Dataflow-Oriented Provenance System for Multifusion Wireless Sensor Networks

Gulustan Dogan[1,*] and Ted Brown[1]
[1]City University of New York, Graduate Center, 365 5th Ave, New York, NY 10016
[*] E-mail of the corresponding author: dogangulus@gmail.com

Abstract
We present a dataflow-oriented provenance system for data fusion sensor networks. This model works best with net- works sensing dynamic objects and although our system is generic, we model it on a proximity binary sensor network. We introduce a network-level fault-tolerance mechanism by using the cognitive strength of provenance models. Our provenance model reduce the limitations of a sensor's capability and decrease the error-prone nature of wireless sen- sor networks. In addition provenance data is used in order to efficiently build the dynamic data fusion scenario and to adjust the network such as turning of some sensors. In a fault-tolerant, self-adjusting sensor network, sensor data produce more accurate results and with the improvements, tasks such as target localization is more precisely done. One other aspect of our network is that by having computation nodes spread to the network, the computation is done in a distributed manner and as nodes make decisions based on the provenance and fusion data available, the network has a distributed intelligence.

Keywords: Multifusion, Wireless Sensor Networks, Open Provenance Model

1 Introduction

This paper is targeted for building a dynamically configured dataflow-oriented provenance system for real-time and continuous monitoring data fusion sensor networks. Sensors collaboratively carry out the sensing task and forward sensed data to the closest data processing center. However, it is not possible to record the data flowing snapshot of the network without provenance. Provenance makes it possible to have a clear picture of the dataflow by tracking the evolution of the data systematically. Besides, failure recovery involves understanding causal chains of events and dataflow model is a solid reference of the phases data goes through (Cheney, et al., 2009b). In this paper, provenance will be used in order to find out causes of faulty behavior, to figure out the circumstances that will determine the connectivity of the network, to produce trustworthy data after elimination of the causes. The possible errors and how provenance will be used to eliminate them is described in Section 4.
To illustrate our concepts we examine a field of proximity binary sensors. In proximity-based wireless sensor networks,

the likelihood of the target position is calculated using the binary values reported by proximity binary sensors. The sensors should be able to tell that there are k intruders and depending on the density give a reasonable location of each of them. A proximity sensor acts as a tripwire i.e. it reports a detection when a target close by triggers it. Examples of these sensors are seismic, acoustic, passive infrared and they can be deployed in large numbers because of their low cost. The binary proximity behavior in sensors is achieved by implementing simple energy detection algorithms where the signal is compared to a threshold. If the signal exceeds the threshold, the sensor node reports a "1" meaning a detection, otherwise a "0" is reported for no detection. In some target localization networks, in case of no detection sensors do not report. A network of such sensors can be used to localize and track targets(Qiang Le, 2010). Several papers have been written about locating a moving intruder as stated in Related Part section. However locating k intruders is still anopen problem although there is ongoing research. Our architecture and fusing algorithms can differentiate the k and locate them. Besides provenance data is captured in our architecture as a support for the reliability of the algorithms. The physical network contains sensors, fewer computation nodes and a central node but these are interspersed within the sensor field.
This paper is organized as follows. In Section 2, we describe the related work on provenance models of sensor networks. In section 3, we give some background. In Section 4, describe our model. In Section 5, we list the challenges we faced about this research. Section 6 concludes the paper.

2 Related Work

There is research done on target localization in Sensor Networks community. Hall and Llinas have a work on multi sensor data fusion applied to Department of Defense (DoD) areas such as target recognition, battlefield surveillance (David Hall, 1997). Previous work was done on single target localization using proximity binary sensors by several re- searchers (A Artes-Rodriguez, 2004; Aslam, et al., 2003; Shrivastava, et al., 2006; Agostino Capponi, 2006). Besides, there has been work done on multiple target localization where number of targets is known(Qiang Le, 2010; Singh, et al., 2007). However provenance information is not used in these research, the computations are based on solely bi- nary localization data reported by sensors. To our knowledge this is the first work leveraging provenance data in target localization sensor networks.

Provenance has been studied in Sensor Network community. Provenance aware sensor data storage systems are pro- posed. In these systems, sensors collect provenance information of the data they are sensoring or the processes they are running (Ledlie, et al., 2005). Furthermore, provenance information associated with sensor data has been used in answering domain specific complex queries (Patni, et al., 2010). Park and Heidemann explore the need for data prove- nance in a sensor network to understand how processed results are derived and to correct anomalies (Unkyu Park, 2008). In addition, provenance-aware Open Provenance Model based sensor systems have been implemented in different do- mains (Liu, et al., 2010; Stephan, et al., 2010). There has been work presenting frameworks for provenance-aware sensor networks where data fusion methods are implemented (Liu, et al., 2009). However, to our knowledge this is the first work where fused nodes are grouped and selected based on an adaptive algorithm.

There has also been work done on provenance management in eScience community. Scientific workflow systems include myGrid/Taverna (Oinn, et al., 2004), Kepler (Bowers & Ludascher, 2005), VisTrails (Freire, et al., 2006), and Chimera (Foster, et al., 2002) etc. They automatically capture provenance during workflow creation and execution to support reproducibility of scientific experiments (Susan B. Davidson, 2008). eScience research has included some work on sensor data access, analysis (Barseghian, et al., 2010) and provenance-based fault-tolerance mechanisms (Crawl & Altintas, 2008; Thomas Huining Feng, 2008).

The database community has also addressed the issue of provenance. Two types of provenance, data and workflow provenance, are described as defined Section 3 (Moreau, et al., 2008b) and data provenance research has made a dis- tinction between where, why and how provenance (Buneman & Tan, 2007). There is also extensive reseach in database community on query models and provenance storage, collection (Moreau et al., 2008b). There is also previous research on *dependency provenance* addressing the need for storing dependencies (James Cheney, 2007).

3 Background

3.1 Provenance

Provenance is defined broadly as the origin, history, chain of custody, derivation or process of an object. In other disciplines such as art, archeology, provenance is crucial to value an artifact as being authentic and original. In com- putational world, as all kinds of information can easily be changed, provenance becomes an important way of keeping track of alterations (Cheney, et al., 2009a). Although data fusion systems will contribute to many research fields by their feasible characteristics, provenance management should be also a concern in order to have an understanding of how results are obtained for later use such as fault tolerance, troubleshooting, result reproduction and performance optimization.

The literature generally divides provenance into data and workflow provenance (Moreau et al., 2008b). Data prove- nance gives a detailed record of the derivation of a piece of data that is the result of a transformation step (Tan, 2007) whereas workflow provenance is the information or metadata that characterizes the processing of information from input to output (Susan B. Davidson, 2008). Both data provenance and workflow provenance will be the concern of this paper and we will elaborate more on them in the other sections.

3.2 Open Provenance Model

Interest for provenance in eScience community is growing since provenance is crucial for scientific workflow systems to support reproducibility, reusability and maintainability. Against this background *First International Provenance and Annotation Workshop* (IPAW'06) held in 2006 in Chicago. Researchers who joined the conference wanted to know about the other systems' capabilities and expressiveness such as design principles, representations, retrieval methods, storage choices. Hence, they agreed on organizing an event for sharing the provenance approaches of different systems and the idea of *First Provenance Challenge* was born. At the end of the Second Provenance Challenge, researchers decided that there should be a standardization of the way provenance modeled, stored, queried and changed to make the systems fully compatible with eachother. Following this consensus, authors met in a workshop in 2007, crafted and iterated a data model named as Open Provenance Model. In the following paragraphs we will briefly write about the Open Provenance Data Model's specifications, properties and design principles.

Open Provenance Model is based on three primary entities defined below.

Definition 1 (Artifact) Immutable piece of state which may have a physical embodiment in a physical object , or a digital representation in a computer system. They are represented by circles.

Definition 2 (Process) Action or series of actions performed on or caused by artifacts and resulting in new artifacts. They are represented as rectangles.

Definition 3 (Agent) Contextual entity acting as a catalyst of a process, enabling, facilitating, controlling, affecting its

execution. They are represented as octagons.

Provenance of objects is modeled as a directed acyclic graph (DAG) representing a past execution, never a prediction of future events. Dependencies are shown with edges, an edge represent a causal dependency between its source, denoting the effect, and its destination, denoting the cause. There are five predefined causal relationships *used, wasGeneratedBy, wasTriggeredBy, wasDerivedFrom, wasControlledBy* and edges are labeled with one of these causal relationships.

4 Overview

4.1 Sensor Network Model

There are two sensor network models used in the literature. In the first model, all sensors get a signal from the target and the energy is added up and compared to a threshold (Qiang Le, 2010). If the voltage is bigger than the threshold then the sensor reports a "1". Whereas in the other model sensors report a "1" if there is a target in their disc area (Singh et al., 2007). We use the second model in our network. We believe that fusion idea works better with the second model; because data fusion nodes will be fusing the data coming from nearby nodes, it is more efficient if sensors go red(detecting, hot) if the intruder is in the disc area. In the first model, there may be faulty reports if many small amounts of energy add up and exceed the threshold.

In our model rather than getting all data sent to a central station, we have many computation nodes doing fusion and necessary computations that are specific to each network. It is a triggered system, at specific time intervals data is collected and data fusion and computations are done in a distributed manner. In our example network, the computations will be related to target localization. We have the distributed approach.

4.2 Dataflow Provenance Model

The ideal provenance model should contain sufficient information to be able to recreate an exact replica of any object (Muniswamy-Reddy, 2010). In this paper, we are aiming for this kind of model. If the provenance of data fusion process is available, cognitive decisions can be made by the sensors using the provenance information. For example in a case

that a data fusing node is waiting for information coming from another node but that node fails to send the information and the waiting sensor times out, the sensor will have the information of other possible sensors that might have the same kind of information it is waiting for. This kind of behavior is supported by dataflow provenance model. Hence in our system we will use a dataflow model based on Open Provenance Model. This model tracks how information flows from one object to another (Andrei Sabelfeld, 2003). With this model we have a snapshot of the network at specific time intervals which we can refer to do some conclusions such as regrouping the fused sensor nodes, omitting a sensor node, changing the dataflow scenario. Dataflow model is the right model because data and control dependencies and the dataflow path is efficiently captured in this model as described in the paragraphs below.

Open Provenance Model has become a standard in modeling provenance as described in Section 3. Dataflow Prove- nance Model is described in the literature based on Open Provenance Model (Muniswamy-Reddy, 2010). Nodes rep- resent objects, edges represent information flow between the source object and the destination object as illustrated in Figure 1. Source object is treated as the ancestor of the destination object (Muniswamy-Reddy, 2010). Open Prove- nance Model of a sensor network is illustrated in Figure 4, Figure 5, Figure 6, Figure 7 and Figure 8.

Provenance answers questions such as "how was the object created", "on what other objects does this object depend",

"how do the ancestries of these two objects differ" (Muniswamy-Reddy, 2010). In our data fusion system, fusion sen- sors will have the provenance information of "on what other sensors does the fused data depend" and this provenance information will be used in analyzing data fusion flow graphs of sensor networks for the responsible nodes in case of faulty behavior. This kind of provenance information is referred as data dependency (Crawl & Altintas, 2008). Two services connected by a dependency characterizes the precedence order between them. Data dependencies are catego- rized as value-dependency and control dependency (Crawl & Altintas, 2008). A value dependency occurs when output data's value depends on the value of previously read data. A control dependency occurs when arrival of the data causes the actor to execute but it is not used. In our model we have value dependencies stored as provenance data. An example of a value dependency is *"Fusing node C depends on data coming from node A*" as shown in Figure 2. In this case, operation of fusing node C depends availability of the data provided by A. Another example illustrated in Figure 2 is *"Fusing node C depends on data coming from node A and node B"*.

For example in our system data dependencies are provenance information and they are defined as followed "Computa- tion Node A depends on Sensor Node 1, Node 2 and Node3", "Computation Node B depends on Sensor Node 3, Node 4 and Node 5"

4.2.1 What-if Analysis

We also have control dependencies in our model. Control flow refers to branching, iteration and jumping via if-then- else, switch-case and while loops. Adding control flow to dataflow model strengthens the power of network in terms of error-recovery, robustness and fault-tolerance (Bowers, et al., 2006). Control flow structure will help sensor network in making cognitive decisions in case of a failure. An example scenario where control flow will be useful is as follows. When a computation node is not receiving correct and sufficient input from the nodes it is data dependent, it should take an action to find a correct result. One possible action can be asking to another fusing node. This behavior can be implemented by a control flow statement such as "If the incoming edges do not send reliable data then ask to fusing node X".

There is also what-if analysis available in our system using the historical data stored in the Central Node. Mining the data, it can be found out how the network will respond if the target moves along a specific path.

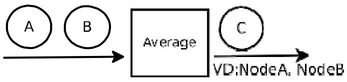

Figure 1: node C depends on value coming from node B

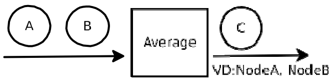

Figure 2: node C depends on value coming both from node B and node A

4.3 Provenance Gathering

Our system will support provenance from its initial design. This method is called observed provenance (Happe, 2010). Provenance data is automatically gathered without depending on user input. In this type of provenance collection, an administrator can specify some parameters to the monitoring process but cannot directly interfere with the collection phase. Advantage of automated monitoring is that it is resistant to malicious attacks since users cannot alter the prove- nance data. As provenance is collected automatically, integrating provenance data with the application will be done during the software development phase.

Wang-Chiew Tan focuses on data provenance and classifies current data based provenance gathering techniques into

two : annotation-based and non-annotation based (Tan, 2007). We will be using the annotation based approach for provenance gathering. In the annotation based approach, each datum will have annotations attached to it. When a transformation occurs, the datum is annotated with the provenance information.

4.4 Provenance Storage

In most sensor network systems, there is a central or distributed provenance storage system (Ledlie et al., 2005). Prove- nance of sensor data is stored for later reuse and reference. We will transmit the provenance data to a central storage. We will keep the amount of provenance data at the required minimum level not to consume much energy for transmis- sion.

In our centralized system, there will be a connection between the data and metadata, they are stored in different systems with different representations. Maintenance is more difficult in a centralized approach but it is easier to query and search provenance since the mechanism is designed keeping this requirement in mind (Simmhan, et al., 2005).

All the data and provenance data flowing over the network related to the localization will be kept in the Central Stor- age. Everytime a new target steps in the field, data and provenance records will be stored in the Central Storage. The historical data will also be available in the Cental Storage for network maintenance such as figuring out the sensors that are silent for very long time, determining the group of sensors that are misreporting. Final decision regarding the location estimation will be done at the Headquarter.

Provenance collecting and processing is very costly. However richer provenance is better for failure recovery. Therefore how much provenance to collect is an important choice.

In our system data is tightly coupled to provenance, data and metadata are stored in the same storage system with the

same keys. Provenance data of the images will be attached to the same picture file as done in the headers of NASA Flexible Image Transport System. We will recursively traverse the ancestry path in order to obtain the full metadata history.

4.5 Data Fusion

Multi sensor data fusion has many advantages over single source data such as increased accuracy (David Hall, 1997). In our model, data from several sensors is combined at computation nodes where data fusion takes place. In a target localization network, there are many sensors detecting one target, in this case it will be less definite where the targets are. In our model, binary localization data coming from sensors will be fused in fusing nodes and more trusted results will be produced. collaboration between sensors

Our paper takes advantage of the research that has been done on "header nodes". There is a lot of research papers on this topic. the general idea in these papers is to use one node in the network as the one that communicates to a hop node (another header node) for a group of nodes. That is, one node in a group (chosen by geometric reasons to be close to the others) as the one that collects and transmits information for the group.this is called the header node. the research

idea in these papers is to reduce the overall energy consumption of nodes in the network since most nodes only have to communicate a short distance). We will use the header nodes as our nodes that do the computation.

One significant innovation our paper has related to data fusion is that the fusing scenario is dynamic. The group of

sensors that are being fused are changing as the network is self-adjusting using the historical provenance data. On the other hand data fusion helps in getting more accurate results compared to a model where all data is collected at a central node without any intelligence.

Data fusing sensors will be invoked when input data is provided by reporting sensors otherwise they will stay inactive (to consume less energy). We believe that data is useful around the place it is created. If sensors are reporting “1”s around a place then the likelihood is that the target is near by and it is not reasonable to forward the sensor data over many hops to the regions of undetecting sensors. Therefore it is not efficient to transmit sensor data and the provenance of the data over long distances. In our model, the data will be transmitted until it reaches to the nearest computation node. Our algorithm will be running on computation nodes and localization of the target will be done.

We are interested in a scenario where a smart fusing node (CN) collects information from multiple nodes. We call a fusing node computation node (CN) in our model. Our fusing scenario is dynamic based on an adaptive method, the group and type of nodes that are fused is not static. Connection between sensors and CNs are not permanent, the connections are formed temporarily. For example if a target is moving along a path, in the first time slot CN 1 can fuse data coming from sensor A, B and C, in the second time slot it can fuse the data coming from sensors C, D, E and a camera sensor. In our model sensors with only positive detection transmit their reading, they report a “1” being in awake cycle or they stay in sleep cycle. To get more accurate results we fuse the data coming from many nearby nodes that are transmitting “1” grouping them based on the target movement path. The computation node can be in three different states (0) waiting (1) collecting (2) asking to another node for more information when the arriving data is insufficient or insensible. In order to prevent noise and energy consumption, sensors do not talk to eachother, they simply report to a CN. However computation nodes talk to eachother in case of a failure or a transmission. The fused data is collected at a central node and computations are done at this central node to find the exact coordinates of the target.

There are many different fusion approaches. In networks, information travel over multiple nodes before reaching to a destination, fusion nodes should be aware of the dependency information to be fused and avoid duplication, this is achieved by provenance in our system. For example in the target localization network we work on, there can be multiple targets in the field, two computation nodes can report the existence of the same target. In this kind of a situation, provenance data will be useful to detect the duplicates (Bal, et al., 2010).

4.6 Fault-tolerance based on Cognitive Decisions

Consider the following scenario: A target steps in to the field and a target localization decision is done by the network. In the headquarter, users see that the target is localized wrongly. However, the other localizations done by the network were totally correct. Provenance will be helpful at this point in order to understand the reasons behind. In our model, provenance graphs of localizations (data flow from one node to another) can be examined and causes for any change or fault can be detected. After finding the error, by using the provenance information the localization can be redone. Our dataflow-oriented sensor network model will capture and tolerate some failure patterns. As our flow model has control and value dependencies stored, fault-tolerance will be doable. Using the snapshots built through provenance, the network gains a cognitive strength and can make intelligent decisions such as omitting a sensor node, debugging a fusion node. For example when a sensor is failing, it can be replaced and retransmission can be done. Some possible exceptions in a sensor network can be (1) sensor node failure : inability of the sensor node to transmit any further (2) deadline expiry : sensors can have transmission deadlines such as in the network the sensors can be implemented such that they will transmit every 20 minutes. But in our target localization sensor network, it will be more meaningful to have sensors which are triggered by a signal of a living being trespassing. (3) resource unavailability : a computation node needs access to data coming from several sensor nodes. if these data is unavailable then the data fusion cannot take place. In our system these dependencies will be defined as data dependency. (4) external trigger : triggers will are used to signal the occurrence of an event such as a detected target in the sensors coverage area. A sensor node can stop being triggered by events, that will rise an exception. (5) constraint violation : according to their manufacturing details (lifetime), due to energy constraints (weak battery), due to environmental conditions (excessive wind, rain).

One of the most frequent problems in wireless sensor networks are path loss and unpredictable multipath (Bal et al., 2010). In our model, we have access to the dataflow diagram that provenance makes available hence any path loss and multipath can be detected.

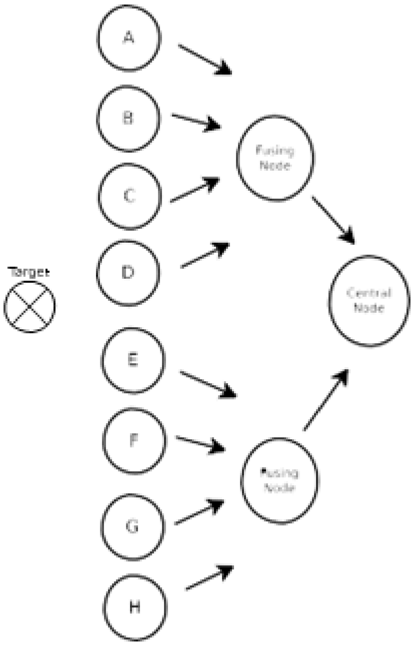

Figure 3: Sensor nodes are forwarding to computation nodes.

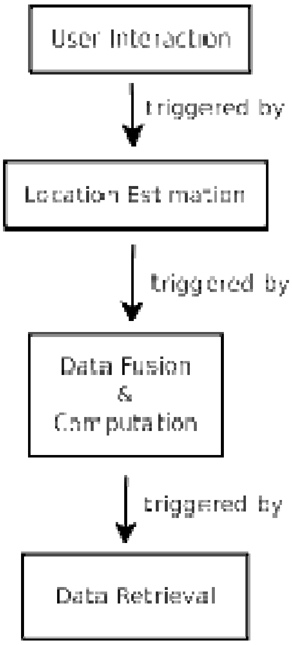

Figure 4: OPM Graph with no details

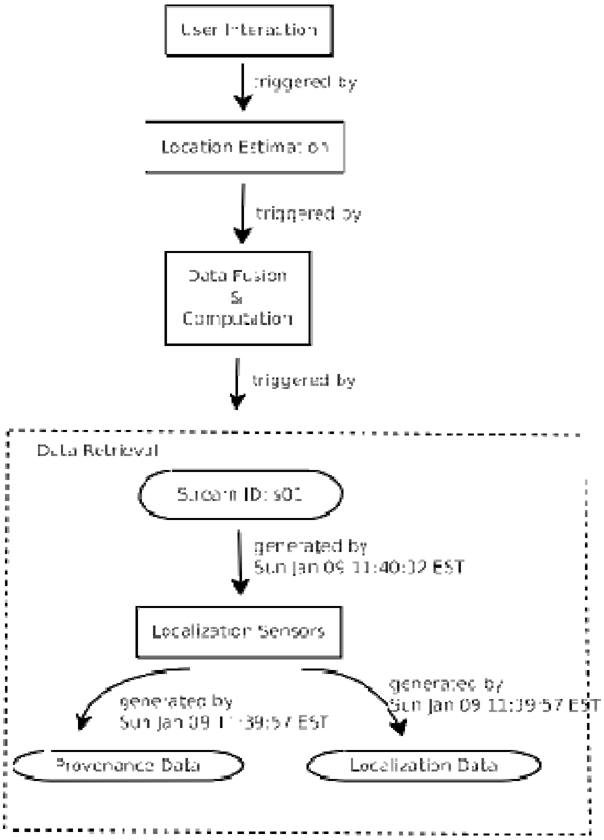

Figure 5: OPM Graph with details on Data Retrieval from binary localization sensors

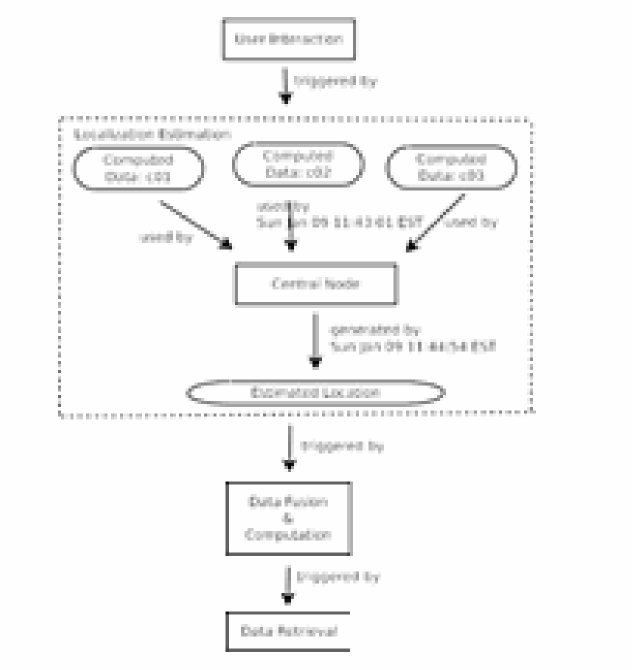

Figure 6: OPM Graph with details on Data Fusion, Computation on computation nodes

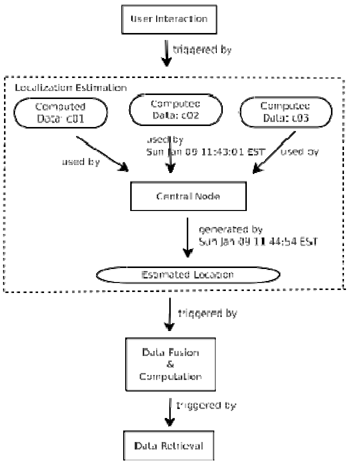

Figure 7: OPM Graph with details on Location Estimation on central node

5 Challenges

There are challenging research questions concerning this study. One question is that data coming from how many sensors should be fused. For instance fusing data of every two sensors can be too costly, fusing data of every 100 sensors can be too inaccurate. A reasonable number should be found according to parameters such as the size of the field, number of sensors. Computation nodes add to complexity of the network and transmissions to them cost energy. But on the other hand we should also take into account the reduction in the cost because of the decrease in amount of the data transmitted. For example at a fusing node, there will be computations done using the reported values of the target detections coming from multiple sensors. The result of the computation which is a single value will be transmitted over the network. This will decrease the data overload in the network. On the other hand data fusing nodes add cost in terms of time and energy to the network as they are extra nodes and computations require energy.

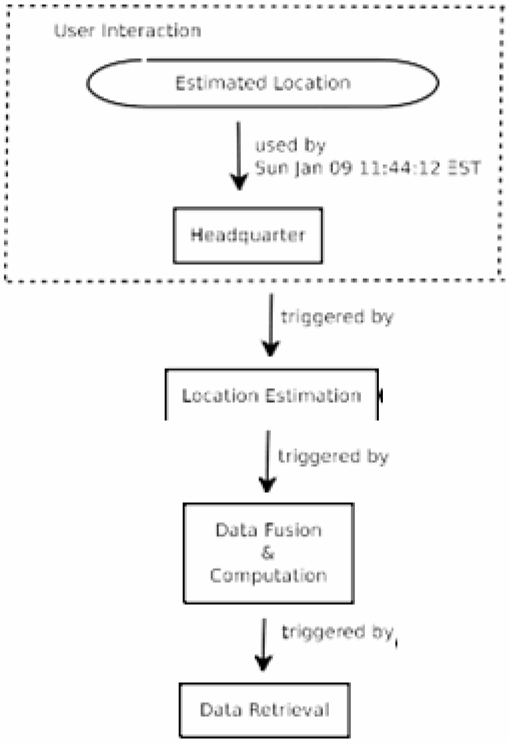

Figure 8: OPM Graph with details on User Interaction in headquarter

6 Conclusion

Dealing with provenance in systems where data moves along such as sensor networks is an open research area because it is hard to manage provenance when objects are mobile or distributed. Various solutions have been proposed to this problem but often solutions are domain-specific. A true solution will require architectural changes at the main levels such as hardware, network, operating system (Cheney et al., 2009b).

References

L. T. A Artes-Rodriguez, M Lazaro (2004). 'Target location estimation in sensor networks usign range information'.*IEEE Sensor Array and Multichannel Signal Processing Workshop* 85(1):6–23.

L. K. Agostino Capponi, Concetta Pilotto (2006). 'Performance characterization of random proximity sensor networks'.In *Asilomar Conference on Signals, Systems and Computers*.

A. C. M. Andrei Sabelfeld (2003). 'Language-based information-flow security'. *IEEE Journal on Selected Areas in Communications* 21.

J. Aslam, et al. (2003). 'Tracking a moving object with a binary sensor network'. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 150–161. ACM.

M. Bal, et al. (2010). 'Collaborative signal and information processing in wireless sensor networks: a review'. In *2009 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3240–3245.

D. Barseghian, et al. (2010). 'Workflows and extensions to the Kepler scientific workflow system to support environ- mental sensor data access and analysis'. *Ecological Informatics* 5(1):42–50.

S. Bowers & B. Ludascher (2005). 'Actor-oriented design of scientific workflows'. *Conceptual Modeling* pp. 369–384.

S. Bowers, et al. (2006). 'Enabling scientificworkflow reuse through structured composition of dataflow and control- flow'. In *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*, pp. 70–70. IEEE.

P. Buneman & W. Tan (2007). 'Provenance in databases'. In *Proceedings of the ACM SIGMOD international conference on management of data*, pp. 1171–1173. ACM.

J. Cheney, et al. (2009a). 'Provenance in databases: Why, how, and where'. *Foundations and Trends in*

Databases **1**(4):379–474.

J. Cheney, et al. (2009b). 'Provenance: a future history'. In *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, pp. 957–964. ACM.

D. Crawl & I. Altintas (2008). 'A provenance-based fault tolerance mechanism for scientific workflows'. *Provenance and Annotation of Data and Processes* pp. 152–159.

J. L. David Hall (1997). 'An Introduction to Multisensor Data Fusion'. *Proceedings of the IEEE* 85(1):6–23.

I. Foster, et al. (2002). 'Chimera: A virtual data system for representing, querying, and automating data derivation'. In *International Conference on Scientific and Statistical Database Management*, pp. 37–46. IEEE.

J. Freire, et al. (2006). 'Managing rapidly-evolving scientific workflows'. *Provenance and Annotation of Data* pp.10–18.

A. Happe (2010). 'Agile Provenance'. Master's thesis, Technical University of Vienna.

U. A. A. James Cheney, Amal Ahmed (2007). 'Provenance as Dependency Analysis'. *Database Programming Lan- guages* pp. 138–152.

J. Ledlie, et al. (2005). 'Provenance-aware sensor data storage'. In *Data Engineering Workshops, 2005. 21st Interna- tional Conference on*, pp. 1189–1189. IEEE.

Y. Liu, et al. (2010). 'A provenance-aware virtual sensor system using the open provenance model'. In *Collaborative Technologies and Systems (CTS), 2010 International Symposium on*, pp. 330–339. IEEE.

Y. Liu, et al. (2009). 'A new framework for on-demand virtualization, repurposing and fusion of heterogeneous sensors'.In *Collaborative Technologies and Systems, 2009. CTS'09. International Symposium on*, pp. 54–63. IEEE.

L. Moreau, et al. (2008a). 'Special issue: The first provenance challenge'. *Concurrency and Computation: Practice and Experience* 20(5):409–418.

L. Moreau, et al. (2008b). 'Special issue: The first provenance challenge'. *Concurrency and Computation: Practice and Experience* 20(5):409–418.

K. Muniswamy-Reddy (2010). *Foundations for Provenance-Aware Systems*. Ph.D. thesis, Harvard University Cam- bridge, Massachusetts.

T. Oinn, et al. (2004). 'Taverna: a tool for the composition and enactment of bioinformatics workflows'. *Bioinformatics*

H. Patni, et al. (2010). 'Provenance Aware Linked Sensor Data'. In *2nd Workshop on Trust and Privacy on the Social and Semantic Web, Co-located with ESWC2010, Heraklion Greece*.

L. M. K. Qiang Le (2010). 'Target Localization Using Proximity Binary Sensors'. *Aerospace Conference* .

N. Shrivastava, et al. (2006). 'Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms'. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pp. 251–264. ACM.

Y. Simmhan, et al. (2005). 'A survey of data provenance techniques'. *Computer Science Department, Indiana Univer- sity, Bloomington IN* 47405.

J. Singh, et al. (2007). 'Tracking multiple targets using binary proximity sensors'. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pp. 529–538. ACM.

E. Stephan, et al. (2010). 'Leveraging The Open Provenance Model as a Multi-Tier Model for Global Climate Research'. In *Proc. of 3rd International Provenance and Annotation Workshop (IPAWâĂŹ10), Troy, NY.*

J. F. Susan B. Davidson (2008). 'Provenance and scientific workflows: challenges and opportunities'. *Proceedings of the ACM SIGMOD International Conference on Management of Data* pp. 345–1350.

W.-C. Tan (2007). 'Provenance in databases : Past, Current, and Future'. *IEEE Data Engineering Bulletin* 30:3–12.

E. A. L. Thomas Huining Feng (2008). 'Real-Time Distributed Discrete-Event Execution with Fault Tolerance'. *Pro- ceedings of IEEE Real-Time and Embedded Technology and Applications Symposium* .

J. H. Unkyu Park (2008). 'Provenance in Sensornet Republishing'. In J. Freire, D. Koop, & L. Moreau (eds.), *Prove- nance and Annotation of Data and Processes*, vol. 5272 of *Lecture Notes in Computer Science*, pp. 280–292. Springer Berlin / Heidelberg.