Incorporating Global Index with Data Placement Scheme for Multi Channels Mobile Broadcast Environment

Agustinus Borgy Waluyo[1] Bala Srinivasan[1] David Taniar[2] Wenny Rahayu[3]

[1] Caulfield School of Information Technology, Monash University, Australia
{Agustinus.Borgy.Waluyo , Bala.Srinivasan}@infotech.monash.edu.au

[2] Clayton School of Information Technology, Monash University, Australia
David.Taniar@infotech.monash.edu.au

[3] Department of Computer Science and Computer Engineering, Latrobe University, Australia
w.rahayu@latrobe.edu.au

Abstract. A scalable information delivery services like data broadcasting scheme is definitely of great interest especially to deal with the exponential increase of mobile users. In this paper, we present Global Indexing with data placement scheme for multi broadcast channels environment, which aims to minimize query access time, tuning time and power consumption. This scheme considers index and data allocation over separate broadcast channel. Moreover, it concerns with single and multiple data items request. A simulation model is developed to analyze the performance of the proposed scheme. We compare the proposed scheme with conventional scheme. It is found that the proposed scheme provides substantially better performances in every aspect of evaluation.

1 Introduction

In recent years, the use of wireless technology has been growing at an unbelievable rate. Most people are now able to connect to the network using portable size wireless computers powered by batteries (e.g. PDAs) and to conduct their business activities anywhere and anytime. These portable computers communicate with a central stationary server via wireless channel, and this paradigm is known as mobile computing [2,7].

Mobile computing environment is facing inherent limitations particularly power, storage and bandwidth capacity. With power capacity, the life expectancy of a battery (e.g. nickel-cadmium, lithium ion) was estimated to increase time of effective use by only another 15% [8]. Thus, efficient use of energy is definitely one of the main issues. Having considered the limitations of mobile environment, it is very much of benefit to incorporate a scalable and energy efficient mechanism for retrieving database information. Data dissemination (or also called broadcast mechanism) is considered the most appropriate strategy to deliver such services. Wireless broadcasting has been applied for decades in radio and television industry and it is very effective to cope with a massive number of clients. Two factors need to be considered to address the issue of conserving battery power and maintaining short access latency: (i) *Access time*: Elapse time from the time a request is initiated until all data items of interest are received. (ii) *Tuning time*. Amount of time spent by the client to listen for the desired broadcast data item(s). Two modes exist for tuning time; active and doze mode. Active mode is when the client listens into the channel for the desired data item, hence costly to power consumption, while doze mode is when clients simply turned into a power saving mode. With regard to minimising power consumption, a broadcast indexing scheme will be incorporated. Indexing scheme provides accurate information for a client to tune in at the appropriate time for the required data [5]. As a result, mobile clients are able to conserve the energy of their unit by switching to power saving mode and back to active mode when the data is about to be broadcast.

In this paper, we *present* Global Indexing with data placement scheme for multi channels mobile broadcast environment, which is designed to minimise query access time, client's tuning time, and power consumption. This paper is an extension of our previous work in [13]. In the earlier work, we focus on the data items ordering and access time performance without involving any indexing scheme. We developed Global indexing scheme in [12]. However, it only concerns with the index structure without considering data instances. This paper combines both data placement and Global indexing scheme forming a single broadcast system. It should be noted that in this paper, the data and index segment are located in different broadcast channel. Figure 1 illustrates the situation when index segment and data segment are in separate channel. The Figure shows the access and tuning time to retrieve data item # 3, which starts from the time client probes into the beginning of the index channel until the desired data item has been obtained.

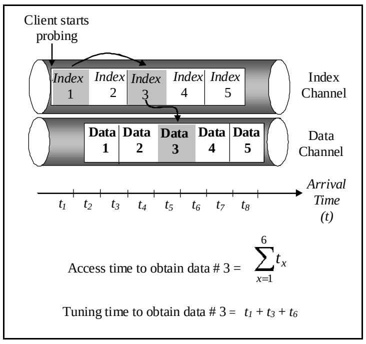


Figure 1. Broadcast mechanism

The rests of the section in this paper are organized as follows. Section 2 contains the related work of the proposed technique. It is then followed by the description of the proposed scheme and its application in section 3. Section 4 describes the simulation model and performance results of our proposed scheme as compared to conventional method. Finally, section 5 concludes the paper.

2 Related Work

Prabhakara et al [5] presented a broadcast ordering scheme, where the hot items or the most frequently accessed data items are broadcast more often than cold items. This basic technique will be used for comparison in the later section.

Broadcast indexing technique is able to minimizing the amount of time client listening into the channel and thus less power consumption. The tree-indexing based on *B*+-tree structure is first introduced in [4]. It is then expanded and modified as what appears in [6]. This technique incorporates a separate index channel to locate the data segment in multi data channel. However, this technique does not concern with broadcast ordering issue. Our proposed technique considers broadcast indexing with data placement scheme into a single broadcast system. The proposed scheme involves single and multiple data items retrieval and allocates the data and index segment in separate broadcast channel. This fact distinguishes our paper from the existing works.

3 Global Index with Data Placement Scheme for Multi Channels Mobile Broadcast Environment: Proposed Method

3.1 Preliminaries

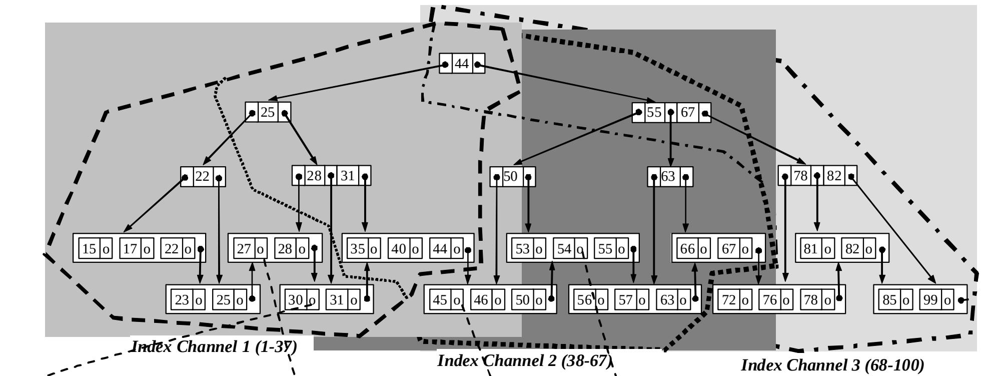
In this paper, we concern with request that returns multiple data items. A number of applications in this category includes a situation when mobile client wants to obtain more than one stock prices information at the same time (i.e. to list the stock price of all car companies under General Motors corp.) To accommodate this type of request, we consider the relationship between one data item and the others. Although, multiple data items retrieval is our primary objective, our proposed scheme also considers single data item retrieval. Thus, the application of this scheme is not limited to either case. Having said so, it is necessary to determine the broadcast order in advance so that the average query access time can be improved.

To find the query access information of mobile clients, we may incorporate either one of the following mechanism. One is to collect the access information from each mobile client at a regular interval, second is to determine the access information from the behaviour of offline or desktop and wireless users with the assumption that the same data items are accessible through the internet. We can assume that offline users and wireless users have a similar access patterns. Some analysis on query patterns and access information has been reported by Microsoft research group in [1]. Once, the access information is received by the server, the statistics will be compiled and the broadcast organisation scheme will be implemented. An example of query patterns is as follows: $Q1 = \{d_1,d_3\}$; $Q2 = \{d_1,d_3,d_4\}$; $Q3 = \{d_5\}$; $Q4 = \{d_6,d_7,d_8,d_9\}$; $Q5 = \{d_2,d_3,d_7\}$.

Having found the query access information, broadcast programs can be classified into two categories: *with replication* and *without replication*. A broadcast program with replication corresponds to the case that the data items appear with different frequencies. Generally, the most popular data item will be broadcast more often than others. On the other hand, a broadcast program is without replication if all data items are broadcast with equal frequencies or uniform frequencies. A broadcast program without replication is also called a flat broadcast program. This paper relates to the later broadcast program and leaves the replication broadcast program for future work. Moreover, our proposed placement scheme is designed under assumption that the queries have equal access frequencies.

3.2 Global Indexing Model

Global Indexing model is designed based on *B*+ tree structure [12]. Global index consists of non-leaf nodes, and leaf node. Leaf node is the bottom most index, where each key point to actual data items.



Data Channel 1			
30	Argentina	MerVal	**903.680**
72	Brazil	Bovespa	**18,796.91**
44	Canada	S&P TSX Composite	**8,191.61**
67	Chile	IPSA	**1,412.79**
53	Mexico	IPC	**9,821.000**
99	United States	S&P 500	**1,098.70**
55	Australia	All Ordinaries	**3,366.800**
78	China	Shanghai Composite	**1,562.688**
63	Hong Kong	Hang Seng	**11,276.86**
66	India	BSE 30	**5,069.87**

Data Channel 2			
25	Indonesia	Jakarta Composite	**722.709**
28	Japan	Nikkei 225	**10,849.63**
17	Malaysia	KLSE Composite	**793.97**
81	New Zealand	NZSE 50	**2,564.730**
85	Pakistan	Karachi 100	**5,454.69**
22	Philippines	PSE Composite	**1,521.61**
23	Singapore	Straits Times	**1,754.96**
27	South Korea	Seoul Composite	**768.46**
31	Sri Lanka	All Share	**1,321.19**
35	Thailand	SET	**609.72**

Data Channel 3			
46	Taiwan	Taiwan Weighted	**5,777.32**
50	Austria	ATX	**1,798.41**
54	Belgium	BEL-20	**2,391.09**
57	Czech Republic	PX50	**750.80**
76	Denmark	KFX	**252.53**
82	France	CAC 40	**3,603.26**
15	Germany	DAX	**3,803.10**
56	Italy	MIBTel	**20,513.000**
40	Netherlands	AEX General	**327.39**
45	Norway	OSE All Share	**327.39**

Figure 2. Global Index Model

When being broadcast, each physical pointer to the neighbouring leaf node as well as actual data item are replaced by a time value, which indicates when the leaf node or data item will be broadcast. Global index scheme is partitioned into a number of channels and each channel has some degree of replication. Each index channel has a different part of the entire index structure, and the overall structure of the entire index is still preserved. As such, Global indexing scheme has the same behaviour as single channel system. Global index is exhibited in Figure 2. Further details on Global Index model can be found in [12]. Query processing in this scheme can be described as follows:

- Mobile client tunes in one of the index channel (i.e. can be of any index channel).
- Mobile client follow the index pointer to the right index key. The pointer may lead to another index channel that contains the relevant index. While waiting for the index to arrive, mobile clients can switch to doze mode.
- Mobile client tunes back on at the index segment that has the right index key, which point to the data channel that contains the desired data item. It indicates a time value of the data to arrive in the data channel.
- Mobile client tunes into the relevant data channel, and switch back to doze mode while waiting for the data item to come.
- Mobile client switches back to active mode just before the desired data item arrives, and retrieves the required items.

3.3 Data Placement Scheme

Let us denote $D = \{d_1, d_2 \ldots d_m\}$, which is a set of data items to be broadcast in the server, and Q as a set of queries $\{Q_1,Q_2,\ldots,Q_n\}$. In this case, we assume the data item has an equal size and the order of the retrieval can be arbitrary, which means if any of the required data by Q_i arrives in the channel, it will be retrieved first. Each query, Q_i, accesses a number of data items j, where $j \subset D$. The broadcast channel is indicated by C, and the length of the broadcast cycle in a channel is given by BC. We denote the broadcast schedule as $S = \{d_x, d_y, \ldots. d_z\}$. Similarly, the broadcast program for each channel is defined by SC.

The mechanism of our proposed scheme is divided into three stages.

- *Firstly*, we calculate the access frequency of each data item given by Q. Subsequently, the data items will be sorted in decreasing order of value of access frequency. The result will be temporarily stored in $ST1$, and incorporated into the final broadcast order S after the second stage is finalized. The algorithm of this stage is given in Figure 4 (a).

- *Secondly*, the relationship between one data item and the other will be analyzed. The access frequency of each data item in relation with other data item will be assessed from given Q. This stage is required to allocate the related data items close to each other. Thus, the number of calculation required for this purpose will be $\frac{m \times (m-1)}{2}$, m is the total number of data items in D. Figure 4 illustrates this stage. It shows that each data item to be broadcast will be analyzed for its relationship with other data item based on Q. The calculation is done in sequence basis. i in Figure 3 reads the data item number. This process iterates until each data item has been counted. A single process includes two ways relationship of the data. Figure 3 also depicts an example when there is 5 number of data items to be broadcast.

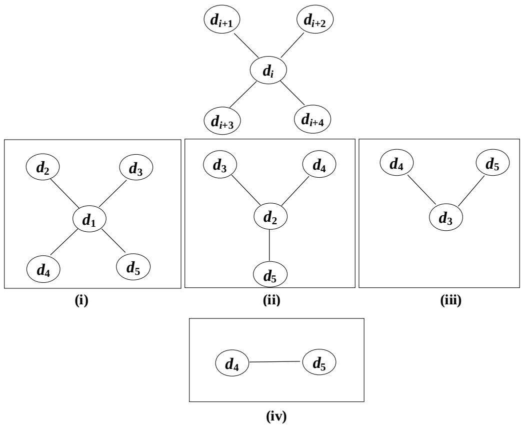


Figure 3. The architecture of the proposed method

Each data item will be counted for its relationship with the next sequence of the data item from given Q. The double point arrows indicate two ways relationship of the two data. In this case, $D = \{d_1, d_2, d_3, d_4, d_5\}$, we count the frequency of the two data items, which appear at the same time in a query. Started from d_1, which is counted towards the relation with the next sequences of the data item that is d_2, d_3, d_4, d_5. This process continues sequentially until m-1 data item. After all data item has been analysed, they are sorted in non-increasing order based on the access frequency with other data item, and stored in *ST2*. Each allocation will involve two data items. If there is a duplication of data item within *ST2*, then data item with the highest *ST2* value are kept and the rests are removed. Figure 4 (b) shows the algorithm of the second stage.

- The *final stage* is to combine *ST1* and *ST2* forming a single broadcast order *S*. We start with allocating the data item in *ST2* into *S* also in non-increasing order. This process continues until just one access frequency left to be processed. Subsequently, we take the result from the *ST1*, analyse it against *S*, the data item in *ST1* that has existed in *S* will be discarded. The left over data item is put into *S* with non-increasing order as well. The broadcast order is then allocated over multiple channels. Assuming the required number of data items for each channel is known, which is indicated by *BC*. Once the *BC* is reached, the subsequent data items will be broadcast in another channel. We can employ our previous work in [11] to determine the optimum number of data items in broadcast channel. The algorithm of this mechanism is given in Figure 4 (c).

Having known the broadcast order enable us to determine the sequence of data items that minimize the average access time. The next issue to consider is to specify the broadcast program. The broadcast program indicates the schedule of the first item in *S* at the broadcast cycle, and the sequential items follow the order that has been generated. To achieve the most effective broadcast program, one must note the statistical patterns of users start listening or probing into the channel. In this paper, we assume the pattern follows the behavior of Gaussian distribution. We allocate the broadcast order *SC* and *SIC* into the program, where the first item in each *SC* and *SIC* is placed at point 2/3 in the broadcast program. The point is chosen to minimize the chance of clients missing the data of interest and have to wait for the subsequent cycle. With this way, majority of requests are served within a single broadcast cycle. Although this means about 50% of requests have to wait for a short time period but the overall performance will be better since only a fraction of requests need to wait for the next cycle. The broadcast program can then be generated at a regular interval.

Algorithm 1. Broadcast Ordering Algorithm – Stage 1

Input: An array of data items (*D*), queries (Q_i) with their related data items (j_i), and the total number of data items to be broadcast (*m*).
Output: An array of broadcast program (*ST1*)
Procedure:

```
Initialise i = 1, T = 0
Do while i ≤ m
        For n = 1 to Q_max
            Count the frequency of d_i in Q_n
            Store and cumulate the result in temporary variable T
            If Q_max then
                ST1_(i) = T
                Clear T
            End if
                Next n
        increment i
Loop
Check the largest value in ST1
Ordered the value ST1 such that ST1 (1) ≥ ST1 (2) ≥…≥ ST1 (m)
End Procedure
```

(a) First Stage Algorithm

Algorithm 2. Broadcast Ordering Algorithm – Stage 2

Input: An array of data items (*D*), queries (Q_i) with their related data items (j_i), and the total number of data items (*m*).
Output: Array of broadcast program (*ST2*).
Procedure:

```
Initialise i = 1, T = 0
Do while i < m
  For k = (i + 1) to m
    For n = 1 to Q_max
        Count the frequency of d_i retrieved at the same time
        with d_k in Q_n
        Store and cumulate the result in temporary variable T
        If Q_max then
           TDarray (d_i, d_k) = T
           Clear T
        End if
     Next n
  Next k
  increment i
Loop
Check the largest value in TDarray
Ordered the value in TDarray from the largest to the smallest
  value such that TDarray (1) ≥ TDarray (2) ≥…≥ TDarray (max)
The coordinate value for each TDarray determine the data item
Place the coordinate value for each TDarray to ST2 with the non-
  increasing order. Check for any duplicate data item within
  TDarray. Data item can only appear one for each cycle. Thus, in
  case there is any duplication, data item with the highest TDarray
  value are kept and the rests are removed.
End Procedure
```

(b) Second Stage Algorithm

Algorithm 3. Broadcast Ordering Algorithm – Stage 3

Input: Two arrays of Broadcast schedule (*ST1 and ST2*), optimum length of broadcast cycle in a channel (*BC*)
Output: An array of Final Broadcast program (*S*), two dimensional array of broadcast program for each channel (*SC*)
Procedure:

```
For i = 1 to ST2_(max)
      Move ST2_(i) to S
Next i
For i = 1 to ST1_(max)
      Check if ST1_(i) exists in S
      If ST1_(i) ⊂ S then
            skip
      Else
            Move ST1_(i) to S_(max+1)
      End if
Next i
Let channel number = x
Initialise x =1, i =1, j =1, k =1
Do While i ≤ S_(max)
      If k ≤ BC
         SC(x, j) = S(i)
         increment j
      Else
         if k > BC then
           k = 1 ; j = 1
           x = x +1
         End if
      End if
      increment i
      increment k
Loop
End Procedure
```

(c) Third Stage Algorithm

Figure 4. Broadcast Ordering algorithms

4 Performance Evaluation

In this section, we analyze the performance of our proposed method in two cases. In the *first* case, we compare the proposed method against conventional ordering scheme. In this conventional scheme, the data items are ordered and broadcasted based on the access frequency without considering the relationship with other data items. The first case concerns with the access time performance. The *second* case relates to determining the tuning time of Global index and non-global index. In this last case, we also calculate the average power consumption of mobile clients for listening to the channel.

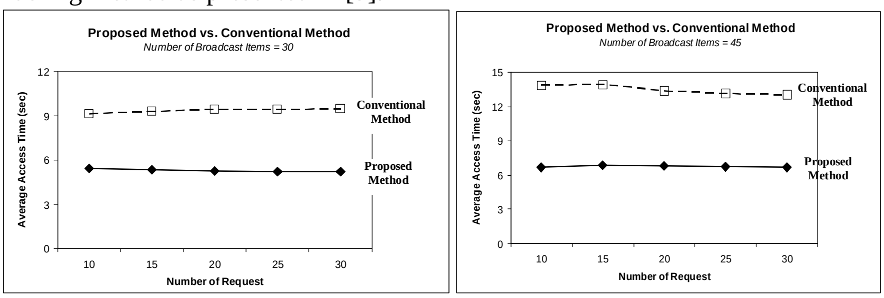
Table1. Parameters of Concern

Parameters	Value
Size of each data Item	2KB
Bandwidth	64Kbps
Query Patterns/Profiles	10
Number of Dependent Items in Query	1-4
Number of Broadcast Channel	3
Number of Broadcast Data Items	30 and 45
Number of Request	30
Global Index	
Node Pointer Size	5 bytes
Data Pointer Size	5 bytes
Indexed Attribute Size	4 bytes
Index Arrival Rate	4 index nodes per sec

In this evaluation, we use the same data and indexing scenario as illustrated in Figure 2. The index structure is broadcasted following a breadth-first order. To analyse the schemes, we develop a simulation model. The simulation is carried out using two software packages namely Visual Basic 6.0 and *Planimate* or animated planning platforms [10]. The prototype of our system has been reported in [14]. The simulation environment is set to apply exponential distribution for data and index inter-arrival rate given an average value. In the query profile, we consider request that return one and up to four numbers of items. The parameters of concern are given in Table 1. The channel bandwidth is set to 64Kbps which follows the EDGE standard.

4.1 Access Time

Case 1: This case compares the proposed method with a conventional data ordering method as presented in [9].



(a) Proposed Method vs. Conventional Method: 30 broadcast items (b) Proposed Method vs. Conventional Method: 45 broadcast items

Figure 5. Proposed Method vs. Conventional Method

The conventional ordering scheme relates to the basic ordering scheme that allocates the data items in the channel according to the access frequency in a non-decreasing order and the entire index structure is broadcast within an index channel. Two analyses are also considered, which involve 30 and 45 number of broadcast items. In Figure 5, we can see clearly that the proposed method provides a better access than the conventional method with about one and half times lower access time. A similar trend occurs when the number of broadcast items increased (refer to Figure 5 (b). It is shown, that the increase of broadcast items severely affects the conventional method. In the contrary, the access time of our proposed scheme with Global Index rises just a few seconds or can be considered as minor increase. Consequently, the gap between our proposed method and conventional method is getting larger.

4.2 Tuning Time

Case 2: This case relates to comparing the tuning time of our Global index with non-global index scheme. Non-global index refers as a conventional index broadcasting method, which utilizes a single broadcast channel to broadcast the index structure. Tuning time defines the amount of time clients must listen to the channel to retrieve the desired data instances. The analysis involves 30 numbers of broadcast items. As exhibited in Figure 6 (a), the tuning time of mobile clients accessing our Global index is far less than the non-global index scheme. More importantly, the result also indicates that clients spend much less power consumptions during query operation with Global indexing scheme.

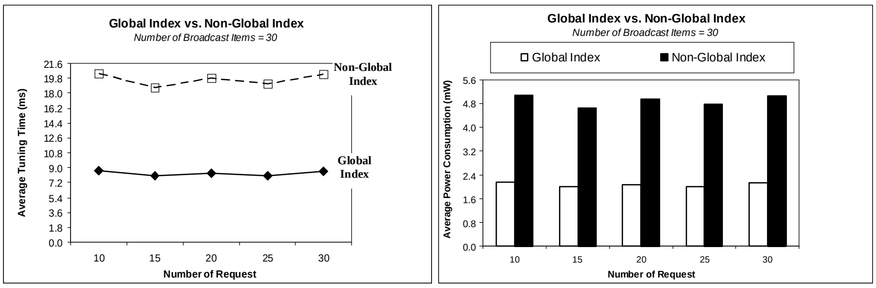


Figure 6. Tuning time and Power Consumption: Global vs. Non-Global Index

According to Imielinski et al [3] a device with Hobbit chip (AT&T) requires about 250mW power consumption during active mode, and 50μW during power saving mode. Thus, we can see that the duration of clients' listening and not listening to the channel has affected the power consumption. We measure the power consumption of mobile clients based on the duration of active mode and power saving mode that has been obtained from the simulations. The calculation is determined from the following formula:

Power Consumption = (250 x *Time during active mode*) + (0.05 x *Time during power saving mode).* As depicted in Figure 6 (b), the power consumption of clients accessing the Global index is substantially less than non-global index. With little power consumptions, client can reserve battery power more efficiently, which is very much desirable considering the lack of resources in mobile devices.

5 Conclusions and Future Work

In this paper, we have presented Global indexing with data placement scheme for multi broadcast channel environment. We have developed a simulation model to analyze the performance of our proposed scheme. We compare the proposed scheme with conventional scheme. The analysis includes access time, tuning time and power consumption. It is found that the proposed scheme provides substantially better performances in every aspect of evaluation.

For future work, we will incorporate a certain degree of data replication in the broadcast program. We will investigate the most effective degree of the replication as well as the structure and scheduling of the broadcast program.

References

1. Adya A., Bahl P., Qiu L., "Analyzing the Browse Patterns of Mobile Clients" In *Proc. of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pp.189-194, 2001.
2. Barbara D., "Mobile Computing and Databases-A Survey", *IEEE TKDE*, **11**(1):108-117, 1999.
3. Imielinski T., Viswanathan S. and Badrinath B. R., "Data on Air: Organisation and Access", *IEEE TKDE*, **9**(3): 353-371, 1997.
4. Imielinski T., Viswanathan S. and Badrinath B. R., "Energy Efficient Indexing on Air", In *Proc. of the ACM Sigmod Conference***,** pp.25-36, 1994.
5. Lee K. C. K., Leong H. V. and Si A., "Semantic Data Access in an Asymmetric Mobile Environment", In *Proc .of the 3rd Mobile Data Management***,** pp.94-101, 2002.
6. Leong H. V. and Si A., "Data Broadcasting Strategies Over Multiple Unreliable Wireless Channels", In *Proc. of the 4th CIKM***,** pp.96-104, 1995.
7. Myers B.A. and Beigl M., "Handheld Computing", *IEEE Computer Magazine*, **36**(9):27-29, 2003.
8. Paulson L.D.,"Will Fuel Cells Replace Batteries in Mobile Devices?" *IEEE Computer Magazine*, **36**(11):10-12, 2003.
9. Prabhakara K., Hua K.A, and Jiang N., "Multi-Level Multi-Channel Air Cache Designs for Broadcasting in a Mobile Environment", In *Proc. of the ICDE*, pp. 167-176, 2000.
10. Seeley D. et al, *Planimate^{tm}-Animated Planning Platforms*, InterDynamics Pty Ltd, 1997.
11. Waluyo, A.B., Srinivasan B., Taniar D., "Optimal Broadcast Channel for Data Dissemination in Mobile Database Environment", In *Proc. of the 5th APPT*, pp. 655-664, 2003a.
12. Waluyo, A.B., Srinivasan B., Taniar D., "Global Index for Multi Channels Data Dissemination in Mobile Databases", In *Proc.of the 18th ISCIS*, pp.210-217, 2003b.
13. Waluyo, A.B., Srinivasan, B., and Taniar, D., "Allocation of Data Items for Multi Channel Data Broadcasting in a Mobile Computing Environment ", In *Proc. of the 1st EUC*, pp. 409 – 418, 2004.
14. Waluyo, A.B., Goh, G., Srinivasan, B., and Taniar, D., "On-Building Data Broadcast System in a Wireless Environment, *International Journal of Business Data Communications and Networking*, **1**(4), 14-36, 2005.