Measuring the Benefits of Summarizing Quantitative Information in Pervasive Computing Systems

Faraz Rasheed, Young-Koo Lee, Sungyoung Lee

Kyung Hee University, Suwon
449-701, Republic of Korea
faraz@oslab.khu.ac.kr, {yklee, sylee}@khu.ac.kr

Abstract

Ubiquitous computing systems continuously sense the environment and collect different kind of data used to provide un-obtrusive, proactive services to users. The amount of data collected grows tremendously consuming considerable storage space, slowing down the query processing and data analysis tasks and negatively effects the distributed data management and data transfer among various nodes. We have proposed earlier that a compact, summarized and aggregate representation of historical data can be used to overcome these problems. In this paper, we analyze the effectiveness of this concept by summarizing the quantitative information using various techniques and calculating different parameters as storage space consumption, time taken to answer queries and the precision of the results provided.[1]

1. Introduction

Ubiquitous Computing envisions [1] an environment where devices and sensors can interact seamlessly to provide proactive services to users. There are a number of projects currently being done in ubiquitous computing field [2][3][4]. Context awareness is an integral feature of ubiquitous computing systems. In order to provide proactive services to user, a system must be aware of the users environmental context and user preference. Although context is still a vague term, different researchers have provided different definition to context, we take context as the 'situational understanding' and anything that describe this situational understanding as context information or simply context.

Usually a ubiquitous system is connected with a number of sensors which sense the physical and computational environment, the users present in the environment and their activities. Hence various types of data are continuously injected to the ubiquitous computing system. The size of this data grows tremendously with time slowing down the query processing and other database operations. This huge amount of data requires careful management to address the issues in ubiquitous data management [5].

We proposed in our earlier papers the idea of knowledge aggregation which we call Context Summarization [6][7][8]. The idea is to represent the information in such a form that it takes comparatively less storage space and can still answer queries with appropriate precision. We have proposed several techniques that can be applied on different type of data.

Our summarization process is totally hidden from the user and she still queries the data as she is using the original data. Since the summarization process change the representation of the data transparently, there is a need for query answering system which can translate the queries on original data so that they can be answered using the aggregate information store. In this paper we are going to describe this query answering component of our summarization system. We will explain how this query answering system can be applied to numerical type quantitative context information. Numerical type quantitative context information includes temperature values, humidity values, available network bandwidth at particular time, network bandwidth usage by particular user, light intensity, etc.

2. Context Summarization

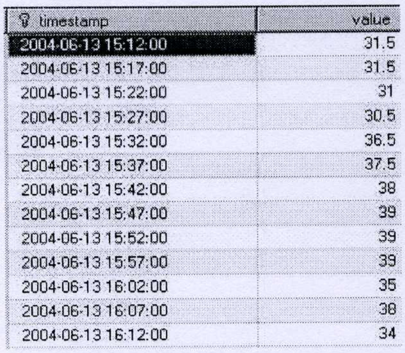
Context summarization is the compact and aggregate representation of raw data consuming less storage space having the capability to answer the queries with appropriate confidence. The summarization can either be performed just as the data is received from sensors (instantaneous summarization) or after the data is received and stored in the repository for future use (delayed summarization). We are basically focusing on

[1] This research is supported by the Driving Force Project for the next generation of Gyeonggi Provincial Government in Republic of Korea
Professor Sungyoung Lee (sylee@khu.ac.kr) is the corresponding author for this paper

delayed summarization where we aggregate the data stored in repository to introduce a new compact, aggregate and summarized data repository.

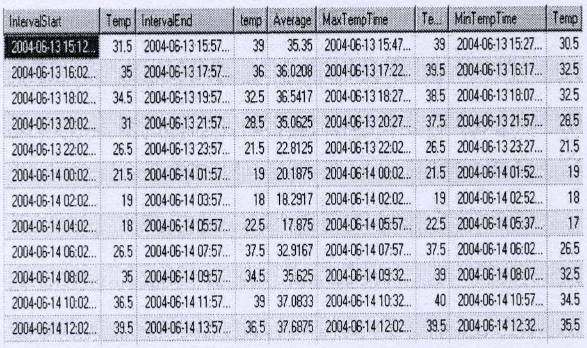
Most of the data received by pervasive system is for the software system's (middleware and application) internal use so that they can process it and provide the proactive services to users promptly and efficiently. Thus we can change the internal representation of this data such that most of the queries can be answered using the summarized information. Consider a location 'X' in a ubiquitous system; we have got few temperature sensors giving us the temperature values after every 5 minutes. Figure 1 presents a sample of data received.

Using Context Summarization, in the simplest case we can record the average temperature values over longer intervals of time along with recording maximum and minimum temperature (See Figure 2)



timestamp	value
2004-06-13 15:12:00	31.5
2004-06-13 15:17:00	31.5
2004-06-13 15:22:00	31
2004-06-13 15:27:00	30.5
2004-06-13 15:32:00	36.5
2004-06-13 15:37:00	37.5
2004-06-13 15:42:00	38
2004-06-13 15:47:00	39
2004-06-13 15:52:00	39
2004-06-13 15:57:00	39
2004-06-13 16:02:00	35
2004-06-13 16:07:00	38
2004-06-13 16:12:00	34

Figure 1. Raw temperature sensed after every five (5) minutes



IntervalStart	Temp	IntervalEnd	temp	Average	MaxTempTime	Te...	MinTempTime	Temp
2004-06-13 15:12...	31.5	2004-06-13 15:57...	39	35.35	2004-06-13 15:47...	39	2004-06-13 15:27...	30.5
2004-06-13 16:02...	35	2004-06-13 17:57...	36	36.0208	2004-06-13 17:22...	39.5	2004-06-13 16:17...	32.5
2004-06-13 18:02...	34.5	2004-06-13 19:57...	32.5	36.5417	2004-06-13 18:27...	38.5	2004-06-13 18:07...	32.5
2004-06-13 20:02...	31	2004-06-13 21:57...	28.5	35.0625	2004-06-13 20:27...	37.5	2004-06-13 21:57...	28.5
2004-06-13 22:02...	26.5	2004-06-13 23:57...	21.5	22.8125	2004-06-13 22:02...	26.5	2004-06-13 23:27...	21.5
2004-06-14 00:02...	21.5	2004-06-14 01:57...	19	20.1875	2004-06-14 00:02...	21.5	2004-06-14 01:52...	19
2004-06-14 02:02...	19	2004-06-14 03:57...	18	18.2917	2004-06-14 02:02...	19	2004-06-14 02:52...	18
2004-06-14 04:02...	18	2004-06-14 05:57...	22.5	17.875	2004-06-14 05:57...	22.5	2004-06-14 05:37...	17
2004-06-14 06:02...	26.5	2004-06-14 07:57...	37.5	32.9167	2004-06-14 07:57...	37.5	2004-06-14 06:02...	26.5
2004-06-14 08:02...	35	2004-06-14 09:57...	34.5	35.625	2004-06-14 09:32...	39	2004-06-14 08:07...	32.5
2004-06-14 10:02...	36.5	2004-06-14 11:57...	39	37.0833	2004-06-14 10:32...	40	2004-06-14 10:57...	34.5
2004-06-14 12:02...	39.5	2004-06-14 13:57...	36.5	37.6875	2004-06-14 12:02...	39.5	2004-06-14 12:32...	35.5

Figure 2. Temperature data summarized (aggregated) over an interval of two (2) hours

Such a summarization not only reduces the storage space consumption but also improves the time taken to answer queries. As the ubiquitous systems are essentially distributed, such a summarized representation also improves the network usage for data migration among different nodes. It also improves the efficiency of knowledge reasoning, user preference and machine learning and data mining by minimizing the size of available data and thus the query processing time.

Since the queries are not answered by using the actual data but by using the summarized, compact representation; mostly there is a precision lost and we can not answer the queries with 100% confidence. We try to maximize this precision value and confidence in our techniques and associate a confidence value with each query result.

Although we try to answer most of the queries using the summarized information, we are not replacing the actual repository with the summarized repository in our current implementation. We still preserve the original data and use it rarely to answer sensitive queries which demand 100% accuracy and can bear some processing delays. Usually, we do not distribute the original repository to different nodes in distributed environment and we only distribute the summarized information among the nodes when needed. Hence querying the actual data may also include the network delays for data migration and data sharing process.

We are using our project called CAMUS (Context Aware Middleware for Ubiquitous Systems [4]) to apply the summarization techniques. Hence, all these techniques are implemented inside the middleware which also host the knowledge repository and provides query interface to user. In the next section we will explain the architecture of our system and in the subsequent section we will show how do we apply the summarization techniques and answer queries over numerical type context and location information. For the architecture of our system see our previous publication [7].

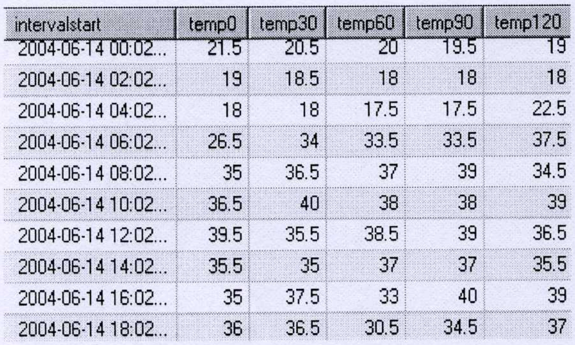
3. Applying Query Answering Interface

Now we will demonstrate the application of summarization and query answering using two data sets as examples. First we will consider temperature values to show how numerical valued context information can be summarized and apply the queries over it. Secondly, we will consider the location information of users, summarize it and apply the query answering system to retrieve results of user queries. We will also mention the benefits and shortcoming of each approach. Another worth mentioned point is that the ubiquitous systems are very resource hungry as they are performing a lot of operations; receiving continuous stream of data from sensor, data filtering, modeling, storage, data processing, reasoning, situational understanding of current context, finding and providing the appropriate service and so on. Hence, the summarization module should be very smart and efficient. That is the reason why we have selected to apply only simple, optimized and less complex techniques to summarize the information and to calculate and return the query result.

3.1. Numerical Valued Context Information

Let we have a temperature sensor reporting the temperature of location 'X' after every 5 minutes. Figure 1 presents a sample data set. Now let we summarize the information using averages as presented in Figure 2 containing time interval end points, temperature values at these end points, the average temperature, the extreme temperature values and the timestamp for the extreme temperature values.

Another possible summarization is to sample the temperature values over elongated interval say after every 30 minutes instead of every 5 minutes as received from sensor. Figure 3 presents a sample summarization using this approach.



intervalstart	temp0	temp30	temp60	temp90	temp120
2004-06-14 00:02...	21.5	20.5	20	19.5	19
2004-06-14 02:02...	19	18.5	18	18	18
2004-06-14 04:02...	18	18	17.5	17.5	22.5
2004-06-14 06:02...	26.5	34	33.5	33.5	37.5
2004-06-14 08:02...	35	36.5	37	39	34.5
2004-06-14 10:02...	36.5	40	38	38	39
2004-06-14 12:02...	39.5	35.5	38.5	39	36.5
2004-06-14 14:02...	35.5	35	37	37	35.5
2004-06-14 16:02...	35	37.5	33	40	39
2004-06-14 18:02...	36	36.5	30.5	34.5	37

Figure 3. Temperature values sampled at larger interval

Now suppose we need to answer the following query: **Query:** *Give me the temperature of location 'X' at time 03:45*
(SELECT tempValue FROM temp WHERE loc='x' AND time='03:45')
There are several ways to answer this query
Ans 1: 8.5 (*Average value for the interval that contains the time 3:45*)
Ans 2: 6.3 (*Temperature value at the initial point (or the closest of initial and final point) of interval*)
Ans 3: 8.6 (*the missing point using curve fitting and interpolation see figure 3*)

For Ans 3 using summary presented in Figure 2, we may use the curve fitting with initial and final points of interval and placing the average at center of the interval, as in figure 4. Another improvement could be made by also plotting the extreme (max and min) temperature values at the timestamp for these extreme values. This adds two more points in the interval and results in better estimation of the missing point. Section 4 presents these in more detail
For Ans 3 using summary presented in figure 3, we may use the curve fitting with five points representing the sample space or can calculate the missing point using linear or non-linear interpolation techniques discussed in the next section.

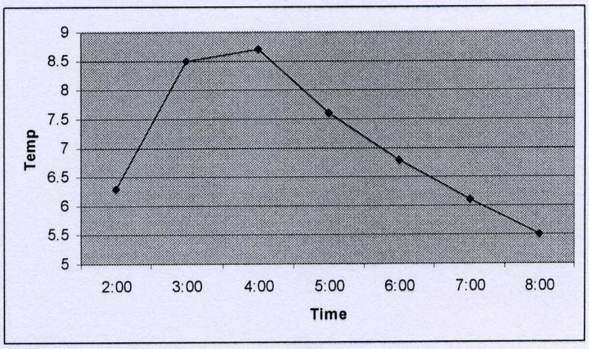


Figure 4. Temperature values plotted over intervals with averages at center

As we go down in the list of answers, the complexity to answer queries increases; thus taking more time for query answering. This time can be reduced by (a) considering the simplest case (Ans 1 and 2), this option is useful when interval size is small. (b) as the considered time interval gets larger, it is found useful to apply more comprehensive techniques (Ans 3 with linear and non-linear interpolation). We can also use these techniques flexibly by considering the current load on the system and/or considering the required precision for query results. These techniques can be applied to summarize other quantitative context information like humidity, noise level, light intensity, available network bandwidth, and bandwidth usage by particular user or at particular location.

4. Experiments and Evaluation

We have evaluated the summarization of numerical valued quantitative information from three perspectives. First of all we have compared the storage space consumption by raw data and information summarized by different techniques. Secondly, we applied the similar queries on both raw data and summarized information and compared the time taken by both of these representations. Finally, we calculated the average random error incurred while answering the queries using various summarized representations. Now, first we will describe our system specifications and then in the following subsection (4.2) we will evaluate and discuss the summarization over numerical valued context information (temperature)

4.1. System Specification

We performed all the experiments on Toshiba notebook using Pentium Mobile using Intel Centrino technology with 1500 MHz processor and 512 MB RAM. We are using Windows XP Professional OS. Our summarization module is implemented using Java programming language. We used mySQL v4.0 as DBMS (running on localhost) and also used MATLAB 6.5 for interpolation techniques. The system was not considerably busy with other process while we performed the experiments and we did not use multiple threads for processing.

4.2. Numerical Valued Context Information

We used the data recorded by Cornell Lab of Ornithology for Bird population studies (USA). They installed a temperature logger in bird's nest and recorded the temperature after every 5 minutes. We used the data recorded in 88 days which amounted to around 22,000 records. We summarized this data over 2 hours interval using 5 different techniques as described in section 3.1. Briefly, in the first technique we used averages to approximate the query result. In the second technique, we return the temperature nearest of initial & final point of interval. In the third technique, we interpolated the data using five points sampled at regular interval (referred in figure as RegInt). In the fourth technique, we interpolate for required point using initial, final and average point where average point is plotted at the center of the interval (referred in figure as 3PtInt). In the fifth technique, we interpolate for required point using initial, final, average (at center), maximum and minimum temperature point (referred in figure as 5PtInt). We used linear, nearest point (NPI), cubic spline (CS), piecewise hermite (HI) interpolation techniques to approximate the missing points. The effect on the storage space consumption by the summarization process is presented in the Table 1

Table 1. Storage space consumption comparison

	Raw data	Summary data
Rows	21,978	924
row size	48	97
table size	1035 KB	90 KB
Days	88	88
Interval	5 min.	120 min.

Obviously, the summary table takes less storage space than raw data. The row size in summary table is larger than that of raw data table as it contains more fields. The same information for 16 days is presented in graphical form in Figure 5.

Now for the time taken by query processing, we applied the simple query to find the temperature at a specific time over 22,000 raw data records and 924 summary records. We calculated the time consumed to reply this query over raw data and the summary data calculated by 5 different techniques described earlier. Chart in figure 6 compares the time taken to answer ten random queries by raw data table and different summaries with different interpolation techniques

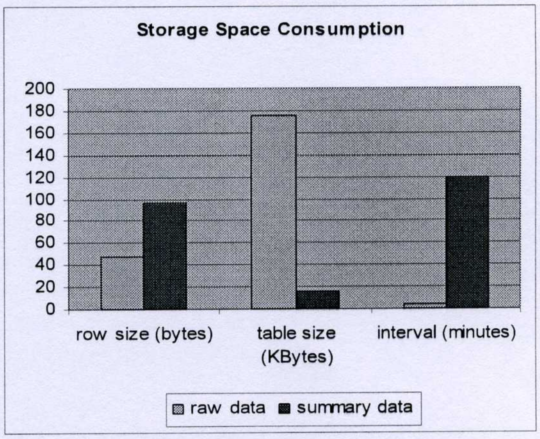


Figure 5. Comparison of storage space consumption

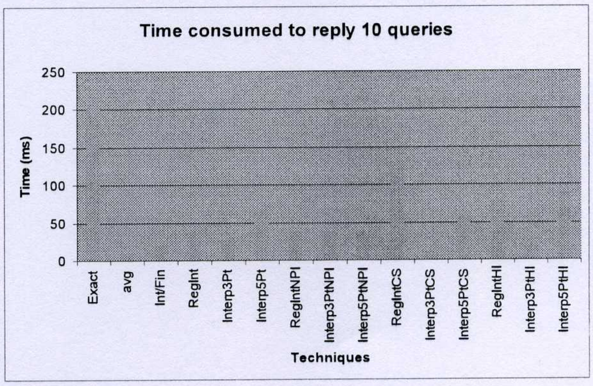


Figure 6. Time taken to reply queries

The time taken to answer queries using raw data is four times the time taken using summary data. This is because the raw data is querying over much larger table (22,000 records versus 924 records of summary table). The best performers are average, closest of initial and final point, regular interval and 3 point interpolation.

Finally, to calculate the lost in precision caused by these queries, we averaged the difference between actual value and those produced by summary tables. We considered an interval from 2004-06-16 to 2004-06-19 and picked randomly ten timestamps in each interval of two hours and calculated error using our considered techniques. Figure 7 presents a chart comparing the random average error values for each of the considered techniques.

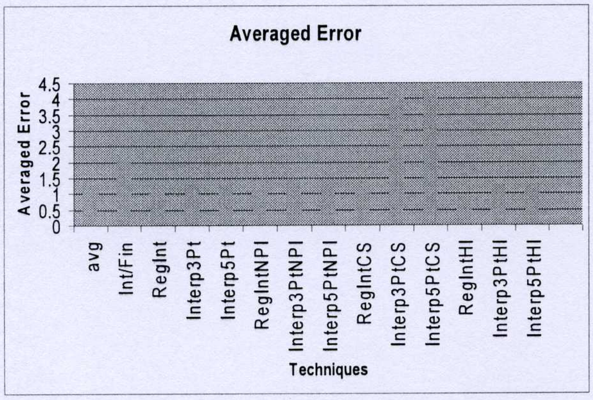


Figure 7. Averaged Error for summarized representations

The best results are achieved using the regular interval interpolation using Hermit Interpolation (0.97 degree celsius) while 5 points linear interpolation (1.24) provides the second to best option (1.3) followed by averaging. Averaging being the easiest provide better estimation when the data has lower standard deviation values, while regular interval and 5 points interpolation being more complex generally provides better estimation with average deviation of less than 1 degree centigrade in considered interval. A sample approximation of sample space by 5 points linear interpolation against the original data is presented in Figure 8 while sample space approximation for another interval using Piecewise Hermite Interpolation is presented in Figure 9. Based on these results we came to the conclusion that the data retrieval by applying piecewise hermite interpolation using data sampled at regular intervals provide better results in terms of time consumption to reply queries and the error caused by approximation. Figure 10 presents the side by side comparison of all these techniques for time consumption and error caused.

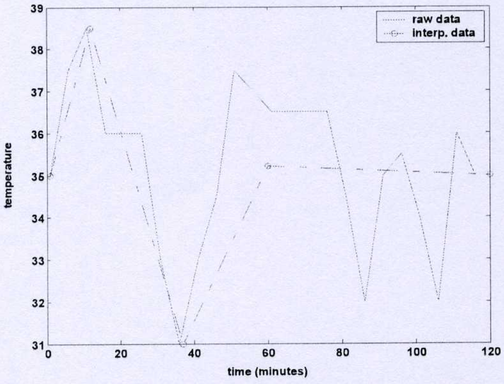


Figure 8. Sample space approximation by 5 points interpolation

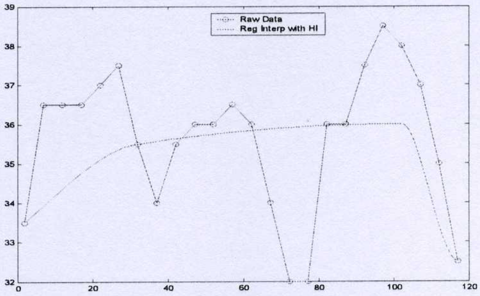


Figure 9. Sample space approximation by regular ineterval hermite interpolation

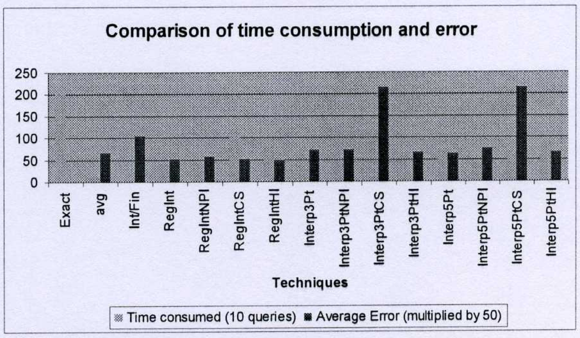


Figure 9. Comparing time consumption and error side by side

5. Issues & Challenges

As we mentioned earlier that ubiquitous computing systems are resource hungry because of their distributed nature, diverse sensors and devices and smart service selection and service delivery. Hence, performance is reasonably important issue for each of its components. We are keeping our summarization system as thin as possible which can work with very limited resources efficiently. There are quite a few mathematical and graph based techniques like curve fitting with limited points to find the missing point in the graph but we are very selective because of inevitable complexity.

Precision and accuracy is another important issue. The tasks that are usually performed using context repository like preference and machine learning, logic reasoning, intention prediction and activity recognition; all of them require data with higher degree of precision. The more we summarize the information, the greater is the loss of precision. To reduce this precision loss we can use more comprehensive techniques but then there is a trade-off between precision and performance. But the important thing is that as the size of data reduces, all the ubiquitous system components that use the context repository tend to perform faster. Security is another significant issue. Since the summarization techniques change the representation of data, hence if the summarization is not done properly, the results produced would be misleading. Hence, summarization components are required to be designed and implemented with great care

6. Related Work

Several existing systems support techniques like feature extraction and generalization [9][10][4] but we want to formally make summarization as part of the ubiquitous system's data management. Our idea is to generate summaries so that later we don't need the raw data any more and can reply to most of the queries with this summarized information with acceptable degree of confidence. In DBMS, data mining [11] and data ware housing [12] use the concept of histogram [13] and multidimensional views of database and work on the aggregate, consolidated data instead of raw data to support the higher level decision making and to identify the hidden patterns in the data. The goal of data mining and OLAP is somewhat similar but we want to transform the raw context to summarized form taking less storage space and provide improved and efficient reasoning and machine learning. Researchers in DBMS have also analyzed the time series data streams for very large databases [14] [15]. Here, they analyze the data coming in continuous streams with time. They have proposed solutions on how to manage, represent and store the time series data streams. This is also highly related to the context summarization. Aggregate data analysis [16] has also been discussed in DBMS for quite sometime which is also helpful for this kind of work.

7. Future Work & Conclusion

For the future work, we are currently trying to summarize the location map of wifi based location awareness system. This map contains the signal strength received at different locations by the device from access points in the surrounding. The size of this data is very huge; just for few floors the number of records grows in multiple of 10,000! We are also optimizing our Query Answering Interface (QAI) so that it can efficiently use multiple summary representations of different strengths to answer queries with different precision requirements accordingly. In the conclusion, we would say that data aggregation and context summarization is an interesting research area which needs to be further explored. The proper application of summarization process does conserve storage space and improves the query processing and data migration in distributed ubiquitous computing environment as supported by our experimental results.

8. References

[1] M. Weiser, The computer for the 21st century. ACM SIGMOBILE 1999 Review

[2] Chen Harry, Tim Finin, and Anupam Joshi: An Intelligent Broker for Context-Aware Systems. In: Ubicomp 2003, Seattle, Washington

[3] Gaia: A Middleware Infrastructure to Enable Active Spaces. Manuel Román et al., In IEEE Pervasive Computing, Oct-Dec 2002

[4] Hung Q. Ngo, Anjum Shehzad, Saad Liaquat, Maria Riaz, Sungyoung Lee: Developing Context-Aware Ubiquitous Computing Systems with a Unified Middleware Framework. EUC 2004: 672-681

[5] Michael J. Franklin, Challenges in Ubiquitous Data Management. . Informatics: 10 Years Back, 10 Years Ahead, LNCS #2000, R. Wilhiem (ed)., Springer-Verlag 2001

[6] Faraz Rasheed, et al, "Context Summarization & Garbage Collecting Context", *UWSI 2005, In the proceedings of ICCSA 2005, Volume III*, Springer Verlag, Singapore, 2005, pp. 1115

[7] Faraz Rasheed, et al, "Towards using data aggregation techniques in ubiquitous computing environment", *PerWare 2006, In the proceedings of PerCom 2006*, IEEE, Italy, 2006

[8] Faraz Rasheed, et al, "Applying Context Summarization Techniques in Pervasive Computing Systems", *SEUS 2006*, IEEE, Korea, 2006

[9] Mike Spreitzer, Marvin Theimer, Providing location information in a ubiquitous computing environment, ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles Dec 1993, Volume 27 Issue 5

[10] Jason I. Hong, James A. Landay, Support for location: An architecture for privacy-sensitive ubiquitous computing, Proceedings of the 2nd international conference on Mobile systems, applications, and services, June 2004

[11] Alex Berson , Stephen J. Smith, Data Warehousing, Data Mining, and OLAP, McGraw-Hill, Inc., New York, NY, 1997

[12] Inmon, W.H., Building the Data Warehouse. John Wiley, 1992

[13] D. Barbara et al., The New Jersey Data Reduction Report, Bulletin of the IEEE Technical Committee on Data Engineering December 1997 Vol. 20

[14] Lin Qiao et al, Data streams and time-series: RHist: adaptive summarization over continuous data streams, Proceedings of the eleventh international conference on Information and knowledge management, Nov 2002

[15] Approximating a Data Stream for Querying and Estimation: Algorithms and Performance Evaluation, Proceedings of the 18th International Conference on Data Engineering (ICDE'02), Feb 2002

[16] Joseph M. Hellerstein, Peter J. Haas, Helen J. Wang: Online Aggregation. SIGMOD Conference 1997: 171-182