GWD-E
Category: Experimental
INFOD-WG
Author: Eui-Nam Huh

April 24, 2007

An Experiment for Event Dissemination on Grid Environment for Sensor Network

Status of This Memo
This memo provides experimental information to the Grid community regarding implementation issues for the Information Dissemination working group**. It does not define any standards or technical recommendations.** Distribution is unlimited.

Copyright Notice

Abstract
In ubiquitous environment, sensor network technologies have advanced for collecting information of the environment states. With the rapid growth of sensor network technology, it is necessary and important to share the collected sensor data with various and huge users. In order to provide dissemination sensor data, we designed an information dissemination system based on INFO-D standard.
This document describes how we designed the information dissemination system based on INFO-D patterns for sensor network.

Table of Content

1. Introduction

In ubiquitous environment, with rapid advances being made in sensor technology, wireless sensor networks (WSNs) have emerged to collect a lot of information of physical world. WSNs are characterized as an interconnection of many resource-constrained sensor devices and sensing capability, data processing, and wireless communication capabilities. Sensor nodes sense their environments such as temperature, humidity and pressure variations.

Wireless sensor networks are increasingly being deployed in many important applications such as healthcare monitoring for patients, weather or disaster monitoring and forecasting, tracking of goods and manufacturing processes, safety monitoring of physical structures and smart homes and offices.

In WSNs environment, a large number of sensor devices are deployed and aggregated over a wide area for data acquisition and processing. However, WSNs have limited sensing capability, processing power, and communication bandwidth. As managing, processing and storing a huge variety of sensor data are necessary in those environments, it is very important to share geographically distributed computing resources.

In recent years, Grid computing has evolved to coordinate sharing of distributed and heterogeneous resources and to support a very powerful way of managing, processing and storing huge amount of data. Grid (combination of computational Grid and data Grid) provides distributed computational resources to solve large-scale problems as well as seamless access to the storage resources and efficient managing and transmission of large amount of distributed data.

By integrating sensor network with Grid, huge amount of sensor data can be stored, processed and accessed more efficiently. For sharing huge amount of sensor data efficiently, an information dissemination mechanism is very necessary and important. And there are very few researches that integrate Sensor-Grid and Information Dissemination mechanism.

In this document, we have proposed an architecture called ARROW (Adaptive and Reconfigurable ResOurce management for Wireless sensors using grid technology), a combination of Wireless Sensor-Grid and Information Dissemination system, to disseminate huge amount of sensor data to various users. Also we have presented a matching algorithm, called CGM (Classed Group Matching) which finds matched subscriptions efficiently for information dissemination system. Our CGM algorithm focuses on disseminating system corresponding to OGF (Open Grid Forum) INFOD-WG (INFormation Dissemination Working Group) standard [8],[9]and disseminates sensor data employing OGSA-DAI(Open Grid Services Architecture Data Access and Integration) on Grid.

The document is organized as follows: Section 2 briefly reviews related works. Section 3 describes our proposed ARROW architecture. Section 4 presents the matching algorithm to find an event data in the information dissemination system. Section 5 shows the result of performance experiments of our matching algorithm and finally Section 6 concludes the document.

2. Related works

We have designed the information dissemination system integrating SensorGrid and information dissemination mechanism. So, we reviewed the researches about SensorGrid and information dissemination as related works.

2.1 Integration of sensor network with Grid

Recently, research efforts are beginning to study the integration of wireless sensor networks and Grid computing. Researchers in the UK are studying how sensors can be integrated into e-Science Grid computing applications.

Discovery Net provides a Grid based framework for developing and deploying Knowledge discovery service to analysis data which gathered from distributed high throughput sensors. This architecture is proposed for building knowledge discovery application based on Grid environment. Discovery Net infrastructure is based on Kensington system (Chattratichat et al., 1999). Discovery Net infrastructure extends its architecture to enable its functions on Grid environment. Discovery Net infrastructure is designed for interacting with protocols such as data transfer protocols, security, discovery process workflow protocols to integrate with other infrastructures such as OGSA.
CIMA (Common Instrument Middleware Architecture) project aims to "Grid enable" instruments and sensors to integrate with the Grid. But there is a limitation that the middleware architecture is complex to be implemented on simple sensor devices with low computational and processing capability.[4],[5]
SPRING(Scalable Proxy-based aRchItecture for seNsor Grid) provides Grid based Architecture for integrating with wireless sensor network and Grid environment. SPRING architecture provides WSN Proxy which is an interface between the sensor network and the Grid. It can expose sensor resources as Grid services and translates the data of sensors from its original format to an OGSA format such as XML. WSN Proxy integrates connectivity between wireless sensor network and Grid network. New Wireless sensor network can be integrated with sensor Grid by using WSN Proxy so that Sensor Grid can be extendable and scalable. WSN Proxy provides various services such as scheduling, security, availability, QoS in wireless sensor network environment.
SGSIA(Secure Grid-Sensor Integration Architecture) is proposed as a secure and scalable architecture for integrating resource constrained sensor network into the Grid. In the SGSIA project, the architecture has three important tasks. First, at the sensor node, the raw data is filtered to reduce the overall data which is transmitted to gird system. Second, forward the query to the sensor network optimally and selectively. Third, provide a secure communication channel for the sensor network to Grid.

However, Discovery Net architecture is efficient and can deliver good performance for the targeted applications, they are not flexible and not scalable. CIMA and SPRING do not been address that balancing of resources, Grid query caching for optimizing the overall number of queries sent to the WSNs, and lack of access to multiple WSNs and data filtering mechanism in network stack for reducing the volume of data generated at the sensor network end. We have designed the sensorGrid part in the ARROW architecture by considering these problems.

2.2 Information Dissemination

There are high demands in information dissemination systems recently on ubiquitous society. Several researchers propose models of information dissemination system such as SDI system and Rebeca.
In the SDI System (Selective Dissemination of Information System), the model integrates managing, and accessing profiles and disseminates information, which matches to user's profile. The architecture indexes the profiles from all users into a multi-level index structure using XML format. It uses similarity-based construction of profiles and similarity-based matching between incoming documents and profiles. The architecture builds an index that relies on collaborative filtering and support documents in XML format to respond to current needs of system dissemination [8].
Rebeca project provides disseminating dynamic information for the mobile clients. The Rebeca is a publish-subscribe middleware which can help disseminate dynamic information to mobile consumers. The Rebeca platform uses content-based subscriptions to give subscribers fine-grained control over the content they receive. And it enables to implement intelligent filtering mechanisms for relieving the infrastructure load and mobile clients [9].

However, these researches of information dissemination system do not consider integrating sensor networks and Grid. So we designed information dissemination system using INFO-D standard in ARROW architecture.

2.3 INFO-D Standard in OGF(Open Grid Forum)

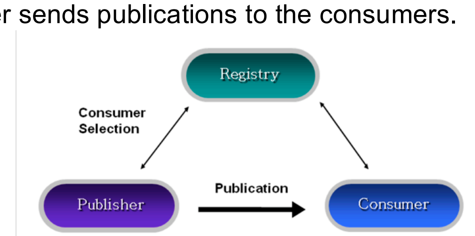
Our disseminating system is designed based on INFO-D standard. The designed architecture is extended and implemented corresponding to a use case for health care described in Info-D patterns [11-12].

The standard defines five entities in the document as follows:

- **Registry** – Each component such as publisher (patients), consumers (doctors), or disseminators should register to the registry with their subscription. Subscription has some information of components such as a name, an address, or what a doctor wants to receive about. Registry provides storing, managing the subscriptions and distributing the subscriptions to the disseminator(s) for disseminating the sensor data to selected consumers.
- **Consumers** – Consumers (doctors) should register to the registry and they can receive patient's data they want.
- **Publisher** – Publisher is a sensor of patient. It sends its status data to our system. The status data is processed and propagated to information disseminator. In the information disseminator, Application specific service works as a publisher.
- **Disseminator**– Disseminator receives the consumer's and publisher's information from registry and schedules disseminating job using this information. Our architecture can handle event data. Disseminator can send the emergency data to consumer immediately when event (emergency) data is collected.
- **Subscribers**– Subscriber support translate consumer/publisher's information format from web page to suitable registry format.
- **Subscriptions** – subscriptions are used to specify consumer's interests.

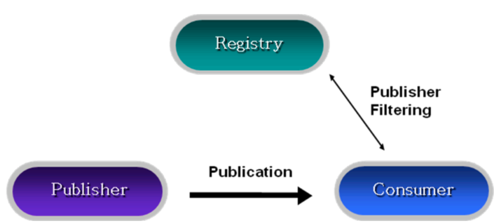
The standard introduces three disseminating patterns as follows:

- **Basic pattern 1** –This pattern shows three entities as registry, publisher, and consumer. Publishers request to the INFOD registry to find matching consumers. After selecting consumers, publisher sends publications to the consumers.



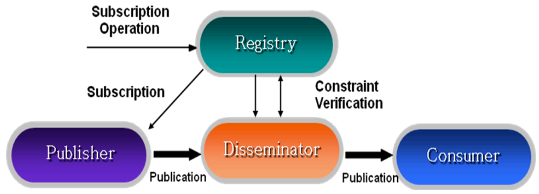
(Figure 1) Basic pattern 1

- **Basic pattern 2** –In this second pattern, the consumer filters messages based on publisher's properties. They request to the INFOD registry to find information of publishers.



(Figure2) Basic pattern 2

- **Single Disseminator** –This pattern includes disseminator to the pattern 1 and 2. In the pattern 3, a disseminator provides sending events to a number of consumers. The disseminator matches the subscriptions of consumers to information of publishers and sends the messages to the selected consumers.



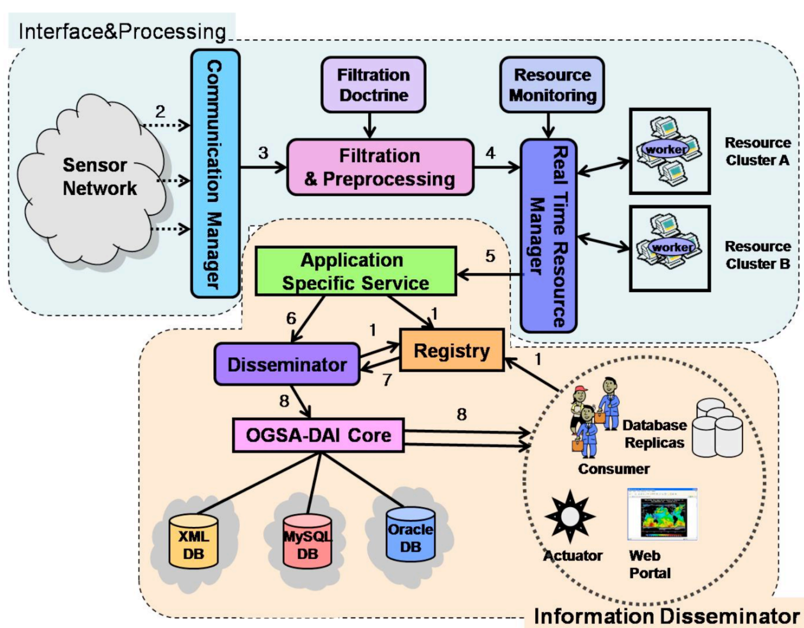
(Figure3) Single Disseminator

Our experiment targets to pattern 3. In the architecture, five entities – sensors or Application specific service as publisher, doctors as consumers, subscriber, registry, and disseminator – are modeled. This will be discussed in detail at section 3.

3. ARROW Architecture

3.1 Overview of ARROW

Our architecture has two parts as processing part and disseminating part. We propose our overall system called ARROW(Adaptive and Reconfigurable ResOurce Management for Wireless sensors using grid technology) which consists of "Interface & Processing" and "Information Disseminator". We will present them in detail of section 3.2 and section 3.3 as shown in figure 4.



(Figure 4) The ARROW architecture

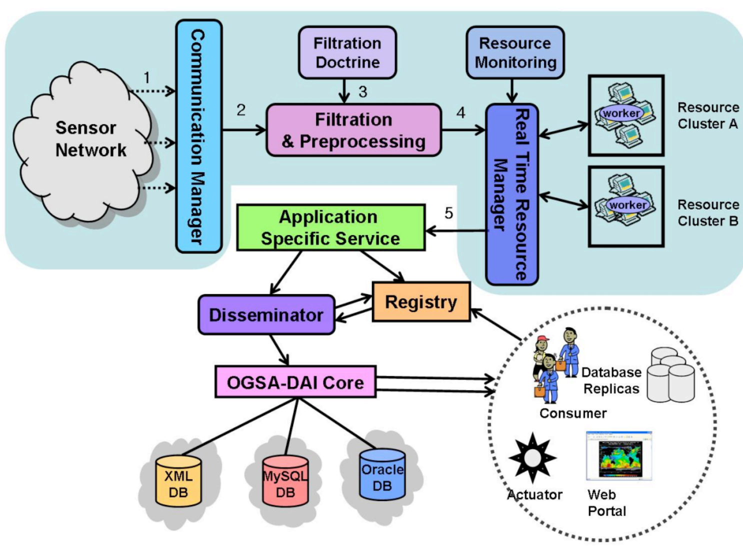
3.2 Interface & Processing

3.2.1. Interface & Processing architecture

The Interface & Processing is designed for collaboration with sensor network and Grid. The raw data can be processed and filtered using Grid resources in this architecture. The Interface & Processing has three components. The definition of each component of the Interface & Processing is explained as follows:

- **Communication Manager(CM)** –It communicates to the wireless sensor devices or wireless sensor gateway. It works as a bridge between wireless entities and mitigates the effect of frequent network disconnection, security and fault recovery. CM also employ important algorithm to retrieve data from wireless sensor devices. After collecting the data from the sensor devices, CM sends it to Filter Manager (FM).
- **Filter Manager (FM)** –FM collects the sensor data and requests for resources to RM (Resource Manager) to aggregate the sensor data and to filter the data using event doctrine file which administrator defined.
- **Resource Manager (RM)** – RM seamlessly serves dispersed Grid resources to FM for filtering sensor data. RM monitors the resource workload to adapt the variations in the resource performance and application requirements.

This system flow as shown in figure 5 is explained step by step as follows;



(Figure 5) Architecture of Processing Interface

Step 1: Sensors sense data from the environment.
Step 2: CM gathers the data and propagates to FM for pre-process and low level filtration.
Step 3: FM performs simple pre-processing sensor data and sends the data and filtration doctrine to RM requesting application level filtration and advanced processing.
Step 4: RM monitors computational resources and allocates the resources to remove the anomalies and inconsistencies of the pre-processed sensor data using the filtration doctrine. RM can reconfigure resource allocation dynamically.
Step 5: RM sends to Application Specific Service to disseminate the data to users using Information Disseminator.

3.3 Information Dissemination Architecture

As we mentioned in section 1, Info-D standard is well explained in [12] by illustrating use cases; NextGRID Graphical Animations use case, car dealer use case, and Sensor Networks use case. Our implementation considering Grid environments need to plug the Data Grid technology. OGSA-DAI, widely used tool, is employed and discussed in following subsection.

3.3.1 OGSA-DAI

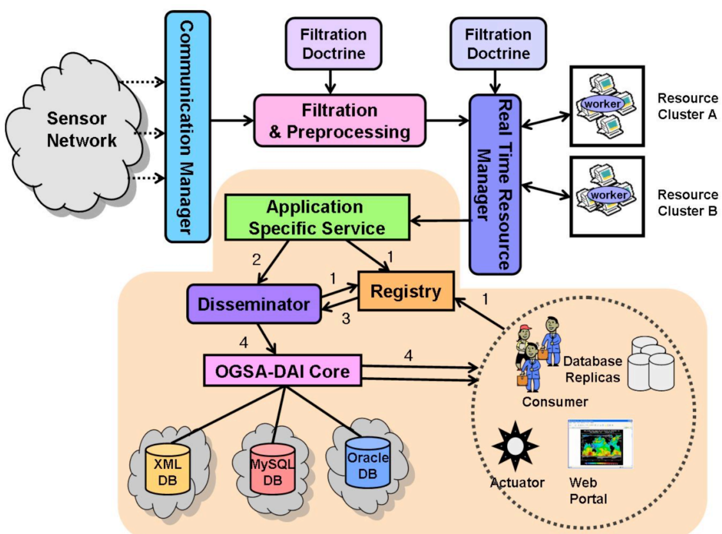
OGSA-DAI [14] handles data resources, such as flat file, relational or XML-databases, to be accessed via stateful web services. It also allows data to be queried, updated, transformed, and delivered. More detail advantages of OGSA-DAI is indicated as follows;

- We don't need additional code to connect to a data base or querying a data. OGSA-DAI support an interface integrating various databases such as XML databases and relational databases.
- The OGSA-DAI provides three basic activities that are querying a data, transforming a data, delivering the results in sequence. The OGSA-DAI can support not only querying and transforming but also delivering using ftp, e-mail, and push services.
- OGSA-DAI allows users to make new activities they want. This function is very scalable and powerful to use the data resource in various ways.
-

It is clean taking advantages of OGSA-DAI for information dissemination in Grid environment

3.3.2 Information Disseminator System flow

Following is the detailed the Information Disseminator system flow as showing in figure 6 ;



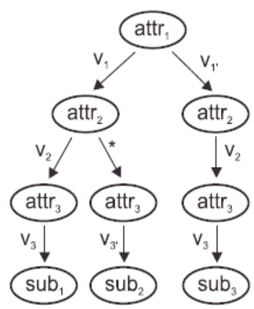
(Figure6) Information Disseminator architecture

Step 1: Each entity should register their information to the registry.
Step 2: Application specific service sends the processed data to the disseminator.
Step 3: Registry can manage, store subscription of each entities and distribute the information to disseminator(s). Disseminator can distribute sensor data to users efficiently using the information

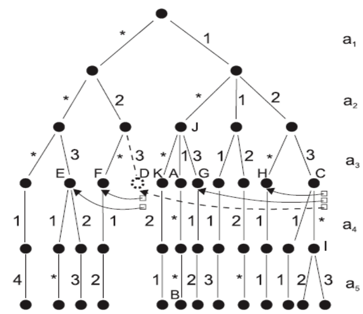
Step 4: Disseminator detects event data for immediately delivery and schedules time based data for regular delivery according to the consumer's subscriptions. It also stores sensor data to heterogeneous and distributed resources in amount of uniform access using OGSA-DAI.

4. Matching algorithm for Information Disseminator

Several researchers proposed event matching algorithms for pub/sub(publish/subscribe) to find event using matching mechanism. In the algorithm [17], authors propose a scheme using tree structure of predicates. They construct a tree and match data by traversing from root(attr1) to sub-nodes(attr2, attr3..) as shown in the figure 7. In the figure, '*' represents a matching condition that means all values are acceptable. When the data reaches the last node, the target subscriptions eventually are matched. Figure 8 shows an example using data=<1,2,3,8,2> with the matching algorithm. The subscription 'I'<1,2,3,*,2> exactly matches to the data and becomes an event to the subscriber.

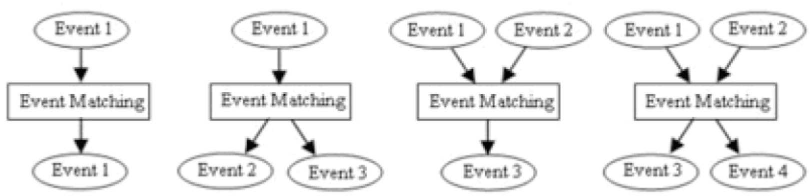


(Figure 7) Subscription tree



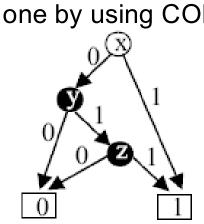
(Figure 8) Predicates construction

The algorithm in [18] triggers events using semantic mechanism in the EAI (Enterprise Application Integration). Figure 9 shows creating events as "on-to-one", "one-to-many", "many-to-one", and "many-to-many".

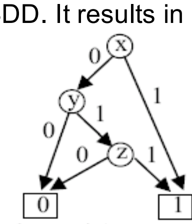


(Figure 9) Patterns of information matching in EAI

After triggering new events, COBDD (Colored Order Binary Decision Diagram) represents their matching procedure as shown in figure 10. Figure 11 shows matching algorithm using OBDD(Order Binary Decision Diagram) for instance x□(y□z). The COBDD can be extended to OBDD as shown in figure 14. They separates x□(y□z) into two predicates, C(x)□O((y□z)). Predicate C and O are different so that they can be combined to one by using COBDD. It results in 0(not matched) or 1 (matched).

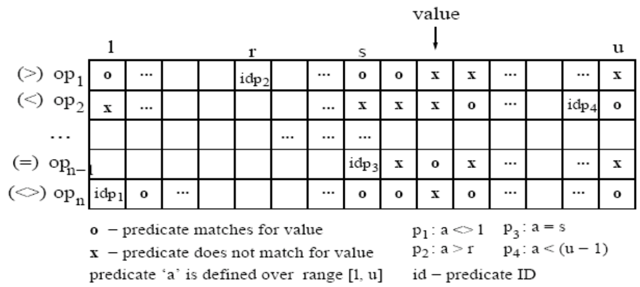


(Figure 10) x□(y□z)



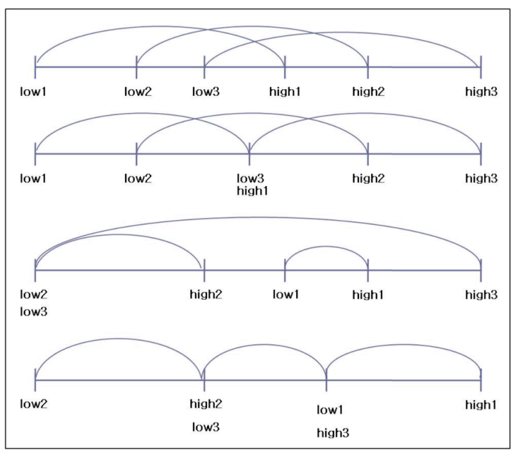
(Figure 11) C(x)□O((y□z))

Another algorithm in [19] uses group of predicates to reduce matching time assuming that subscriptions may have many identical predicates. The proposed scheme reduces resources consumption and matching time efficiently, while redundant sequential matching wastes computational resources. As shown in figure 12, grouped index of predicates is used to search subscriptions joining predicate id and subscription id.



(Figure 12) Grouped Index of predicates

These algorithms result a value, when the predicate of subscription matches to a specific constant. This condition can be resolved simply using tree based sequential matching as solved in [17][18][19]. However, if predicate's condition requires certain range like u-Health sensor event processing system that consumers (doctors and nurses) express their interests into a range, then the problem become harder and difficult to apply above techniques to trigger events. As patient states are changed continuously within certain range(min, max) like body temperature and heartbeat, each classified consumer will require events in various range to take care patients safely. So, ranged predicates are more efficient to describe consumer's interests. Furthermore, the ranged predicate has several conditions as equivalence, subset, or intersection as shown in figure 13. In order to match efficiently to this situation, we propose a novel scheme called "Classed Group Matching(CGM)".



(Figure 13) Ranged Predicates on u-Health event processing system

The specific feature challenges of the proposed system are as follows:

1) The information dissemination (denoted to Info-D) system should send sensor event messages to consumers directly on triggering and deliver them every period subscribed.

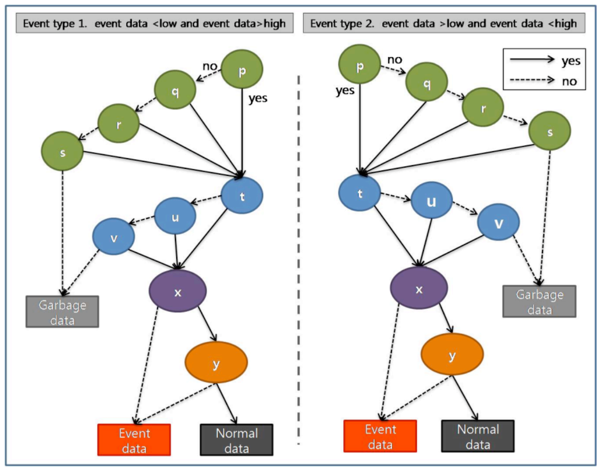
2) The Info-D system should support several types of data such as blood pressure, temperature, pulse, or electrocardiogram in u-Health environment.

3) The Info-D system should allow subscribers to describe two ranged predicates, (low_threshold_value< observed_data < high_threshold_value) and (low_threshold_value>= observed_data, high_threshold_value=<observed_data)

4) The Info-D system need to process efficiently to handle a number of subscriptions of various consumers.

4.1 Event detection model

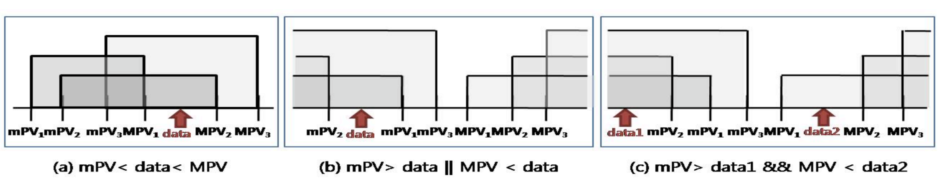
We designed the event detection model showing detection of event is triggered by period or on time using BDD(Binary Decision Diagram). Sensor data include not only simple sensed value but also a department of hospital such as Circulatory organ, Digestive organ, Respiratory organ, and kidney organ (denoted to p, q, r, s, respectively) and event type such as blood pressure, temperature, blood pressure (denoted to t, u, v, respectively shown in figure 18). In figure 18 showing our BDD model, there are four types of node to find which data belongs to the range of predicates. Node x represent a minimum threshold value of the range and node y for maximum threshold value of the range in a predicate of the subscription. Event triggering, which is shown in figure 14, needs to match a sensed data to all predicates of subscriptions. This causes complexity on matching in the time and space domain, so we need an efficient matching algorithm considering various types of subscriptions requested by many subscribers such as nurses, staffs, and doctors. Therefore, next sections, 4.2 and 4.3, introduce our effort to design of an efficient matching algorithm for pub/sub systems.



(Figure 14) Simple matching BDD model

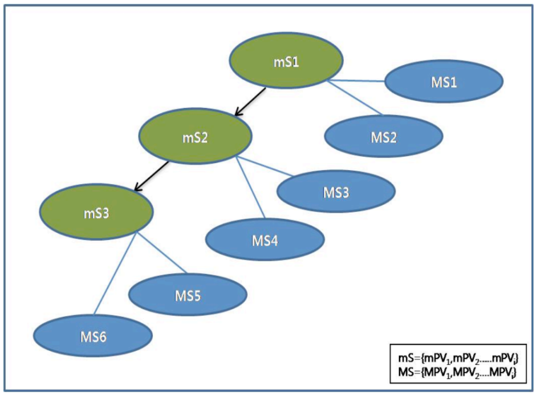
4.2 The grouping of predicates

Subscriptions are composed of consumer's personal information and predicates which show consumer's requirement. Each predicate, P, constructs of five elements such as P={event_type, expression, op, mPV, MPV}, where the 'event_type' for detail event name, 'expression' for expression of predicates,(i.e. '1' for single data matching with mPV and MPV, and '2' for two sensed input data to match mPV and MPV at the same time. The 'op' represents an operation such as "<<", or "<>". Also, mPV and MPV stand for minimal threshold value of the predicate and maximum threshold value of the predicate, respectively. The Information Dissemination system in the ARROW supports various expressions of predicates. First, "(mPV< data < MPV)" is used when consumers want to know normal patterns of sensed data as shown in figure 15-(a). Second, "(mPV>data || MPV<data)" is used when consumers need to receive unusual states of the patient as shown in figure 15-(b). Third, "(mPV> data1 && MPV<data2)" is used to support for two outputs of a sensor such as blood pressure as shown in figure 15-(c).

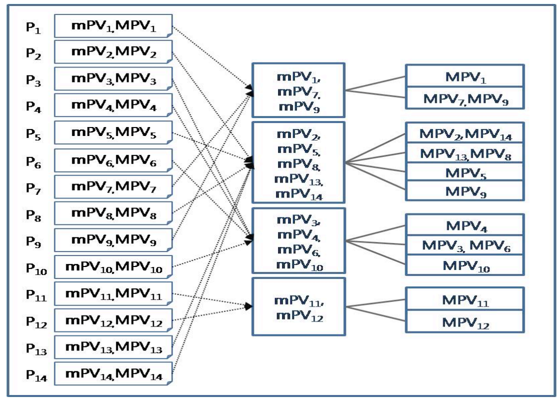


(Figure 15) Expression case of CGM

As we discussed in section 4.1, a data which should be delivered on time will be matched to all predicates of subscriptions exactly. We employed the grouping scheme to reduce overhead in matching as shown in figure 16. First we grouped all minimal threshold value of predicates from all subscription denoted to mS for the set of mPVs, i.e. mS={mPV1,mPV2…..mPVi} and MS for the set of MPVs, i.e. MS={MPV1,MPV2....MPVi}. The figure 17 describes grouping technique of predicates in detail.



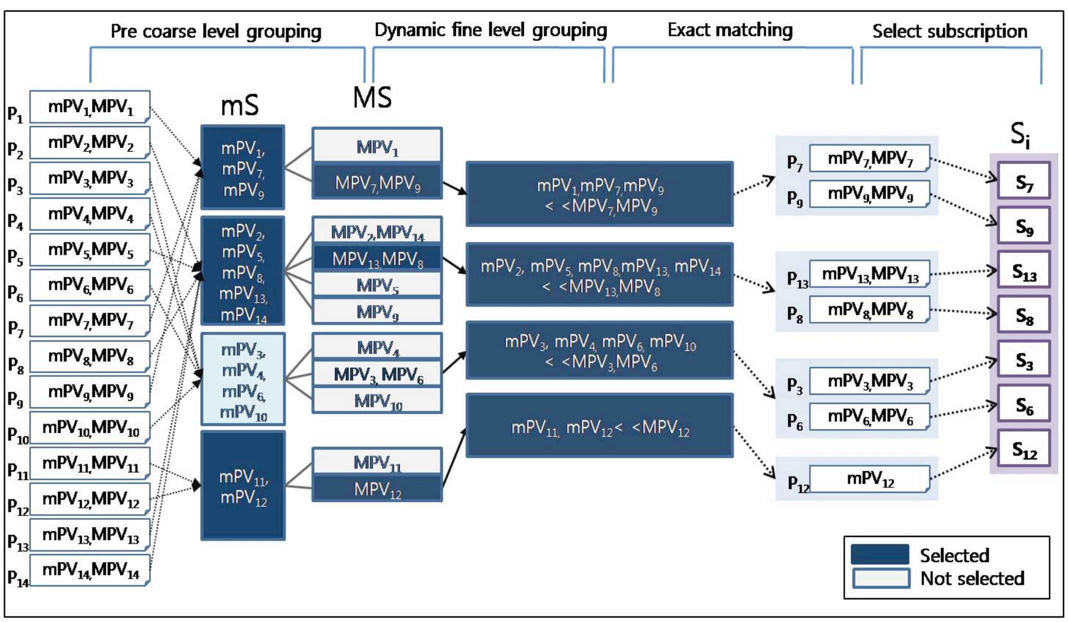
(Figure 16) Grouping tree



(Figure 17) Grouping predicates

4.3 CGM (Classed Group Matching) Algorithm

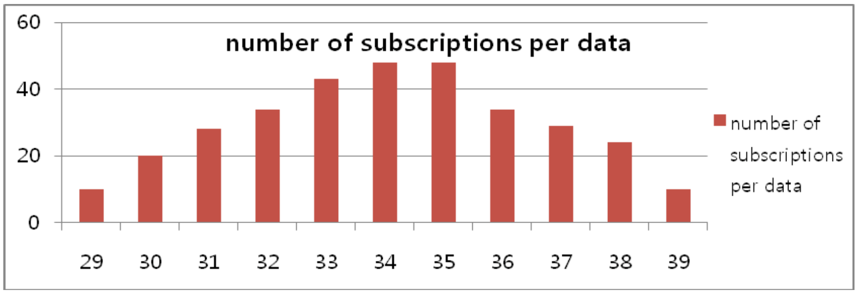
In this section, we explain the procedure of our matching algorithm with figure 18. First, coarse level grouping partitions mS and MS among all subscriptions into small groups explained in section 4.2, which is done at the system initialization. Second, dynamic fine level grouping compares sensed data to maximum mPV. If matches, then comparison with MS within same predicate's ID as mS is tested. This step produces feasible candidates, which is not enough to be the final predicate, as each ranged predicate can not be matched to the set, mS and MS. So, third, exact matching step is required to compare data with each predicate in selected predicate groups. Finally, the algorithm finds subscriptions by joining selected predicate id(s) and subscription id(s).



(Figure 18) Procedure of matching algorithm

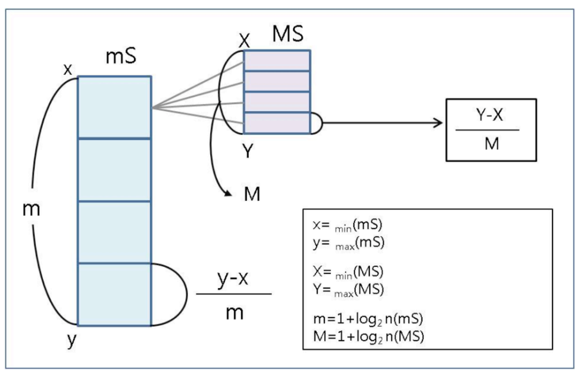
5. Experiment

In this section, we exam our CGM algorithm on u-Health care system with various scenarios. As the range of the human body conditions has limitation, consumer's interest(subscription) is also within reasonable boundary. Figure 19 shows subscription's distribution of body temperature submitted by nurses, staffs and doctors.



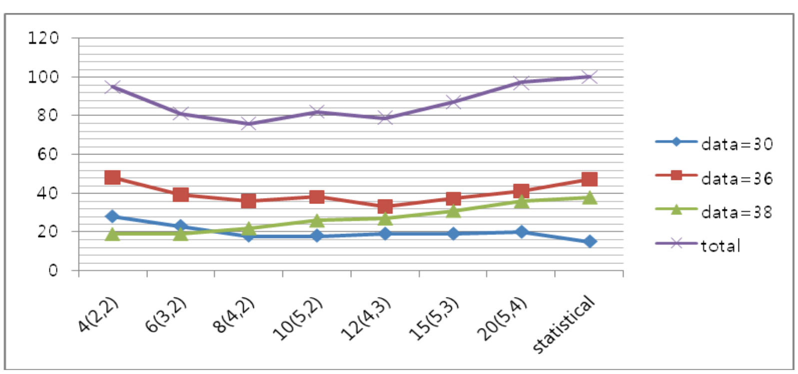
(Figure 19) The numbers of subscriptions per data

The basic partitioning idea is employed to decide the number of class for mS and MS (denoted to m and M, respectively) by m=1+log2 n(mS), and M=log2 n(MS), where n(mS) is the number of element in the set, mS. The statistical method decides an equivalent interval of each class by (y-x)/m and (Y-X/M) for mS and MS, respectively. Figure 20 shows the detail of partitioning.



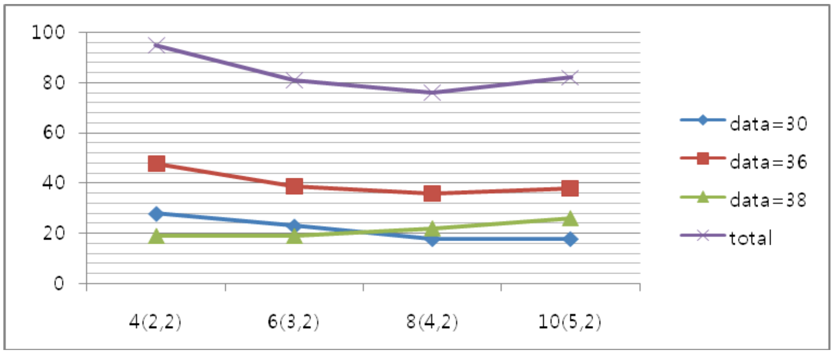
(Figure 20) Statistical partitioning

We simulated experiment with predicates case1. In our experiment, we used 50 subscriptions and 3 sensed data (data=30, data=36, data=38) with statistical class and various number of classes and CGM is executed. Figure 21 shows the result of our first examination.



(Figure 21) Event triggering with various number of classes

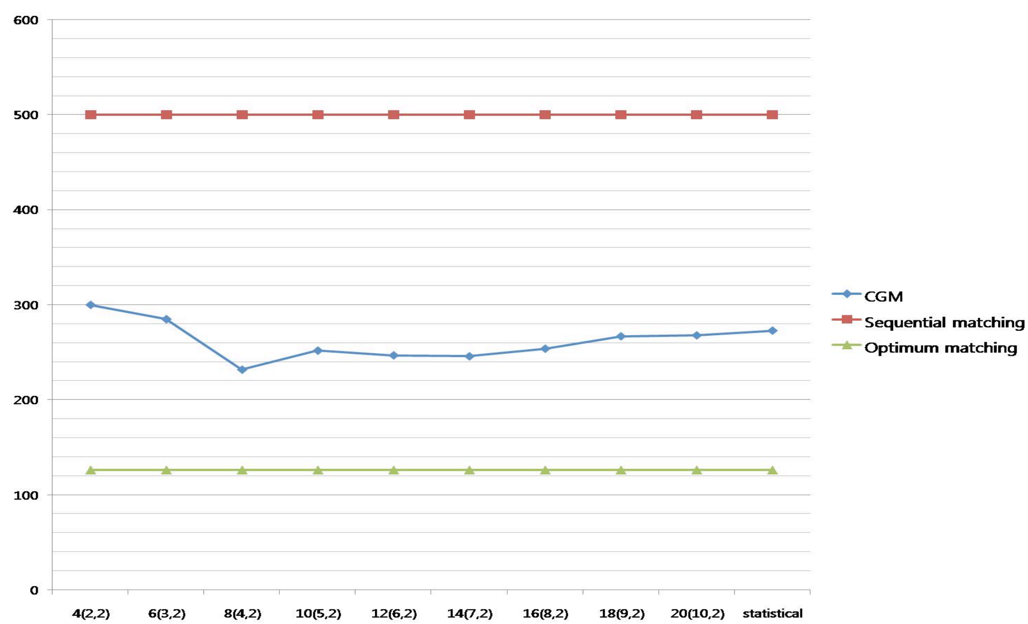
From the figure 21, we found that CGM performs well and different number of classes affects matching performance. The x axis in figure 21 shows the different number of classes. So the "10(5,2)" indicates 5 predicates in each mS class, 2 predicates in each MS class, and 10 means total number of classes. In here statistical approach, 7 predicates are equally assigned to each class.
According to the figure 21, and 22, searching time also depends on 2 factors: input data and the number of classes. If sensed data is very low or high value, the matching will go through to the end of class by traversing tree as shown in BDD model. Also the number of predicates in a class affects comparison time. The figure 22 shows the impact of matching time on varying number of classes.



(Figure 22) Matching simulation with the change of mS

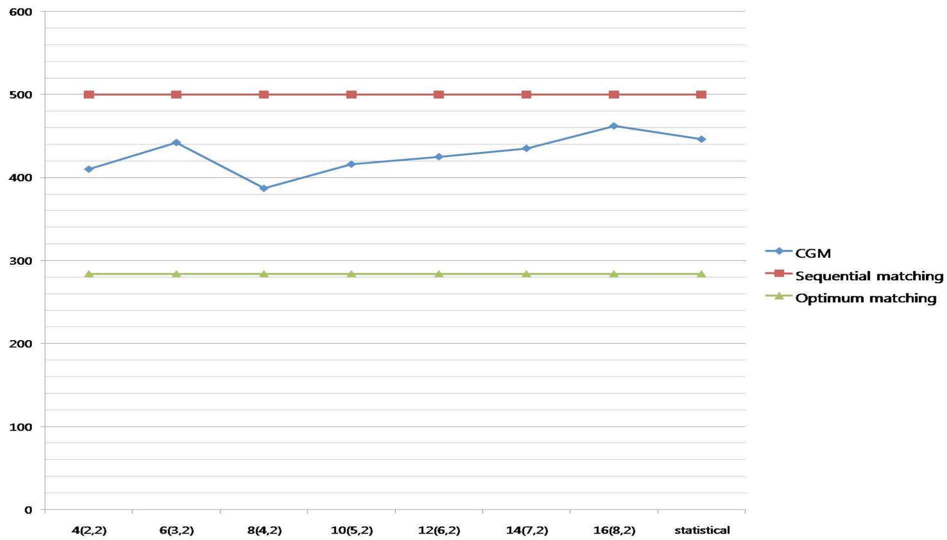
The pub/sub system needs to handle various cases of predicate expressions for diversity of subscribers. As we mentioned, three different cases are employed in CGM. The experiment for each case is explained below.

-Case 1: mPV < data < MPV
Following experiment shows the total time of matching for all subscriptions and matching for grouped subscriptions using CGM. Figure 23 shows the matching time of the CGM algorithm under different numbers of groups with 5 sensed data for the predicate case 1. If the system matches all subscriptions sequentially for 5 data, it takes 500 matching, while the proposed CGM matches 262 times on the average. The CGM improves 48% of the system performance.



(Figure 23) Matching for the predicate case 1

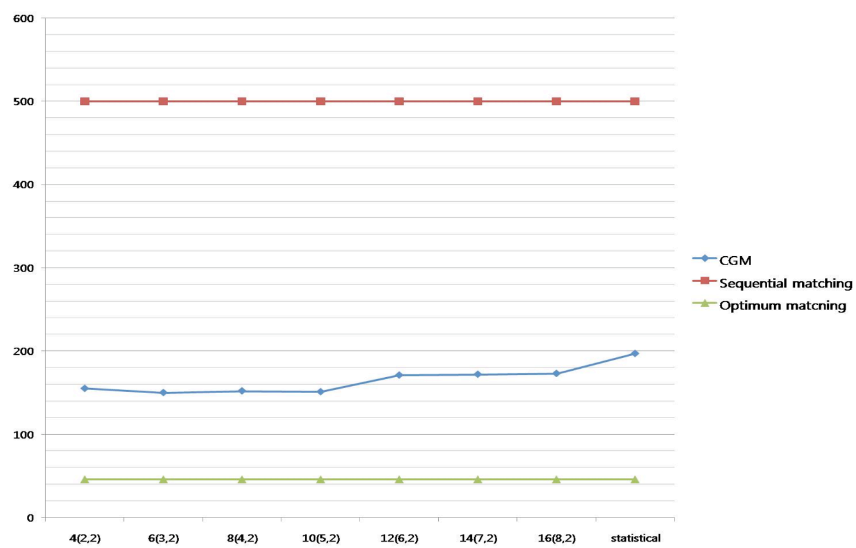
- Case 2: mPV>data || MPV<data
This experiment is done for the case 2 as same condition as Case 1.



(Figure 24) Matching for the predicate case 2

The CGM takes 427 matching time on the average. If the system matches sequentially all subscriptions, 216 matching times is wasted, while CGM takes 143 unnecessary matching times compared to the optimal searching case. Only for case 2, simple algorithm can match better, however, note that the dissemination system need to handle various expressions with an identical algorithm.

- Case 3 : mPV>data1 && MPV<data2
This case is designed for pair matching in the case of 2 sensed inputs. In this experiment, we measured the total time of matching for 4 pair of subscriptions such as (mPV=29>data1, MPV=38<data2), (mPV=29>data1, MPV=33<data2), (mPV=33>data1,MPV=37<data2), and (mPV=32>data1,MPV=35<data2).



(Figure 25) Matching for the predicate case 3

As shown in figure 25, the sequential searching wastes 408 matching times, while CGM takes 165 times and wastes only 119 matching times compared to the optimal searching case.

- Glossary of information disseminator

As we explain the CGM algorithm for information dissemination, it has the following contributions:
1) Provision of time and event driven delivery
2) Allowing multiple event data generation with ranged predicates
3) Flexible expression for diverse consumers
4) Efficient matching and partial optimization

6. Conclusion

In this document we have proposed the ARROW architecture which integrates sensor network and Grid for information dissemination. The ARROW supports a gateway between sensor network and Grid to process, manage huge amount of data, provide interfaces for consumers, and employs the pub/sub system. An efficient matching algorithm called CGM considering three types of predicates reduces matching overhead significantly.
In the future research, we will find the relation between the number of groups and matching time, and an efficient algorithm will be proposed to find appropriate the number of class rather than well-known statistical method.

Author Information

Eui-Nam Huh
Kyung Hee University
Koera
Johnhuh@khu.ac.kr

Intellectual Property Statement

Full Copyright Notice

References

1. YAN YuJie, WANG Shu , Hao Zhao, "MPAS: a Connecting Platform for Integrating Wireless Sensor Network with Grid", 2005 Asia-Pacific Conference on Communications.

2. Weidong Bao, et. al."VECON: a Cooperative Event Detection System for Video Sensor Grid",Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC'06)

3. Salman AlSairafi, et. al. Michelle Osmond, Anthony Rowe, Jameel Syed, Patrick Wendel, “THE DESIGN OF DISCOVERY NET: TOWARDS OPEN GRID SERVICES FOR KNOWLEDGE DISCOVERY”, The International Journal of High Performance Computing Applications, Volume 17, No. 3, Fall 2003, pp. 297–315

4. R. Bramley, et. al. "Instruments and sensors as network services: Making instruments first class members of the grid". Technical Report 588, Indiana University CS Department, December 2003.

5. Hock Beng Lim, et. al. "Sensor Grid: Integration of Wireless Sensor Networks and the Grid", Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary (LCN’05)

6. Chen-Khong Tham, Rajkumar Buyya, "SensorGrid: Integrating Sensor Networks and Grid Computing", CSI Communications, pages 24-29, Vol.29, No.1, ISSN 0970-647X, Computer Society of India (CSI), Mumbai, India, July 2005.

7. Zubair A. Baig, Mohamed Baqer, Asad I. Khan, "SGSIA-In-Network Data Preprocessing for Secure Grid-Sensor Integration", Proceedings of the Second IEEE International Conference on e-Science and Grid Computing (e-Science'06)

8. Gero Mühl, Andreas Ulbrich, Klaus Herrmann, and Torben Weis, ” Disseminating Information to Mobile Clients Using Publish–Subscribe”, Vol. 8 ,46-53, 2004, ISSN:1089-7801,IEEE Internet Computing

9. Ioana Stanoi, George Mihaila, Sriram Padmanabhan, “A Framework for the Selective Dissemination of XML Documents based on Inferred User Profiles”, Proceedings of the 19th International Conference on Data Engineering (ICDE’03)

10. Open Grid forum, “http://www.ogf.org”
11. Vijay Dialani, Ronny Fehling , et. al. “INFOD Base Specification”, www.ogf.org

12. “INFOD Patterns”, www.ogf.org.

13. “INFOD_Base_Use_Case_Scenarios_for_GGF16”, www.ogf.org

14. “www.ogsadai.org” , OGSA-DAI

15. “Senor Model Language(SensorML)”, http://vast.nsstc.uah.edu/SensorML/

16. J.Shneidman,P.Pietzuch, et.al. “Hourglass: An Infrastructure for Connecting Sensor Networks and Applications”, Harvard Technical Report TR-21-04