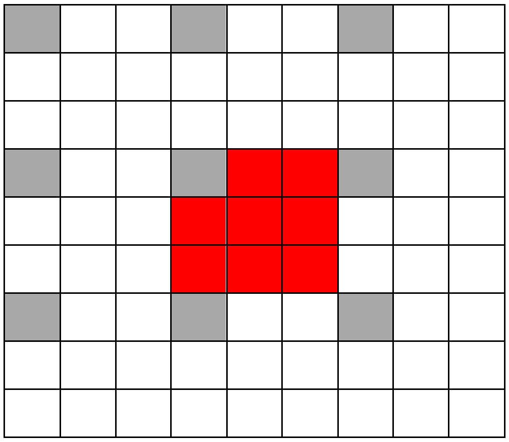


Figure 2: The example for the measurement model

In the figure, we assume that we have a 9×9 cell space, and each cell has the size of 15 meters. At some time step, the fire spreads as shown in the figure (red cells are burning). The sensor deployment strategy is to place a temperature sensor every three cells. Now let's define δ. Assume each burning cell has the temperature of 376°F (Long 2008), and the temperature of a cell decreases with the increase of the distance from this cell to burning cells with the function $T = 376e^{\frac{-d^2}{2\sigma^2}}$, where T, d, σ are the temperature, the distance, and a constant. In this case, we use the distance from the cell to the closest ignited cell, and the constant of 50. According to the mapping function, we obtain the temperature map corresponding to the fire map *TM*={122, 240, 240, 240, 359, 359, 240, 359, 344}.

3.3 Data Assimilation

Generally speaking, there are two main approaches to approximate the dynamic systems, which are linearizing the model (e.g., Kalman filters in (Kailath et al. 2000)) and particle filters. Others include their extended versions or their combinations for specific cases. In this work, due to the nonlinear behavior of wildfire, particle filters are chosen as the data assimilation method. Particle filters, also called sequential Monte Carlo methods, are a numerical approach to approximate conditional filtering distribution. We generate samples from target density according to related algorithms including SIR (Sampling Importance Resampling), perfect sampling, importance sampling, acceptance-rejection sampling, and Metropolis-Hastings independence sampling, which can be found in (Liu et al. 1998; Bølviken et al. 2001; Chib et al. 1995). Normally, we intend to apply SIR algorithm to the model because the whole process is independent on initially generated samples. The possible applications of particle filters in DEVS-FIRE can be found in (Gu and Hu 2008).

3.4 Computational Architecture

In the dynamic data driven system for DEVS-FIRE, there exist a large amount of computations. This is because of the structure and mechanism of DEVS formalism and modeling the uncertainty into the system. The number of particles has a large impact on the data assimilation of the model. Aiming to this purpose, we use the architecture of multiple computers. In the application, we assign the computations of different particles to a couple of computers, and collect the outputs of DEVS-FIRE from them to deal with after a period of execution time. Obviously, this is the client-server software architecture. The server side assigns computational tasks to the client sides, and collects the results to process. The client computers execute DEVS-FIRE, and send their predictions to the server computer.

4 EXPERIMENTS AND ANALYSIS

Based on the dynamic data driven system for DEVS-FIRE above, we construct the experiments to look at its working mechanism. Because it is hard for us to collect the real time data and the corresponding fire growth, we use DEVS-FIRE itself to generate data as the real time data, thus to test the dynamic data driven system for DEVS-FIRE. In the experiment, we use the GIS map from Texas, and assume the weather data is that the wind speed and the wind direction change to random values (4~12 miles/hours and 130~230 degrees) every 10 minutes. The time step is set to 1200 seconds, and we run the simulation of DEVS-FIRE for 9 steps. Therefore, the temperature map for each time step would be the real time data to be utilized in the dynamic data driven system. For different deployment policies, we use the cases of 1 sensor every 5 cells and 1 sensor every 10 cells. Actually, because the weather data is not accurate (wind direction: 180 degrees; wind speed: 8 miles/hour), it will generate errors between these two simulations. Based on the configurations above, two sets of experiments were done using the client-server software architecture to improve the fire prediction. We will present and explain the related results as follows.

Firstly, we run the simulation to obtain the sensor temperatures used as the real time data in the dynamic data driven system for DEVS-FIRE. Figure 3 displays the fire propagation at time step 9. If we use static prediction model, because of the errors from the weather data, the fire spreads as shown in Figure 4.

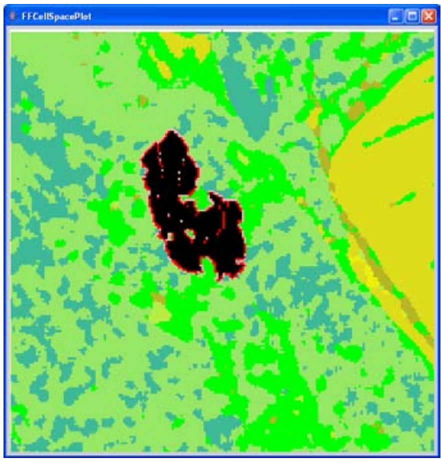


Figure 3: Fire growth as real time data

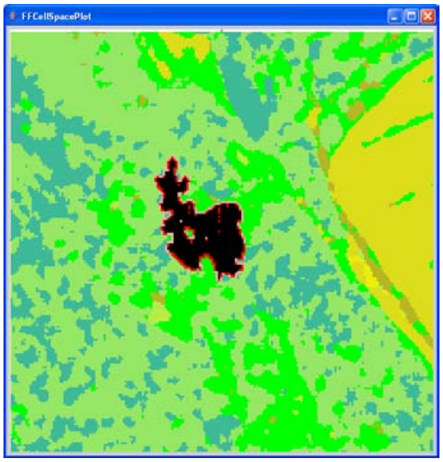
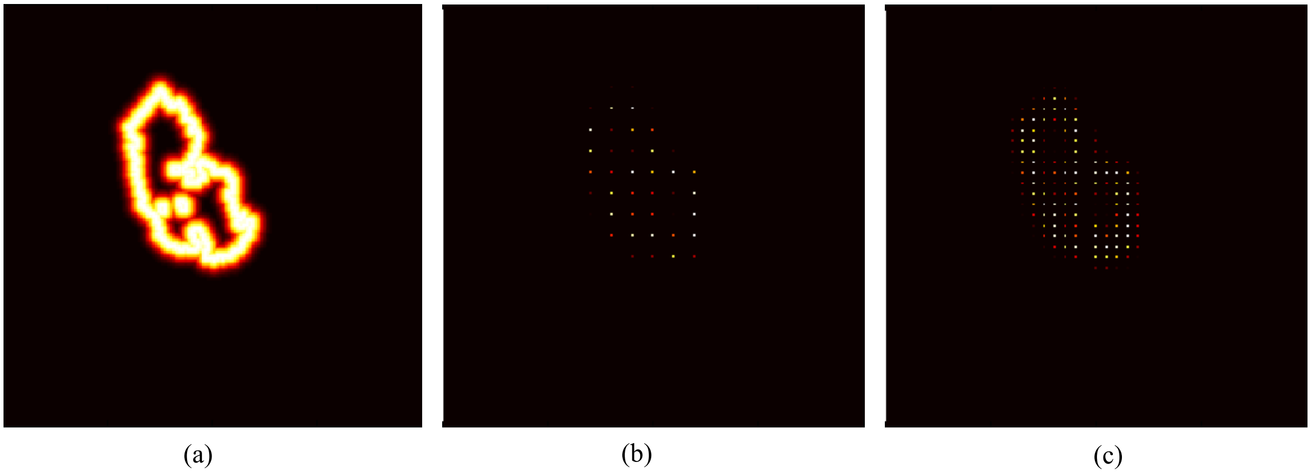


Figure 4: Fire growth using static prediction model

From Figure 3 and Figure 4, we can know that there are large prediction errors between these two fire propagations. We use different sensor deployment strategies as shown in Figure 5 to dynamically estimate the fire. Figure 5(a) shows the temperature map in the ideal case of placing one senor every cell. Figure 5(b) and Figure 5(c) display the temperature maps for two different deployment strategies, which are one sensor every 10 cells and every 5 cells.


(a)

(b)

(c)

Figure 5: Temperature maps of different sensor deployment strategies

Figure 6 and Figure 7 display two fire predictions using dynamic data driven system for DEVS-FIRE based on two sensor deployment strategies as shown in Figure 5(b) and Figure 5(c) respectively.

From Figure 5(a), Figure 5(b), and Figure 5(c), we can see that when there are more sensors, more information can be obtained, although all these three cases are hard for us to well estimate the fire shape. Comparing Figure 3, Figure 4, Figure 6, and Figure 7, we clearly know that these two fire growths using dynamic data drive system for DEVS-FIRE improve the fire estimation, and for the sensor deployment strategies, the case, in which we place one sensor every 5 cells, has better predictions.

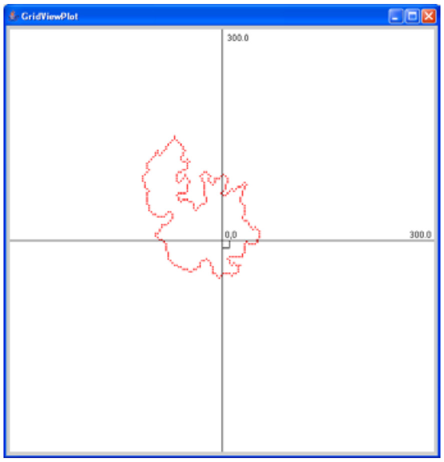


Figure 6: Dynamic fire state prediction (1 sensor/10 cells)

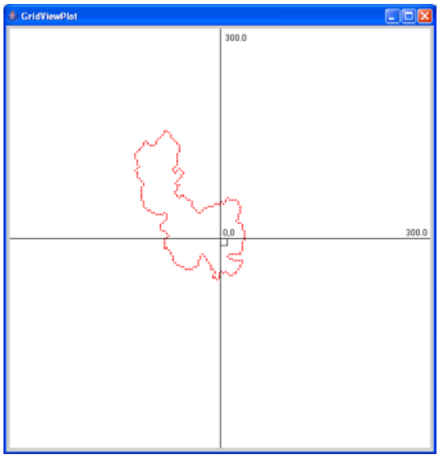


Figure 7: Dynamic fire state prediction (1 sensor/5 cells)

5 CONCLUSIONS

This paper presents a dynamic data driven application system for wildfire spread simulation. The wildfire spread simulation is based on the DEVS-FIRE model. The overall structure of the dynamic data driven application system and its main components are described. Experiment results show that by incorporating real time data, the dynamic data driven application system is able to estimate the evolving fire fronts, and thus to provide more accurate fire spread predictions. The future work will proceed along the following directions: (1) Improve the performance to support uncertainty generation; (2) Import other types of real time data such as aerial images; (3) Investigate other methods to be applied in data assimilation to decrease the estimation errors.

REFERENCES

Allen, G. 2007. Building a dynamic data driven application system for hurricane forecasting. In *proceedings of the 7th international conference on Computational Science*, 1034-1041. Berlin, Heidelberg: Springer-Verlag.

Andrews, P. L., C. D. Bevins, and R. C. Seli. 2005. BehavePlus fire modeling system, Version 4.0: User's Guide. Gen. Tech. Rep. RMRS-GTR-106WWW Revised. Ogden, UT: Department of Agriculture, Forest Service, Rocky Mountain Research Station, 132 p.

Bølviken, E., P. J. Acklam, N. Christophersen, and J. M. Størdal. 2001. Monte Carlo filters for non-linear state estimation. *Automatica* 37(2): 177–183.

Chib, S., and E. Greenberg. 1995. Understanding the Metropolis–Hastings algorithm. *The American Statistician* 49(4): 327–335.

Coen, J. L., J. D. Beezley, L. S. Bennethum, C. C. Douglas, M. Kim, R. Kremens, J. Mandel, G. Qin, and A. Vodacek. 2007. A wildland fire dynamic data-driven application system. In *11th Symposium on Integrated Observing and Assimilation Systems for the Atmosphere, Oceans, and Land Surface* (IOAS-AOLS), CD-ROM, JP3.12.

Darema, F. 2000. Dynamic Data Driven Application Systems (Symbiotic Measurement & Simulation Systems): A new paradigm for application simulations and a new paradigm for measurement systems. Available via < http://www.nsf.gov/cise/cns//dddas/dd_das.htm> [accessed August 11, 2009].

Darema, F. 2007. Introduction to the ICCS 2007 workshop on dynamic data driven applications systems. In *proceedings of the 7th international conference on Computational Science*, 955-962. Berlin, Heidelberg: Springer-Verlag.

Douglas, C. C., R. A. Lodder, R. E. Ewing, Y. Efendiev, G. Qin, J. Coen, M. Kritz, J. D. Beezley, J. Mandel, M. Iskandarani, A. Vodacek, and G. Haase. 2006. DDDAS approaches to wildland fire modeling and contaminant tracking. In *Proceedings of the 2006 Winter Simulation Conference*, eds. L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 2117-2124. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Gu, F., and X. Hu. 2008. Towards applications of particle filters in wildfire spread simulation. In *Proceedings of the 2008 Winter Simulation Conference*, eds. S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler, 2852-2860. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Farhat, C., J. Michopoulos, F. K. Chang, L. J. Guibas, and A. J. Lew. 2006. Towards a dynamic data driven system for structural and material health monitoring. In *Proceedings of the 6th international conference on Computational Science*, 456-464. Berlin, Heidelberg: Springer-Verlag.

Finney, M. A. 1998. FARSITE: Fire area simulator–model development and evaluation, United States Department of Agriculture Forest Service Rocky Mountain Research Station Research Paper, RMRS-RP-4 Revised March 1998, revised February 2004.

Fujimoto, R. M., R. Guensler, M. Hunter, H. K. Kim, J. Lee, J. Leonard, M. Palekar, K. Schwan, and B. Seshasayee. 2006. Dynamic data driven application simulation of surface transportation systems. In *Proceedings of the 6th international conference on Computational Science*, 425-432. Berlin, Heidelberg: Springer-Verlag.

Grossi, P. 2007. The 2007 U. S. wildfire season lessons from southern California. Available by <http://www.rms.com/Publications/2007_US_Wildfire_Season.pdf> [accessed August 11, 2009].

Kailath, T., A. H. Sayed, and B. Hassibi. 2000. Linear estimation. *Information and System Sciences Series*. Upper Saddle River, New Jersey: Prentice-Hall, Inc.

Liu, J. S., and E. Chen. 1998. Sequential Monte Carlo methods for dynamics systems. *Journal of American Statistical Association* 93: 1032–1044.

Long, A. 2008. Available via <http://www.forestencyclopedia.net/p/p498/?show_cited=1> [accessed August 11, 2009].

Morais, M. 2001. Comparing spatially explicit models of fire spread through chaparral fuels: A new model based upon the Rothermel fire spread equation, MA Thesis, University of California, Santa Barbara.

Natimo, L., X. Hu, and Y. Sun. 2008. DEVS-FIRE: Towards an integrated simulation environment for surface wildfire spread and containment. *SIMULATION*: *Transactions of The Society for Modeling and Simulation International* 84(4): 137-155.

Oden, J. T., K. R. Diller, C. L. Bajaj, J. C. Browne, J. Hazle, I. Babuska, J. Bass, L. F. Demkowicz, Y. Feng, D. Fuentes, S. Prudhomme, M. N. Rylander, R. J. Stafford, and Y. Zhang. 2006. Development of a computational paradigm for laser treatment of cancer. In *Proceedings of the 6th international conference on Computational Science*, 530-537. Berlin, Heidelberg: Springer-Verlag.

Qiao, G., F. Riddick, and C. McLean. 2003. Data driven design and simulation system based on XML. In *proceedings of the 2003 Winter Simulation Conference*, eds. S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, 1143-1148. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Rey, M. 2004. USDA Testimony to Congress, 2004. Available via < http://www.fs.fed.us/congress/108/house/oversight/rey/050504.html> [accessed August 11, 2009].

Tannock, J., B. Cao, R. Farr, and M. Byrne. 2007. Data-driven simulation of the supply-chain--Insights from the aerospace sector. *International Journal of Production Economics* 110(1-2): 70–84.

Wang, W. 2004. Data-driven techniques for hardware and software synthesis for embedded systems. Ph.D. Dissertation, Princeton University, USA.

AUTHOR BIOGRAPHIES

XUEFENG YAN is an associate professor in the College of Information Science and Technology at Nanjing University of Aeronautics and Astronautics. His research interests include modeling and simulation, computer networking, and grid computing. His email is <xuefeng.yan@gmail.com>.

FENG GU is a Ph.D. candidate in the Computer Science Department at Georgia State University. His research interests include modeling and simulation and system validation and calibration. His email is <fgu@student.gsu.edu>.

XIAOLIN HU is an assistant professor in the Computer Science Department at Georgia State University. His research interests include modeling and simulation, autonomous agent and multi-agent systems, and simulation-based development. His email is <xhu@cs.gsu.edu>.

SONG GUO is a Ph.D. candidate in the Computer Science Department at Georgia State University. His research interests include modeling and simulation and high performance computing. His email is <sguo3@student.gsu.edu>.