From Static to Dynamic Models: Enabling Real-Time Geocomputation Infrastructures

Mallikarjun Shankar, Cheng Liu, and Budhendra Bhaduri
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831

Email: {shankarm, liuc, bhaduribl}@ornl.gov

Abstract

We consider the challenge of incorporating existing and legacy application models into real-time geocomputation infrastructures. Several steps of translation and pre-processing are required before data can be consumed by models that have so far assumed a statically derived data set or must use coarse information due to computational constraints. A systematic procedure to provide dynamic data to legacy models saves software rewrite costs and enables the infrastructure to minimize communication costs of sending raw data to the back-end models. We describe our system that filters and aggregates real-time data before it sends it to back-end models. We focus on the location (within the networking infrastructure) of the computation and communication steps required for the pre-processing of raw data. We convert the pre-processing needs of a specific class of legacy models, whose spatio-temporal data requirements can be identified in a process-flow graph, into processing modules distributed in our infrastructure. We choose a placement of the processing modules directed at minimizing communication costs.

1. Introduction

Geocomputing practitioners often find that data sensed and incorporated in real-time offers additional value to their analysis and simulation models. Live geospatial data provides an immediacy and relevance to applications in domains such as situational awareness, population dynamics, transportation, and evacuation planning (Franzese, 2001). The availability of distributed computer and sensor networks continues to drive the development of real-time and dynamic computational infrastructures that respond to current data and provide actionable information to users in the field. Legacy constraints may, however, force existing applications to accept static data sets in a batch mode. Additionally, the best existing models may not be able to handle the computational requirements of processing in real-time the cumulative set of detailed fine-grained data being made available by the sensor networking infrastructure. In this paper, we show how a real-time sensor network infrastructure integrates with existing (and legacy) application models, transforms (pre-processes by spatial and temporal aggregation) dynamically available data to make it suitable for the application models and tools. The computed results are presented to users at control centers and returned through the infrastructure to remote subscribers in the field.

We focus on a system scenario that relies heavily on geospatial information to highlight the computational constraints and approaches required when handling spatio-temporal data within a distributed infrastructure. We refer to our end-to-end implementation as an end-to-end application tool. The tool operates on off-line and real-time measurement data obtained from a sensor networking infrastructure (currently being deployed by the SensorNet program at the Oak Ridge National Laboratory (ORNL)). The processing or analysis model component in the "back-end" is the Oak Ridge Evacuation Modeling System (OREMS) (Franzese, 2003) which we consider as a black-box model within the end-to-end tool. In treating it as a black-box model we take for granted its operations and requirements but ensure that the sensor network takes the required steps to produce input suitable to the model. OREMS uses an input transportation network topology and parameters from available GIS databases, parameters of the area at risk, and its own traffic parameters generation model and computes the feasibility of large-scale evacuation. The tool uses LandScan (Bhaduri et al., 2002) to create the demographic model of the population involved. We divide the input data to the tool into two categories, static and dynamic. The static data includes a transportation network, historic traffic flow data, and population distribution. The dynamic data includes more current traffic flow and population estimates. We prepare static data ahead of time, and the network of sensors gathers the dynamic data.

We argue that much of the dynamic data needs to be processed within the sensor network infrastructure before it is made available to applications due to a set of system constraints: communication constraints prevent the transport of large amounts of raw sensor data (image, automobile crossings, etc.) over the network to the application models, computation constraints prevent models (even some that are at the state of the art) from accepting input in copious detail, and cost constraints limit existing monitoring systems to sensors located at critical (bottleneck) points in the network, and consequently, relevant data may need to be extracted from a partial set of readings. We overcome these constraints by pre-processing the sensor data within the sensor network infrastructure at suitable points in the deployment hierarchy using data-aware schemes. We process spatio-temporal metering and imaging data to develop aggregate representations of population and traffic and we interpolate collected values to estimate measures (such as traffic flow) at different points in the network before we provide this data to the application tools. We call the steps of population and traffic aggregation and estimation P_{agg} and T_{agg} respectively and do not go into their internal details in this paper - a wide choice of algorithms are available to carry out those steps - our intention instead is to describe the choice of the location within the infrastructure to carry out those pre-processing steps.

The framework surrounding the application tools must not only translate the data to the appropriate input format for the application tool, but must present and return the results to the users in the field (e.g., first responders). The network infrastructure uses deployed communication mechanisms to reach the end-users. Once the evacuation model has computed its results, the post-processing step distributes them to subscribers of the results in real-time. We show progressive snapshots of the operational procedures that tie the data from network of sensors to the application tool and end-user display. The outline of the rest of the paper is as follows. Section 2 describes the scope of the infrastructure and the operation and requirements of the evacuation model. Section 3 presents the representation model (a process flow graph) of the processing steps and describes our approach to map the processing requirements to the specific dynamic data infrastructure that we are currently deploying. Section 4 offers resulting screen shots of the functional operation of the system. Section 5 includes a short discussion of our work in the context of related efforts.

2. The Sensor Networking Infrastructure and Analysis Models

2.1 SensorNet Infrastructure

The SensorNet architecture includes a distributed network of SensorNet Nodes (Gorman et al., 2005) that communicate directly with sensors. Figure 1 shows how nodes fit into the SensorNet architecture. Nodes connect to sensors and collect data from them. These sensors could also be soft sensors, i.e., the node may simply function as a software actor at the site of a source of stored or collected data obtained from a previously existing infrastructure. The distance between sensors and node varies from a few feet to a few thousand feet. Collected data may be pre-processed at the node before it is passed into the central infrastructure.

SensorNet Nodes within a region (chosen with topographical considerations –generally intended to have a granularity of a few city blocks in urban areas) typically aggregate their sensor data at local data hubs. Local data hubs then submit their information to hubs at one level higher in the hierarchy which we call regional data centers. It is at the regional centers that heavy-weight application models evaluate and analyze the data. As the first collection point of sensor information, Nodes interpret and compress the data into samples guided by a set of predefined operational rules. (For example, during highly varying traffic loads, Nodes report their data more often. During better-understood time intervals, such as night or rush hours, Nodes report their information less frequently thus reducing communication bandwidth needs.) This computation is performed at the Node itself and increases the throughput of the system by offloading computational overhead from the back-end system where the modeling tools reside. Such a deployment hierarchy of Nodes allows physical sensor data and images to travel freely on the first messaging segment, which is typically a communication channel that supports higher bandwidths (e.g., 802.11 wireless Ethernet). On the upstream direction, by performing image processing and feature extraction, the infrastructure uses the lower-bandwidth links to send processed data.

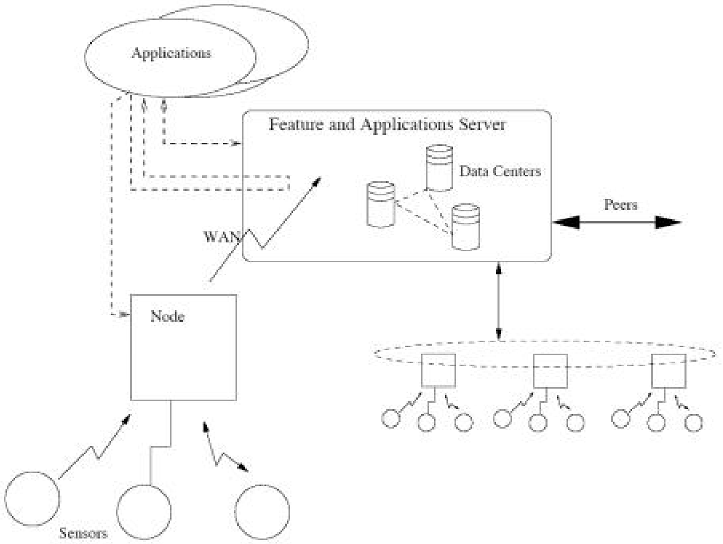


Figure 1 Sensor Network Infrastructure

A deployment also comprises a collection of regional data repositories along with their accompanying services serving regional applications. The data repositories use web-based services to share information with regional peers. Directed by applications, services at the regional data centers instruct service software within the node to collect sensor data periodically. Application services may also directly interact with the node, e.g., if shorter turnaround times are needed (e.g., to actuate transducers).

The input sources of data that we consider include traffic metering and image capture sensors to estimate traffic flows and population distributions. Note that these sensors may not be deployed densely in the area of interest, and consequently, the data gathered by these traffic sensors needs to be interpolated within the infrastructure to infer traffic flows (which include models for population in transit) and pedestrian population distribution. This pre-processing can take place at any one or more of nodes and regional data centers. Aggregate terms of the flows are created within the network nodes, which provide this computed information to the (black-box) analysis models. This computational work performed within the network infrastructure allows the tools to operate purely on the input parameters that they are designed to accept and process, and mitigates the requirement that they be reconstructed or rewritten to handle available data.

2.2 Preparing Data for the Evacuation Models

OREMS, the black-box analysis model, is a microcomputer-based system for simulation of traffic flow during an emergency evacuation. The evacuation can be undertaken in response to a natural or man-made catastrophe. It is designed to allow comprehensive evacuation planning studies including estimates of evacuation times, development of traffic management and control strategies, identification of evacuation routes, and traffic control points and other elements of an evacuation plan. Essential outputs from OREMS include: the length of time associated with evacuation (complete or partial) of the population at risk within the emergency planning zone (EPZ), the number of people that may be at risk based on the rate of evacuation, the best route choices for those evacuating from different areas of the emergency planning region, the best strategy for evacuating people as a protective option by itself or in combination with other protective action strategies, and a list of potential "hot spots" or "trouble spots" (in terms of traffic operations) within the EPZ. Clearly, the effectiveness of these outputs grows with the accuracy of the input data. Thus, a critical goal for our tool is providing realistic and timely analysis of the feasibility of large-scale emergency evacuation for different sizes of the affected area as cost effectively (in terms of time and resources) as possible, and doing this for data that reflects ground truth (i.e., in the form of near-real-time data) as much as possible.

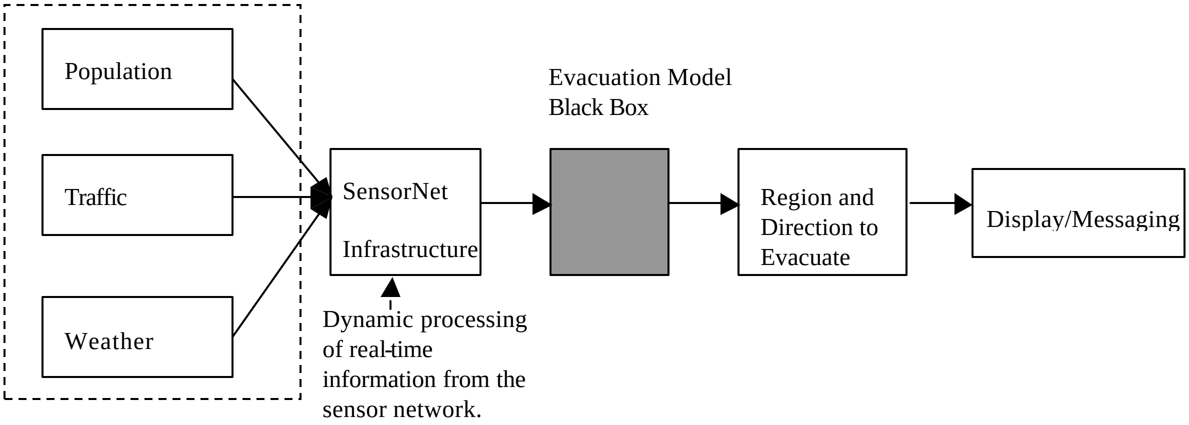


Figure 2. Data-Flow Operation of Evacuation Models

Figure 2 shows the overall data flow to the evacuation model. Our focus is on ensuring that the real-time data available in the network is presented in an efficient manner to the evacuation model. The static descriptions of the transportation network, any historic traffic flow data, and population distribution is provided to the tool directly and co-located with the evacuation model. Static information is also input in the form of topology files and population estimates where there are no deployed sensors. Some of the static information is pre-processed for the models beforehand. Figure 3 shows this pre-processing step done to sample maps in Tiger – a readily available geospatial database that offers basic transportation information (network and population) for evacuation planning. The pre-processing steps here chiefly include road network aggregation and connectivity completion. We perform these pre-processing steps offline to prepare the network as input to OREMS.

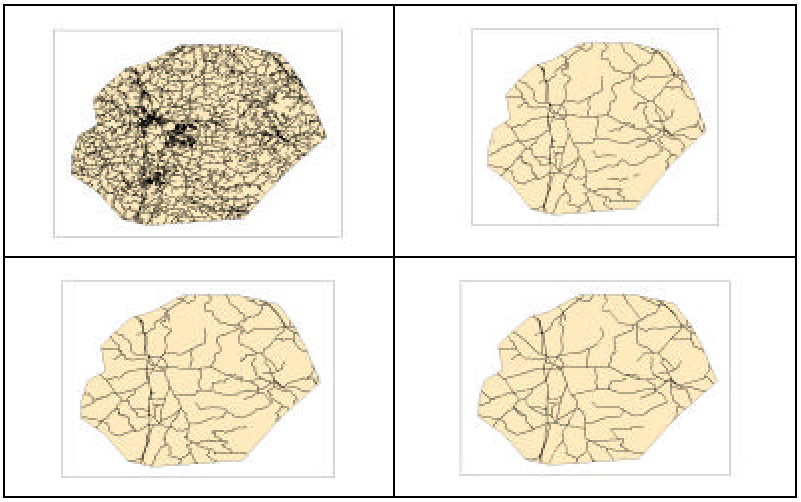


Figure 3. Transportation Network Aggregated Offline and Hierarchically

3. Pre-Processing Dynamic Spatio-Temporal Data

Again, our specific focus is on determining the locations for pre-processing large volumes of fine-grained population and traffic image data for evacuation models. Since the models only handle aggregate measures of traffic flows and population, we define a dynamic spatio-temporal alignment (DSTA) module that takes a spatio-temporal bounding box of data and converts it to an estimate of population and traffic at a particular road or set of roads of interest. The internals of the function module that collects population information and traffic information from image and crossings data are deployment specific and can be modified without significantly changing the approach described in this paper, i.e., we do not mandate any specific design for the internals of these modules and instead abstract their computational requirements.

3.1 Process Graph Model

We first define the process flow that takes place in converting spatio-temporal data into processed information. We label a spatio-temporal bounding box for a region as $BB[xyz_1, xyz_2, (t1, t2)]$. We drop the parameters when the meaning is clear and for simplicity refer to a bounding box for a region i as BB_i. We assume that library operations that perform unions (we limit our focus here to aggregations and hence unions) of bounding boxes exist, and the meaning of $BB_k = BB_i \, U \, BB_j$ is not ambiguous. The metering and image input data collected from a region i is $I(BB_i)$. The relevant population information from the region is $P(BB_i)$ and the traffic information for a road R is $T_R(BB_i)$.

The Dynamic Spatio-Temporal Alignment module takes image and traffic sensor data and converts it to a form usable by the evacuation model. We represent this as follows: $DSTA(I(BB_i))$? $\{P(BB_i), T(BB_i)\}$, where $T(BB_i)$ without the subscript R represents traffic information for all relevant roads within the bounding box. As mentioned above, a variety of techniques can be a part of the estimation procedure $DSTA()$ – we do not preclude the inclusion of any particular technique here as long as the technique can provide the estimate based on the inputs from $I(BB_i)$. Also, extensions that take into account adjacent BB_j are also possible for computing $P()$ and $T()$– in the systems we consider, we focus on the case that communication and messaging bandwidth for these extensions is small compared to the data sent to superset bounding box computations.

The output of the DSTA operation needs to be aggregated from various regions following the population extraction and traffic estimation steps from a bounding box. We divide this aggregation step into two functions P_{agg} and T_{agg} which have the following functional behavior: $P_{agg}\{P(BB_i), P(BB_j)\}$? $\{P(BB_k)\}$ and $T_{agg}\{ T(BB_i), T(BB_j) \}$? $\{T(BB_k)\}$, where $BB_k = BB_i \, U \, BB_j$.

The above steps induce a class of networks in which the application tool's data-flow is represented by a process flow graph. Figure 4 represents the data-flow steps we consider in this paper that take place as the real-time dynamic data arrives at evacuation analysis models.

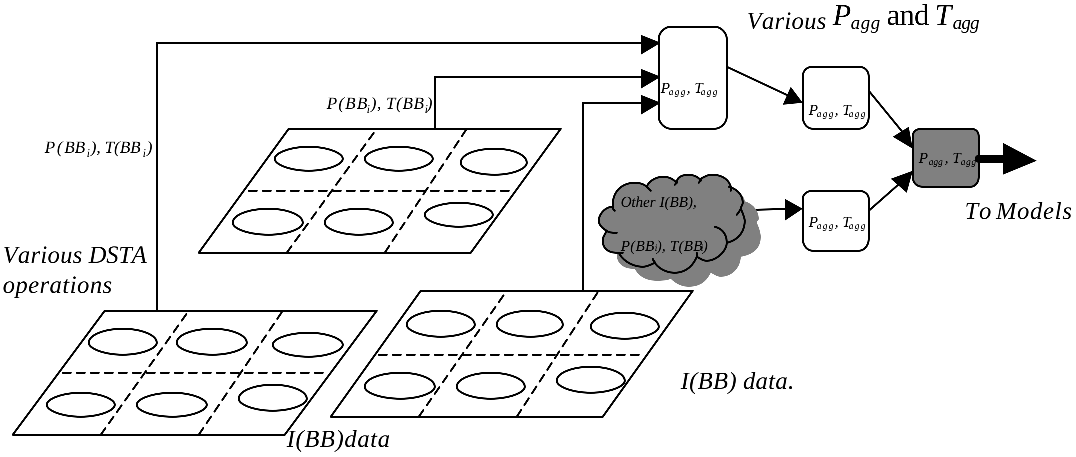


Figure 4. Process Flow Graph

Available communication and computation capabilities at different points in the network constrain the choice of what data is transported in raw form and what is sent in processed form over the network links. We can now address the problem of determining where pre-processing of the sensor data takes place within the sensor network infrastructure deployment hierarchy.

3.2 Deployment Graph

The dynamic data-flow is represented by the process flow graph, which is created off-line. The processing locations for components of this graph are determined by a cost minimization analysis for the deployment scenario.

We represent our deployment as a set of sensors and nodes arranged hierarchically to bring the data back to analysis modules as shown in Figure 5. Each link on the hierarchy has an associated cost with transferring a bit of data. The cost can be along the time dimension (which is the case we consider here) or along the price dimension (when bandwidth is paid for as a commodity). We wish to minimize the cost of the overall processing in the sensor network when components of the process flow graph are situated on the deployment graph, while obeying precedence (data and functional) and computation constraints.

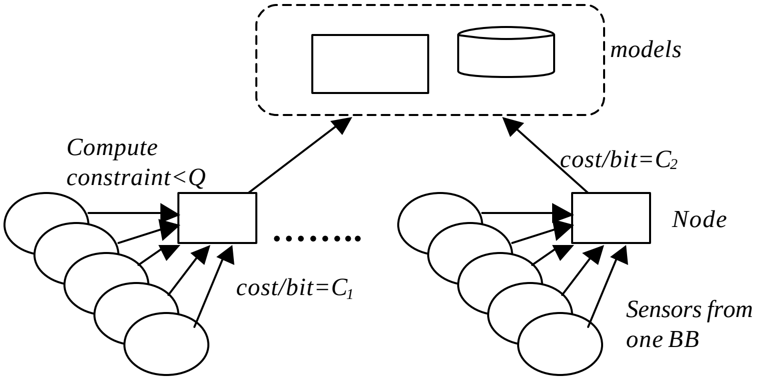


Figure 5. Deployment Graph

In general, the problem of mapping a process graph to a deployment graph where both are represented by directed acyclic graphs and the goal is to minimize the total cost incurred is computationally expensive. However, we can justify the mapping process to the particular and initial instantiation of our simplified deployment hierarchy in a straightforward manner. (In future work we are addressing this problem in the case of a fully general deployment hierarchy.) The bi-layered structure allows us to make a simple argument for the deployment. We consider the case of our two-stage hierarchy. The first stage of the hierarchy has a cost/bit = C_1 (process cost at node) and the second stage of the hierarchy has a cost/bit = C_2.(transmission cost from node to center). Each node in the sensor network has a compute constraint less than Q. For simplicity we assume this is the same for all nodes in the network.

3.3 Pre-processing Placement

Our network hierarchy obeys the following rules:

- $C_1 << C_2$.
- $DSTA < Q$ for *BB's* and sensors for a node's region.
- *I(BB)* is much greater in size than *P(BB)* and *T(BB)*

The P_{agg} and T_{agg} terms must all be computed and for that to happen *I()* must be collected over all BB. This creates a minimum mandatory cost. *DSTA(I(BB))* gives *P(BB)* and *T(BB)*. Since the size of *P(BB)* and *T(BB)* is much smaller than *I(BB)*, we become concerned only with the costs of transmitting image and counts data collected over a bounding box.

To minimize cost, we must minimize:

$$\text{Total Cost} := I(\text{over all BB}) * C_1 + I\,(\text{over all links transmitted to center}) * C_2 + DSTA\,(\text{over all BB}) + \{P_{agg} + T_{agg}\}\,(\text{over all DSTA's and BB}) \qquad (1)$$

The DSTA and aggregation steps cost the same if done at the center or at the nodes. This is not strictly true in general because the computational capability of the nodes may limit the extent to which this can be done. (In the interests of clarity we do not expand the discussion here to the more complex optimisation with Q not equal for all nodes and *DSTA* not less than Q for all nodes.) We observe that since $C_1 << C_2$, the second term in Equation 1 dominates due to the given capabilities of the sensor network infrastructure - thus we must not allow $I()$ to be transmitted over the communication links upstream from the Nodes. This leads us to a placement of DSTA on all the nodes. We place $\{P_{agg} + T_{agg}\}$(over all DSTA's and BB) at the centre. Again, we note that for deeper hierarchies that have multiple levels of nodes the placement of the aggregation steps needs to be optimized further and the simple argument used here will not suffice.

3.4 Display and Messaging

A post-processing phase uses a dynamically created table connecting data results to consumers in the hierarchically divided region. A post-processing module evaluates this relation at run-time and creates the messages for subscribing users (for example, in some cases the message may be sent as a direct page, in other cases the message is a synchronous notification to another application model). The infrastructure and framework surrounding the application tool uses a rule-driven approach to provide targeted messaging to consumers. In certain instances, the message reaches the relevant Node within the region, and first responders may retrieve directives and current system state from the nearest accessible Node. The most common method is for the consumers of the output at the control center to view a visually mapped display. First responders and consumers in the field receive messages based on their location (collected and accounted for separately over the SensorNet infrastructure) which is maintained in the dynamically created table mentioned above.

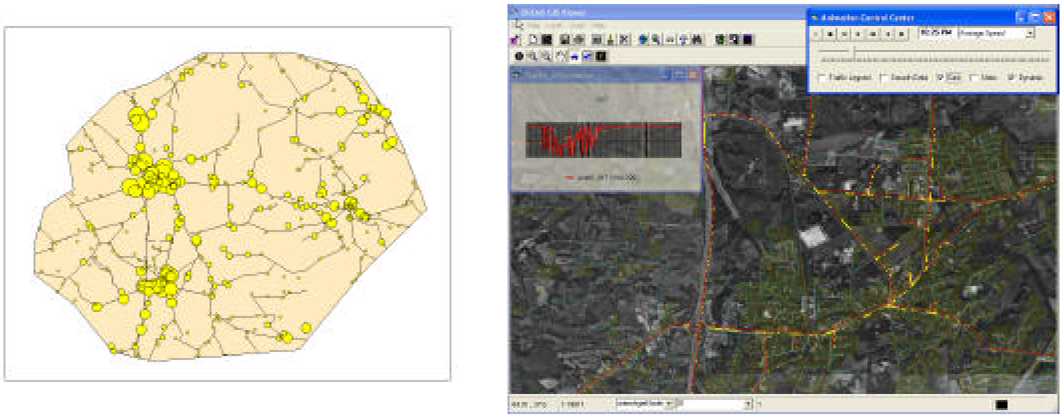
4. Results of Operation: Process Snapshots

This section gives operational screen captures of the system results at different stages of the pre-processing and output process. The population aggregation screen shots include data scraped from existing databases due to the absence of a sufficiently dense current sensor deployment.

4.1 Pre-Processing Stages

Figure 6. Population Information Aggregated Hierarchically in Stages

Figure 6 shows the stages of population data accumulated from bounding boxes as they aggregate over the region of interest. A snapshot of the input immediately before it is submitted to the OREMS model is in Figure 7 (a).



(a) (b)

Figure 7. Final Population and Transportation Network for OREMS and Output Screens presented to the users at the control center.

When input data is processed by OREMS, we process the output to take the form shown in the Figure 7 (b) which is the display for end-users to visualize the movement of the people and the temporal behaviour of the evacuation process in the region.

5 Related Efforts and Discussion

The Department of Transportation's Federal Highway Administration (http://www.fhwa.dot.gov) has taken strides in establishing programs to enable Dynamic Traffic Assignment. The Intelligent Traffic Management Systems (http://www.its.dot.gov) aim to take various input parameters (real-time and static) into account to create better traffic routing and congestion control plans. However, incorporating evacuation programs with population data and remote sensing has received increased attention only recently (Franzese, O. et al 2001; Kafotos, M. et al 2002) and is an acknowledged challenge (see, e.g., presentations from reference for NATMEC 2004). Much of the past work has not addressed the challenge of structuring the processing for the models in a way that real-time infrastructures can share the workload. This is probably due to the lack of a live and accessible sensing infrastructure in the field. Consequently existing work has forced models and tools to consider the analysis of static data. Our work leverages an ongoing deployment of a sensor network infrastructure to collect live and dynamic data and consequently leads us to devise systematic procedures to process spatio-temporal data eagerly within the network to make it amenable to existing models.

The convergence of an available real-time infrastructure and an integrated population and traffic-aware evacuation modeling application has provided us with important systems research questions - particularly with regard to describing and laying out the application process-flow on an available deployment. We have aimed to take a systematic approach to migrating static application systems to dynamic real-time sensor networking deployments. Applications models in the domain of geocomputation are often overwhelmed with more data than they are designed for and a systematic procedure to bring in and prepare input data from the field is an urgent need. We have taken initial steps in proposing a model for representing the computations required in the spatio-temporal domain as a process graph. By judiciously overlaying the process flow on the deployment graph, cost outlays can be minimized. Anticipating progress in our sensor networking deployments, we are currently modeling our application and deployment components with greater accuracy and considering broader constraints as we map process-flows to the deployment graph.

6. Acknowledgements

We acknowledge the ORNL SensorNet project and Office of Naval Research for their support. Discussions and input from Oscar Franzese on the functioning of OREMS is sincerely appreciated. This work has been done for the Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy under Contract No. DE-AC05-00OR22725.

7. References

Oak Ridge National Laboratory, SensorNet Program. http://www.sensornet.gov

Budhendra Bhaduri, Edward Bright, Phillip Coleman, and Jerome Dobson, April 2002. LandScan: Locating People is what Matters. Geoinformatics Vol. 5, No. 2, pp. 34-37

Gorman, B.L., Shankar, M., and Smith C.M., The SensorNet Node: Last-Mile Platform for Interoperability, Sensors Magazine, Vol. 22, April 2005.

Lieu, H., "A Roadmap for the Research, Development, and Deployment of Traffic Estimation and Prediction Systems for Real-Time and Offline Applications (TREPS, TREPS-P)," FHWA Technical Report, July, 2003.

Franzese, O. et al.(2003), OREMS 2.6 User's Guide, Oak Ridge National Laboratory.

Franzese, O. and L. Han, 2001, "A Methodology for the Assessment of Traffic Management Strategies for Large-scale Emergency Evacuations, 11th Annual Meeting of ITS America, 2001

Kafatos, M. et al, Utilizing Remote Sensed Data InA Quick Response System, ISPRS Special Session, FIEOS/ISPRS, 2002

Chakraborty, J., Tobin, G.A., and Montz, B. E., Population Evacuation: Assessing Spatial Variability in Geophysical Risk and Social Vulnerability to Natural Hazards, Natural Hazards Review, Vol. 6, No. 1, February 2005, pp. 23-33

NATMEC - North America Travel Monitoring Conference 2004, http://trb.org/Conferences/NATMEC/2004/. See for example: Robert Collins, Post, Buckley, Schuh & Jernigan, Inc. (PBS&J), The Hurricane Evacuation Analysis and Decision Support Utility Program (HEADS UP): Florida. And The Role of Complex Data in a One-Pager World by Timothy Lomax, Texas Transportation Institute