Multiversion Spatio-temporal Telemetric Data Warehouse

Marcin Gorawski

Silesian University of Technology, Institute of Computer Science,
Akademicka 16, 44-100 Gliwice, Poland
Marcin.Gorawski@polsl.pl

Abstract. One of the crucial problems characterizing current data warehouses is the implicit assumption of dimension invariance with respect to the time dimension. This assumption inhibits the proper treatment of changes in dimension data. Meanwhile, we can give examples indicating that it is necessary to take into consideration the modifications of dimension data - ignoring such changes leads to incorrect analysis which then results in wrong decisions. This article describes the implemented temporal telemetric data warehouse system, which provides the user with the ability to query about the time interval embracing many structure versions. The system also informs the user about modifications which occurred in separated structure versions.

Keywords: data warehouse, spatial data warehouse, temporal data warehouse, structures versioning, temporal operations.

1 Introduction

Regulation of energy sector in EU created new market – media (electricity, gas, heat, and water) recipient market. The main problem is automated reading of hundreds of thousands recipients meters and critically fast analysis of terabyte sets of meters' data. Condition for achieving this goal is usage of an Integrated Meter Reading (IMR) and a Spatio-Temporal Telemetric Data Warehouse (STDW(t)) – Monitoring and Media Distribution Decision Support System (2MDSS) (fig. 1) [6, 9].

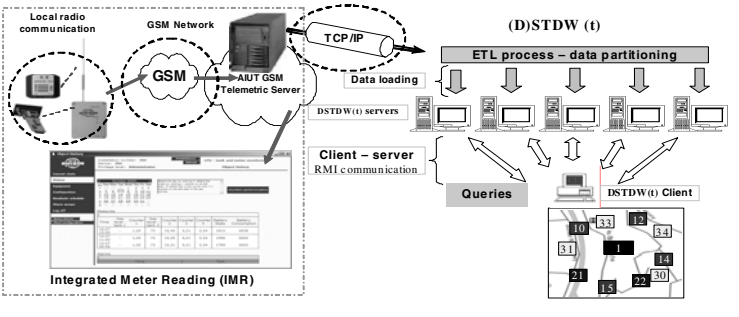

Fig. 1. The Monitoring and Media Distribution Decision Support System (2MDSS)

J. Grundspenkis et al. (Eds.): ADBIS 2009 Workshops, LNCS 5968, pp. 63–70, 2010.

These meters measure the usage of water (W), gas (G), and electricity (E). IMR system sends data from media meters to a database system via a cellular telephony network (GPRS). IMR is a transactional system, which services four types of meters: electrical energy, water, gas, and heat. Data stored in the 2MDSS telemetric server are in raw state and need to be adequately formatted. STDW(t) system gathers data from telemetric servers in extraction process via a network with TCP/IP protocol. In distributed STDW(t) (DSDW(t)) of 2MDSS, during extraction process additional partitioning is performed followed by data loading to system nodes [7, 8].

2 The 2MDSS Modeling

The 2MDSS modeling in a spatio-temporal scenarios is focused on classified objects {k*} and georegions {R} dimensions variability. 2MDSS objects can be classified as follows.

A. Single dimension k* objects (dimensions) – single data, data packages, and data streams, from e-receptor measurements can be presented as tuples with attributes (e.g. size, measurement frequency and period, place, name and type).
B. Spatial k* objects (spatial data types) – a spatial object can be: point (e.g. meter, counter), line or region in a 2/3D space (georegion). Spatial objects include characteristics of object geometry, metrics, relations, density, and objects decomposition.
C. Spatio-temporal k* objects (spatio-temporal data types). These objects are described with spatial data expanded with time (spatio-temporal data) and modeled with transaction or validity time. Spatio-temporal object can change its position and/or its form. The standard models have 3-5 dimensions.
D. Stream k* objects (stream data types). These objects are described with stream data, that create data sets of single spatio-temporal data and spatio-temporal data packs and data streams.

The 2MDSS works in spatio-temporal scenarios, certain Aggregates Space can include any topological georegion that has a dynamic characteristic (like in [15]).

The georegion services by 2MDSS can be denoted as the R set of two-dimensional regions (2D) with the smallest granulation of static aggregations $\{R_i\ (1<=i<=N) \mid R_i \in 2D, R_i \in R\}$ (e.g. constant number of segments if a road network or e-receptors, concentrators) or time changeable $\{R_i(t) \mid R_i(t+1) \neq R_i(t), R_i(t)/t>0\}$ (e.g. various number of e-receptors – regions of a cellular network antennas, dependable on weather conditions, capacity, etc.).

The time axis is denoted with discrete timestamps t_c $\{t_c \mid t_c \in T, 1<=c<=|T|\}$, where T is a set of ordered timestamps, describing $R_i(t)$ such that for $t_{c+1}-t_c=\Delta t>0$, the period of regions structure changes fulfills $R_i(\Delta t)>>0$ condition.

For example each counter k $\{k \mid k \in S,\ S \in R_i(t)\}$ generates one measurement – value ms_k, e.g. each $\Delta t = 15$ (minutes) in a $R_i(t)$ region, that changes (various number of counters) every 4 hours ($R_i(\Delta t) >= 240$). Each e-receptos is connected with the

ms_k value (e.g. measurements), then the $f_{agg}(1)$ function calculates number of e-receptors and the $f_{agg}(ms_k)$ calculates aggregates for qualified e-receptors.

Each of $R_i(t)$ regions is connected with the measurements aggregates set $R_i(t).f_{agg}(ms_k)$, obtained from k e-receptors localized in $R_i(t)$, which values are actualized cyclically. The $R_i(t).f_{agg}(ms_k)$ is calculated with distributive aggregation function f_{agg} (count, sum, max, min).

The 2MDSS is modeled as one of 6 characteristic STDW(t)(k*R) classes:

- STDW(t)(k*A) – constant $R_i(t)$ regions and stationary spatial objects.
- STDW(t)(k*B) – characteristic, constant $R_i(t)$ regions and stationary spatial objects – subclasses: STDW(t)(k*B1) cluster characteristic, STDW(t)(k*B2) bucket characteristic.
- STDW(t)(k*C) – slowly changing $R_i(t)$ regions and stationary spatial objects.
- STDW(t)(k*D) – quickly changing $R_i(t)$ regions and stationary spatial objects.
- STDW(t)(k*E) – constant $R_i(t)$ regions and a known set of mobile objects, which position in a t moment cannot be precisely calculated.
- STDW(t)(k*F) – constant $R_i(t)$ regions and known mobile objects trajectories and known mobile objects trajectories.

3 Multiversion STDW(t)

Previous research on *Multiversion Data Warehouse* (MVDW), concerns mainly multiversioning of classical data warehouses (DW) [2, 3, 5, 16, 18].

The majority of this researches present incremental refreshing of traditional DW forced with source data change. We can distinguish actualization of dimensions schema and a fact table. Through *dimensions schema actualization* we mean change (actualization) of a dimension structure and their instances (objects). The most important research on multiversioning of classical DW are [1, 4, 19].

The schema DW is represented as a graph with defined algebra that allows the creation of new versions [5]. The DW evolves in a direction of: a) new versions and b) upgrading schema of every previous version. The idea of history in DW is a set of versions, which includes a special type of cross-version querying. In [2, 18] the Multiversion Data Warehouse - MVDW is proposed, and is defined with a set of its versions. Each version consists of a schema version and an instance version. The two kinds of versions can be distinguished: (1) real, considering real changes in data sources and (2) alternative, considering changes for various simulation scenarios. Real versions create a linear order, while alternative versions create tree structures with data common for different versions. The concept of multiversion data warehouse is fully presented in [19].

The evolution of STDW in the 2MDSS system is presented. In a mutliversion form, through time ordered set of a data schema versions and instance versions. Our research on multiversion STDW(t) are connected with defining changes in $R_i(t)$ regions, that influence:

- Data validity time while keeping coherence and correctness of a dimensions structure change.
- Necessity of those changes (minima and complete set of the $R_i(t)$ structure).

4 The Temporal Aspects in Data Warehouses

The traditional data warehouses are ideally suited for analysis of facts changing in time [4]. It is expected that the OLAP data warehouse allows reliable data analysis even in a long time period [17]. In this context, the quite interesting fact is that data warehouses cannot efficiently manage modifications of dimensions structure (e.g. introducing the next unit or branch in a company) – even when usually in that warehouse time is represented as one of dimensions. There are many examples that show how important is the possibility to change and update dimensions in time – ignoring this fact leads to obtaining incorrect analysis results [11]. Problems like, e.g. comparison of data from different time periods or designating trends need adequate manner of managing changes in a dimensions structure. Otherwise, we have to accept the possibility of obtaining incorrect results and in effect making incorrect decisions [4,11].

The correct and consistent inclusion of dimension structures modification in time, needs describing dimensions with *time stamps,* to obtain *validity time* of a data warehouse dimensions [4].

The validity time is a time, in which "the fact is true in modeled reality" [10]. The time stamp is denoted as $[T_s, T_e]$ and means that the dimension is valid in this period where: T_s marks the period beginning, T_e marks the period end, and $T_e \geq T_s$.

When we present all time stamps of all modifications on a time axis, then the range between two time stamps on this axis marks the structure version. Through this we mean a data warehouse view that is valid in a certain time period. In one structure, the dimension structure is consistent and constant. Each modification operation for the dimension forces creation of a new structure version – actual structure loses its validity. This kind of modification operations is called *temporal operations.* In case of designing the new system such operation can be, e.g. connecting a new node in a certain point of time, connecting new meter, updating a range of operation for a node or a meter failure. We also have to introduce the idea of a *time unit* also called the *chronon.* The chronon is a time range with certain determined and undivided minimal length [10]. When setting the chronon's length simultaneously we set the precision of a data representation in a data warehouse. The length of chronon is set with consideration of the data warehouse character (data character and its usage). In case of designing the system the most important is the best and precise representation of time, because measurements data can be send to central even every couple of minutes. That is why the chronon length is set for one second.

Fig.2 presents the time axis with several data structure versions in a data warehouse. The data warehouse stores data from a certain region in which we have certain number of nodes along with meters. The user can perform analysis and queries starting from the beginning point T0. In this point, there are three nodes A, B, and C, and each of them services several meters (fig.1). Until T1 there was only one structure version, with the validity time [T0, NOW] (NOW denotes present moment). In T1, B2 meter malfunctioned, and this is the temporal operation that creates instability in structures consistency and forced the creation of a new structure version. So now there are two structure versions: SV1 [T0, T1] and SV2[T1, NOW]. Up to the "most actual" version there were three more temporal operations – in points T2, T3 and T4. So the most actual version has the SV5 identifier and its validity time is [T4, NOW]. In this time we have two nodes B and C along with meters.

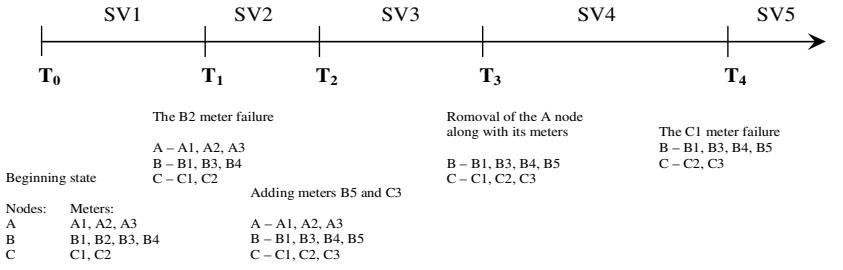

Fig. 2. An example of SDW structures versioning

5 Aggregate Tree

The designing of the aggregated structure in data warehouse systems that process spatial and temporal data is a key problem [14]. The queries performed in a spatio-temporal data warehouse system concern aggregated data so preliminary aggregation in indices greatly reduces response time. In the presented problem we implemented the so called aggregate tree, which is based on [13]. The aggregate tree is an index created by data warehouse systems in the operation memory. This approach considerably decreases query response time (in comparison to performing query directly in a base system). Before creating the aggregate tree certain factors should be defined, e.g. tree height and a size of a net created from so called minimal bounding rectangle – MBR. The MBR is the smallest, indivisible fragment of space (map), for which we aggregate data stored in the database. If there is a need to collect aggregated results from an area smaller than MBR, then we have to increase the MBRs grid density. Along with the increase of the tree height and the MBRs number (which is consistent with the increase of a MBRs grid density overlaid on the map) the tree construction time also increases and the ability to perform more adequate queries emerges. The user will have the ability to modify the tree parameters, so he can choose them empirically, for to compromise query precision and a tree creation time.

6 Data Modeling

The schema of STDW(t) based on the so-called cascaded star [9, 12] is shown in fig.3. For best understanding the schema includes only tables along with connecting relations. On the highest abstract level the schema consists of a main fact table – INSTALLATION and five dimensions: NODES, METERS, WEATHER, MEASURES, and MAP. These dimensions store data about nodes and meters along with their attributes, weather conditions, measures from the meters, and the terrain map. The separate sub dimension tables of the main dimensions, store attributes connected with time, spatial localization and other attributes – such approach makes the schema clearer and easier to upgrade.

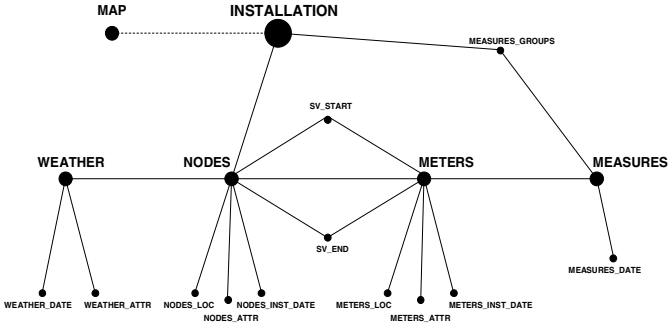

Fig. 3. A schema of a measurement oriented versioned cascaded star for STDW(t)

Table 1. The example of SV_START i SV_END tables

SV_START		*SV_END*	
ID_SV_START	**SV_START_DATE**	**ID_SV_END**	**SV_END_DATE**
1	01/01/2001 00:00:00	1	03/01/2001 15:38:53
2	03/01/2001 15:38:54	2	05/01/2001 10:38:04
3	05/01/2001 10:38:05	3	10/01/2001 13:04:09
4	10/01/2001 13:04:10	4	16/01/2001 01:44:52
5	16/01/2001 01:44:53	5	25/01/2001 19:17:50
6	25/01/2001 19:17:51	6	31/12/2199 00:00:00

To fulfill earlier assumptions, the implemented system should be not only spatial (stores information about objects spatial localization) but also temporal (it has the ability to incorporate modifications of data in time). The temporal character of our model is assured with tables: SV_START and SV_END. These tables store data about structures importance ranges. The tables are used both by NODES and METERS dimensions so it is possible to set importance ranges for nodes and meters. This fact transforms the cascaded star model into fact constellation model in which some tables can be used by several dimensions, just like in the presented model.

Table 1 presents an example of the six structure versions in tables SV_START and SV_END. The chronon between next structures equals one second. The version is created using dates pointed with the same identifiers. For example number 2 structure version is valid through 3 January 2001 15:38:54 to 5 January 2001, 10:38:04. The last structure (no. 6) is valid from 25 January 2001, 19:17:51 up to current moment (NOW). To mark this date we use date 31 December 2199, 00:00:00. The NODES dimension along with its subtables store information about nodes, their attributes, localization and installation dates. Below there is more information about columns:

- NODES_LOC.X, NODES_LOC.Y store information about node localization.
- NODES_ATTR.R stores information about the node area radius. Meters connected with this node are placed in this circular area where the middle point is marked with (NODES_LOC.X, NODES_LOC.Y) and the radius equals NODES_ATTR.R. This results from a fact, that the data transmission is through radio connection, so the node range is appointed by radio range (radius).

Table 2. Values in the END_REASON field in the NODES table

Value	*Description*
D	Node deletion
R	Change of node radius
RT	Change of node type and radius
T	Change of node type
–	No operation – node exists until present moment

Table 3. Example of joint tables – NODES and NODES_ATTR

id_node	*id_sv_start*	*id_sv_end*	*end_reason*	*type*	*r*
2	1	3	D	WG	26
4	1	4	RT	WG	35
4	5	6	–	WGE	40
5	1	6	–	WE	26

- NODES_ATTR.TYPE stores the information about what type of meters are serviced by this node (W, G, E, WG, WE, GE, WGE).
- NODES.END_REASON informs why a certain node was not included in the next version, or what nodes attributes were modified (tab. 2).

From table 3 we can see that, e.g., node no. 2 with the 26 radius and the WG type is still valid in structures 1 – 3 (1 January 2001 00:00:00 to 10 January 2001, 13:04:09, see Tab.1). The validity was lost when the node was deleted (END_REASON = 'D'). Node no. 4 with unchanged attributes (radius=35, type=WG) is still valid in structures 1-4. However, between structures 4/5 the radius is changed (35 → 40) along with the node type (WG → WGE), which is reflected by value 'RT' of END_REASON. After those changes the node is still valid until the present moment (until the end of the structure no. 6 and this is the most actual structure which keeps validity to the point marked as NOW). That is why the value of END_REASON for this node equals '–'. Node no. 5 keeps its value in time for all structures (1-6) until this moment, that is why the value of END_REASON equals '–' (just like in case of node no. 4).

7 Summary

The goal of our work was to design and implement of the multiversion spatio-temporal telemetric data warehouse. We created the working system that uses mechanisms and conception of the aggregate tree and structures versioning, which is based on the cascaded star model. This project can be upgraded and expanded in multiple manners. For example it can be extended into the cascaded star schema with materialized aggregate trees or geographical distribution in Distributed Spatial Telemetric Data Warehouse DSDW(t) [9].

References

1. Baril, X., Bellahsène, Z.: Designing and Managing an XML Warehouse. In: Akmal, B., Chaudhri, A., Zicari, R., Awais Rashid, A. (eds.) XML Data Management. Native XML and XML-Enabled Database Systems, pp. 455–474. Addison-Wesley Professional, Reading (2003)
2. Bebel, B., Eder, J., Koncilia, C., Morzy, T., Wrembel, R.: Creation and Management of Versions in Multiversion Data Warehouse. In: ACM SAC, Nicosia, pp. 717–723 (2004)
3. Bellhase, Z.: Schema Evolution in Data Warehouses. Knowledge and Information Systems 4, 283–304 (2002)
4. Eder, J., Koncilia, C.: Changes of Dimension Data in Temporal Data Warehouses. In: Kambayashi, Y., Winiwarter, W., Arikawa, M. (eds.) DaWaK 2001. LNCS, vol. 2114, p. 284. Springer, Heidelberg (2001)
5. Golfarelli, M., Lechtenbörger, J., Rizzi, S., Vossen, G.: Schema Versioning in Data Warehouses. In: Wang, S., Tanaka, K., Zhou, S., Ling, T.-W., Guan, J., Yang, D.-q., Grandi, F., Mangina, E.E., Song, I.-Y., Mayr, H.C. (eds.) ER Workshops 2004. LNCS, vol. 3289, pp. 415–428. Springer, Heidelberg (2004)
6. Gorawski, M., Dyga, A.: Indexing of Spatio-Temporal Telemetric Data Based on Adaptive Multi-Dimensional Bucket Index. Fundamenta Informaticae 90(1-2) (2009)
7. Gorawski, M., Gorawski, M.J.: Multiversion spatio-temporal data warehouse. In: Grundspenkis, J., Morzy, T., Vossen, G. (eds.) ADBIS 2009. LNCS, vol. 5739, pp. 291–297. Springer, Heidelberg (2009)
8. Gorawski, M., Malczok, R.: On efficient storing and processing of long aggregate lists. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2005. LNCS, vol. 3589, pp. 190–199. Springer, Heidelberg (2005)
9. Gorawski, M.: Extended Cascaded Star Schema and ECOLAP Operations for Spatial Data Warehouse. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 251–259. Springer, Heidelberg (2009)
10. Jensen, C.S., Dyreson, C.E.: The Consensus Glossary of Temporal Database Concepts. In: Temporal Databases, Dagstuhl (1997)
11. Mendelzon, A.O., Vaisman, A.A.: Temporal Queries in OLAP. In: 26th International Conference on Very Large Data Bases, VLDB 2000, Egypt, pp. 242–253 (2000)
12. Nabil, A., Vijayalakshmi, A., Yesha, Y., Yu, S.: Efficient Storage and Management of Environmental Information. In: IEEE Symposium on Mass Storage Systems (2002)
13. Papadias, D., Kalnis, P., Zhang, J., Tao, Y.: Efficient OLAP Operaions in Spatial Data Warehouses. In: Advances in Spatial and Temporal Databases, 7th International Symposium, SSTD, CA (2001)
14. Papadias, D., Tao, Y., Kalnis, P., Zhang, J.: Indexing Spatio-Temporal Data Warehouses. In: 18th International Conference on Data Engineering, ICDE, San Jose (2002)
15. Tao, Y., Papadias, D.: Historical spatio-temporal aggregation. ACM Transactions on Information, TOIS 23, 61–102 (2005)
16. Vaisman, A.A., Mendelzon, A.O., Ruaro, W., Cymerman, S.G.: Supporting Dimension Updates in an OLAP Server. In: Pidduck, A.B., Mylopoulos, J., Woo, C.C., Ozsu, M.T. (eds.) CAiSE 2002. LNCS, vol. 2348, pp. 67–82. Springer, Heidelberg (2002)
17. Wrembel, R., Koncilia, C.: Data Warehouses and OLAP: Concepts, Architectures and Solutions. In: Advances in Data Warehousing and Mining, p. 332. Idea Group, Inc., USA (2007)
18. Wrembel, R., Morzy, T.: Managing and Querying Versions of Multiversion Data Warehouse. In: Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 1121–1124. Springer, Heidelberg (2006)
19. Wrembel, R.: Management of Schema anad Data Evolution in Multiversion data Warehouse. Dissertation 411, Pub. House of Poznan Univ. of Technology, p. 338 (2007)