ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.11

Framework Based Materialized View Evolution: A survey

Anjana Gosain[1], Sangeeta Sabharwal[2], Rolly Gupta[3]
Professor[1, 2], Research Scholar[3]
Department of University School of Information Technology[1], Department of COE[2, 3]
GGS Indraprastha University, Delhi, India[1]
Netaji Subhas Institute of Technology, Delhi University, Delhi, India [2, 3]

Abstract:
Data Warehouse evolution is a critical problem in present scenario due to perpetual transactions (data changes) and change in data warehouse structure (schema changes) emerging out of persistent evolving users' requirements. Handling all type of changes properly is an urgent process. It forms the core component for modern decision support systems. Therefore DW has to be refreshed intermittently as indicated by different type of evolution in information sources. We refer to the problem of evolving an appropriate set of views as the materialized view evolution problem. Many different materialized view evolution methods have been proposed in the literature to address this issue. The present paper provides a survey of materialized view evolution methods. This paper aims at studying the materialized view evolution in relational databases and data warehouses as well as in a distributed setting. It defines an evolutionary approach for highlighting the materialized view evolution problem by identifying the main dimensions that are the basis in the classification of materialized view evolution methods. Firstly, we proposed three main dimensions namely; (i) Framework, (ii) Architecture and (iii) Model/Design Model. Secondly, we classified the materialized view evolution methods according to several dimensions, namely; (i) View Evolution, (ii) Basic View Maintenance, (iii) Incremental VM, (iv) Self Maintainable Maintenance, (v) Not self Maintainable Maintenance, (vi) View selection, (vii) View Synchronization, (viii) View Adaptation. Based on this classification, this study reviews most of the framework based materialized view evolution methods by identifying respective potentials and limits.

General Terms: Information Systems, Database Management, Database Administration.

Keywords: View Maintenance, Materialized views, Integrity maintenance.

I. INTRODUCTION

A data warehouse and views present in warehouse must evolve whenever there is any modification or update in the requirements or base relations. In fact, data warehouse evolution process never ceases. The datasets that are generated in the response to the queries raised by the users are called as views. Views represent functions derived from the base relations to support viewing of snapshots of stored data by the users according to their requirements. These derived functions are recomputed every time the view is called upon. Re-computing and selection of views becomes impossible for each and every query especially; when the data warehouse is very large or the view is quite complex or query execution rate is high. Thus, we accumulate some pre-calculated results (or views) in our central repository (i.e. data warehouse) in order to provide faster access to data and enhance the query performance. This technique is referred as materialization of views. Materialized views act as a data cache that gather information from distributed databases and support faster and reliable availability of already computed intermediate result sets (i.e. responses to queries). Evolution in data warehouse may be generated by change in schema, changes in software and the change in data warehouse requirement [1]. Data warehouse evolution may be classified into three different approaches namely schema evolution, schema versioning and materialized view evolution. Schema evolution of data warehousing consist of various levels updates that is dimension updates, structural updates, instances updates, facts updates, attributes updates, cube updates, hierarchy updates, measure updates, quality updates and constraints updates. Dimension updates reflect static aspect of data warehouse evolution and structural updates reflects dynamic aspect. In versioning approach, focuses on keeping trace of changes by keeping different versions of a given DW. Materialized view evolution approach focuses on choosing materialized views in the design process of data warehouses or maintaining a materialized view in response to data changes or to data sources changes and sometimes to monitor the DW quality under schema evolution.
The materialized view evolution issue has been investigated in several contexts: query optimization, warehouse design, data placement in a distributed setting, web databases, etc. Many diverse solutions to the materialized view evolution problem have been proposed and analyzed through surveys [Dhote et. al. 2009, Halevy 2001, Labrinidis et. al 2009].

The survey [Dhote et. al. 2009] concentrates on methods of finding a rewriting of a query using a set of materialized views. The study presented in [Halevy 2001] focuses on the state of the art in materialization for web databases. A critical analysis of methodologies for selecting materialized views in data warehousing is provided in [Labrinidis et. al 2009]. However, none of the above mentioned surveys provides a classification of materialized view evolution approaches in order to identify their advantages and disadvantages. Our survey fills this gap.
This paper aims at studying the materialized view evolution in relational databases and data warehouses as well as in a distributed setting. First, we define a dimension for highlighting the materialized view evolution problem. Thus, we present a classification of materialized view evolution methods based on the main materialized view evolution dimensions that we have identified. This study also reviews

existing framework based materialized view evolution methods by identifying respective potentials and limits. The rest of the paper is organized as follows: Section 2 identifies the main materialized view evolution dimensions along which materialized view evolution methods can be classified. Section 3 presents a critical literature survey of existing framework based materialized view evolution methods. Section 4, presents a comparative study of the various research works explored. Section 5 contains the conclusion and discusses open issues.

II. STATE OF THE ART

The goal of the materialized view evolution is to simplify the design, implementation, maintenance and management of data warehousing approaches. Therefore, in this section we classified the materialized view evolution into following dimensions -- Framework, Architecture and Model/Design Model. Based on the methods involved in evolution of materialized views in a data warehousing dimensions, they can be categorized further. So, the taxonomy used for further classification is illustrated in Figure 1 as – View Evolution, Basic View Maintenance, Incremental VM, Self Maintainable Maintenance, Not self Maintainable Maintenance, View selection, View Synchronization, View Adaptation. We have discussed the literatures being reviewed in context of data warehousing dimensions and methods. In subsequent section, we have described the data warehousing dimensions and the methods designed to handle the evolution of a materialized view in response to the modifications in data sources.

III. LITERATURE SURVEY

In the literature, different authors have proposed different techniques to handle materialized view evolution at different dimensions. The evolution approach focuses View Evolution, Basic View Maintenance, Incremental VM, Self Maintainable Maintenance, Not self Maintainable Maintenance, View selection, View Synchronization, and View Adaptation. These section discuss about these framework based materialized view evolution methods in detail.

3.1 FRAMEWORK

In [1], authors proposed the Posse framework for optimizing incremental view maintenance at data warehouses. They showed how for a particular method of consistent probing it is possible to have the power of SQL view queries with multiset semantics, and at the same time have a spectrum of concurrency. They also showed, how optimization of the probing process can be used to select various degrees of concurrency for the desired trades of concurrency against processing cost and message size

In [2], the authors proposed a data warehousing framework, which adopts an efficient incremental lightweight view maintenance technique; motivated by the fact that different views in a Data Warehouse (DW) can have different freshness requirements. This fact has been used for enhancing the view maintenance performance in huge DWs by installing the source updates when the freshness constraints are violated and only for the DW views that have the violated constraints.

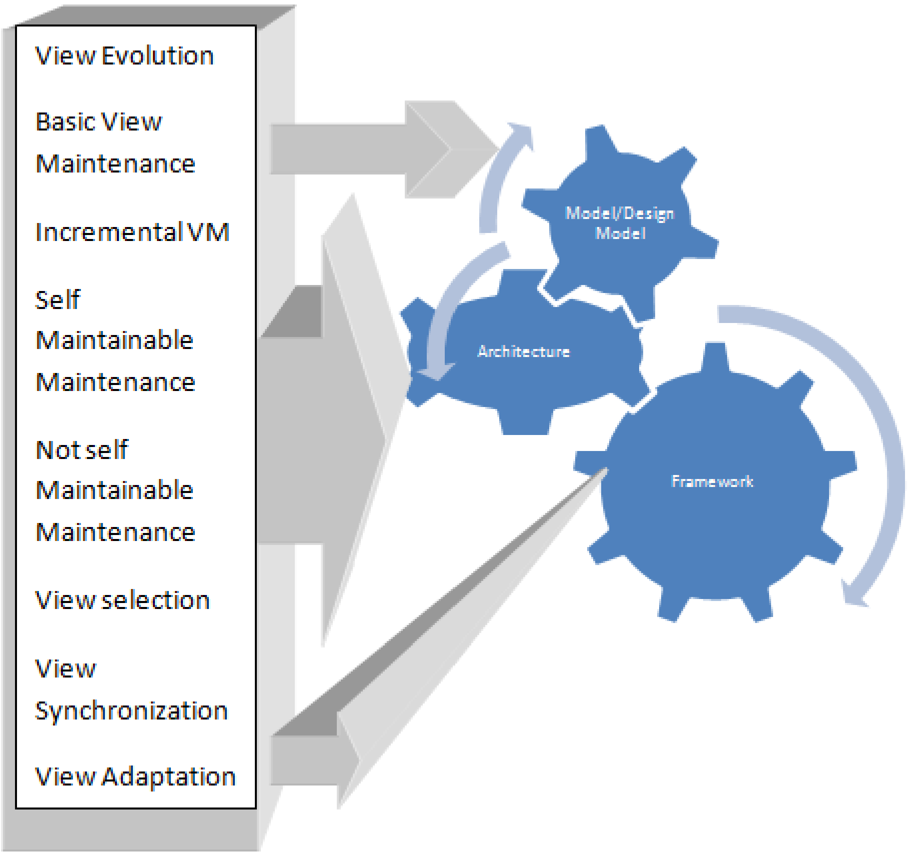

Figure.1. Classification of Data Warehousing Solutions

In [3] paper authors presented a system, named DynaMat that dynamically materializes information at mul tiple levels of granularity in order to match the demand (workload) and also takes into account the maintenance restrictions for the warehouse. The proposed system unifies the view selection and the view maintenance problems. They compared DynaMat against a system that is given all queries in advance and the pre-computed optimal static The experiments showed that DynaMat's dynamic view selection outperforms the optimal static view selection. In [4] authors described a solution for warehouse refreshment in a federation of data warehouses, by defining a special kind of materialized view and providing a prototype for materializing this kind of views. In [5] papers, authors briefly present a theoretical framework for the DW design problem, which concerns the selection of a set of views. They also formalized the DW design problem as a state space search problem by taking into account multiuser optimization over the maintenance queries and the use of auxiliary views for reducing the view maintenance cost. Finally, presented incremental algorithms and heuristics for pruning the search space. In [6] paper, authors proposed a framework to support the physical design of a data warehouse; it is based on a logical model, called the view synthesis graph. A level of overlapping is then defined to characterize the graph, which allows the search space. This framework also integrates some quality factors that can be used either to drive the exploration of the search space, or to reduce the search space, or both collectively. In [7], authors provided a solution that guarantees concurrent view definition evolution and view extent maintenance of a DW defined over distributed ISs by introducing a framework called SDCC (Schema change and Data update Concurrency Control) system. This framework integrates existing algorithms designed to address view maintenance sub problems, such as view extent maintenance after IS data updates, view definition evolution after IS schema changes, and view extent adaptation after view definition changes, into one system by providing protocols that enable them to correctly co-exist and collaborate. They also proposed a local correction strategy, called Local Compensation (LC) for conflicting concurrent IS changes. In [8] paper a data warehouse framework is presented supporting a data warehouse evolution. The framework handles not only changes in data sources, but also direct changes in a data warehouse schema by supporting versions in the development environment as well as in reports in the user environment. In [9] authors settled a control system named, COBIT (Control Objectives for Information and related Technology) for data warehouse projects. This consisted of a number of control objectives plus a methodology to accomplish the audit process. In [10] paper, presented the evolution and the objectives of CI. Authors have proposed a framework of a decision support system based on web mining techniques in order to enhance capabilities of organization’s competitive intelligence. In [11] paper authors outlined a general methodological framework for data warehouse design, based on Dimensional Fact Model (DFM). Firstly, the existing information system and collected the user requirements is analyzed, then conceptual design is carried out semi-automatically starting from the operational database scheme. Thereafter, a workload is then characterized, to be used as the input of the logical and physical design phases to generate output; the final scheme for the data warehouse. In [12], paper makes two contributions: Firstly, enriches the meta data about DW architectures by explicit enterprise models. Secondly, developing many very different mathematical techniques for measuring or optimizing certain aspects of DW quality. Authors have adapted the Goal-Question-Metric approach from software quality management to a meta data management environment in order to link these special techniques to a generic conceptual framework of DW quality. The approach was implemented on top of the Concept Base repository system and has undergone some validation by applying in support of specific quality-oriented methods, tools, and application projects in data warehousing. In [13], paper they applied and extended the NFR Framework to bridge the gap between NFRs and implementation. They implemented the extended NFR Framework, as a modeling language including a soft goal interdependency graph validation tool with a language workbench. Then further extended the NFR Framework with a concept of measurable NFRs that enables to empirically verify the realization of defined NFRs. In [14] authors, proposed an enterprise modeling framework for the deployment of data warehouses. This framework provides the information roadmap coordinating source data and different data warehouses across the business enterprise. They also introduced a solution to address data warehousing issues at the enterprise level while avoiding the pitfalls of creating enterprise data warehouses and universal data marts. It further proposes a change of paradigm from point solutions focus to a methodology driven by enterprise requirements to meet the challenges of the new economy. The proposed framework emphasizes the separation of the conceptual construct from the physical and operational constructs of an enterprise. It points out the differences and dependencies of analytic and operational processes and how data warehouses and operational data stores respectively support their information requirements. This paper will demonstrate how the enterprise modeling framework for data warehousing can produce business benefits. In [15] paper, they briefly presented a framework, called QUADRIS, relevant for adopting a quality improvement strategy on one or many dimensions and considering the collateral effects on the other interdependent quality dimensions. In [16] ,authors addressed the problem of the static design of a DW and its evolution. They used expression AND/OR dags to represent alternative ways of evaluating multiple queries and views, and sub expression sharing; to allow the representation of complex queries including grouping/aggregation queries. The mapping scheme can be used for searching space for the general view selection problem, and by exploring algorithms. This approach can also be applied to particular DW design cases. In [17] authors presented FIESTA, a framework for the evolution of conceptual multidimensional (MD) schemas. As examples for Multidimensional Information Systems (MDIS), they focused on OLAP systems. In [18] authors proposed an Evolvable View Environment (EVE) which addresses of evolving views under IS capabilities changes, coined as view synchronization problem. The key principles of EVE include a user specified preference model for view evolution (Evolvable- SQL (E-SQL)) and a Model for Information Source Descriptions (MISD). In [19], they proposed an approach that does not materialize all the views, but rather to materialize certain “shared views” from which the query results can be generated. They addressed some issues related to determining set of shared views to be materialized in order to achieve the best performance and low maintenance. They have provided an algorithm for achieving goal of ‘shared views’. In paper [20], authors proposed a novel framework called DyDa that handles both source data updates and schema changes. They identified three types of maintenance anomalies, caused by source data updates, data-preserving schema changes, or non-data-preserving schema changes. They also proposed a compensation algorithm to solve the first two types of anomalies. The third type of anomaly is caused by the violation of dependencies between maintenance processes. Then, they have proposed dependency detection and correction algorithms to identify and resolve the violations. That is, DyDa extends prior maintenance solutions to solve all types of view maintenance anomalies. In paper [21], authors proposed a framework to determine if the existing materialized views will be affected by the requirement changes, and how they are affected. The situation was dealt by means of algorithms; to efficiently get the new materialized views by analyzing the relationships among queries using MVPP. In paper [22], authors outlined a solution to challenging new problem of how to adapt views in such evolving environments. They also outlined the Evolvable View Environment (EVE) approach that proposes a framework for solving the view synchronization problem, along with decisions concerning the key design issues surrounding EVE. The main contributions are: 1) an E-SQL view definition language with which the view definer can direct the view evolution process, 2) a model for information source description which allows a large class of ISs to participate in system dynamically, 3) formally defined what constitutes a legal view rewriting, 4) developed replacement strategies for affected view components which are designed to meet the preferences expressed by E-SQL, 5) proved the correctness of the replacement strategies, and 6) provided a set of view synchronization algorithms based on those strategies. In [23], authors provided a solution that guarantees concurrent view definition evolution and view extent maintenance of a DW defined over distributed ISs by introducing a framework called SDCC (Schema change and Data update Concurrency Control) system. This framework integrates existing algorithms designed to address view maintenance sub problems, such as view extent maintenance after IS data updates, view definition evolution after IS schema changes, and view extent adaptation after view definition changes, into one system by providing protocols that enable them to correctly co-exist and collaborate. They also proposed a local correction strategy, called Local Compensation (LC) for conflicting concurrent IS changes. In [24] paper, authors dealt with the problem of evolution in the context of databases. Firstly, they presented a coherent, graph-based framework for capturing the effect of potential changes in the database software of an Information System. Then, they described a generic annotation policy for database evolution and proposed a feasible and powerful extension to the SQL language specifically tailored for the management of evolution. Finally, demonstrated the efficiency and feasibility of their approach. In [25] paper, authors presented a framework, for the view selection problem, which intends to achieve the best combination of low query processing cost, low view maintenance cost and good query response. The framework optimizes the maintenance, storage and query processing cost by selecting the most cost effective views to materialize. Thus, providing an efficient data warehousing system. In [26], authors presented a theoretical framework called DWEVOLVE for supporting data warehouse evolution. The framework thus enhances the functionality of previously designed framework by taking into account the requirements specified by the users. In [27], authors proposed a framework to allow designers in generation of schema based on a set of queries. A Hybrid Model Driven Development Framework for the Multidimensional Modeling of Data Warehouses In [28], authors discussed a framework based on the Model Driven Architecture (MDA) for the development of a hybrid MD model at the conceptual level and for automatic derivation of its logical representation. In paper [29], they proposed two interpretations of the event rules which provide a common framework for classifying and specifying deductive database updating problems such as view updating, materialized view maintenance, integrity constraints checking, integrity constraints maintenance, repairing inconsistent databases, integrity constraints satisfiability or condition monitoring. In [30], authors presented a framework for data integration, based on a special class of mediators, called Squirrel integration mediators. They can support the traditional virtual and materialized approaches, and also hybrids of them the paper also presents formal notions of consistency and "freshness" for integrated views defined over multiple autonomous source databases. In [31], authors presented the data warehouse framework that is able to track evolution process and adapt data warehouse schemata and data extraction, transformation and loading (ETL) processes. In [32], they presented incremental view maintenance algorithms for a data warehouse derived from multiple distributed autonomous data sources. The paper also proposes algorithms to perform the compensation locally by using the information already available at the data warehouse. The SWEEP algorithm ensures complete consistency of the view at the data warehouse in the presence of concurrent updates. In [33] authors presented presents a systematic approach to matching categories of query language interfaces with the requirements of certain user types. The approach is based on a trend model of query language development on the dimensions of functional capabilities and usability. In [34], author's objective is to define a process for maintenance management and to classify maintenance engineering techniques within that process. They presented a generic model for In [35], authors have addressed the problem of generating information system designs from requirements specifications, with the presentation of a framework for representing requirements and a mapping assistant, IRIS, that facilitates the design generation process. In [36], authors have presented a model of P2P-based collaborative view maintenance, which can not only exploit the potential merits of P2P and CSCW, but also overcome some major problems. Furthermore, a collaborative maintenance algorithm is also proposed. In [37], authors reviewed notes is to define more precisely the emerging concept of e-maintenance and then to propose and discuss a conceptual e-maintenance framework based on a Zachman framework, as this can facilitate a widespread understanding of e-maintenance and provide useful guidance for supporting e-maintenance deployment through services, processes, organisation and infrastructure. In [38], they extended the framework of Grin and Libkin to include maintaining views with aggregation in the general case, where aggregation can include group by attributes and a view can include several aggregate operators. In [39], authors provide the foundation for defining a framework for materialized views over heterogeneous data sources in an event stream processing environment. Specifically, they explored LINQ as a materialized view definition language for defining views over both relational and XML structured data sources. In [40], authors proposed a cost-driven view maintenance framework which generates optimized maintenance plans tuned to a given environmental settings. This can significantly improve view maintenance performance especially in a distributed environment. In [41], authors presented a mapping framework for information system development and their usage in generating system designs. They presented the description of a prototype implementation—IRIS—of aspects of the proposed mapping framework and illustrates its features through a sample session. In [42], they described a framework for supporting ad-hoc declarative update requests in an OODB, while maintaining database consistency and atomicity of update requests. This is mainly based on the emulation of classic update methods in an OODB by a controlled, active and user-transparent interaction between a predefined set of elementary updates and a set of integrity methods designed to maintain database consistency upon violations of integrity constraints. In [43], authors reviews the role of performance measurement systems (PMS) in maintenance, with particular reference to developing a new PMS using the quality function deployment (QFD) technique and a framework is developed to embrace key facets. In [44], authors investigated variety of the CASE tools developed under an impact of MDA growing progress and discussed the differences between model driven architecture and model driven development in the context of automatic capabilities for software development. In [45], authors investigated two maintenance strategies, i.e. extended batching and view graph transformation, for maintaining general join views where join conditions may exist between any pairs of data sources possibly with cycles. They proposed a cost-driven view maintenance framework which generates optimized maintenance plans tuned to the environmental settings. In [46], the purpose of author is to define a process for maintenance management and to classify maintenance engineering techniques within that process. They presented a generic model proposed for maintenance management. In [47], authors proposed a framework based on the Model Driven Architecture (MDA) for the development of a hybrid MD model at the conceptual level and for the automatic derivation of its logical representation. In [48], author introduces the concept of End-To-End Data Warehousing, which focuses on establishing a framework of concepts centered on the Data Warehouse architecture. It also examines key human and technological factors that influence the success rate of the Data Warehouse. Finally, paper explains how tools and technology can be optimally deployed in Data Warehouse architecture. In [49], authors worked with the modeling of the dynamic parts of the data warehouse and proposed a met model of data warehouse operational processes capable of modeling complex activities, their interrelationships, and the relationship of activities with data sources and execution details. They exploited the framework to revert the widespread belief that data warehouses can be treated as collections of materialized views.

IV. COMPARATIVE STUDY

We have analyzed various research works (Data Warehousing solutions) on several parameters and presented their comparative results (including sub-classification) in the table below:

Table.1. Comparison of Various Research Works

S. No	Authors	Techniques /Category Adapted	Issues Addressed/ Changes Handled	Process support/ perspective	Method's Activities / Goals	Addressed attributes	Applicable framework stage	Advantages	Disadvantages	Types of Queries/ Operation	Tool Support/ Implementation
1.	Kevin O'Gorman, Divyakant Agrawal, and Amr El Abbadi	Incremental VM	Compensating for interfering updates	POSSE framework	Concurrent updation, multiset semantics, optimization of query plans	Concurrency , consistency	VM	Complete concurrency of probing process	Address investigation of query plan dynamically and explore optimization criteria	Relational Model	SQL
2.	Yousry Taha, Arsany S. Sawiros, Noha Adly	Incremental VM	Source updates + DW views & consistency	Data Warehousing Framework	Per-view constraint based deferred Maintenance for distributed objects	Modularity, Scalability, Interoperability	VM	Distributed object VM strategy, Plug/Play technique	Not to be used on mobile machine, No crash recovery measure	Relational Model	----
3.	Yannis Kotidis, Nick Roussopoulos	View Maintenance	Dynamically materialize information at multiple levels of granularity + update views	DynaMat Framework	Multidimensional Range Fragments (MFF)	Time and space efficiency	VS VM	Complete self tunable system, Optimal static view	No specific pattern in workload	Relational Model	Experimental Analysis
4.	Aristides Triantafillakis, Panagiotis Kanellis, Drakouli	View Maintenance	Warehouse refreshment in federation of DW	Agent-based framework	Triggers on source table, Hyper-view approach	Information Quality, Adopting system	VM	Complete refreshment process	Seek Data cleansing, merging and customiz	Relational Model	SQL Server 2000

	s Martakos								ation		
5.	Dimitri Theodoratos, Timos Sellis	Incremental VM	Designing issues in DW	DW design framework	Multiquery Optimization, Auxiliary views approach	Cost effective optimization	Design/VM	Minimized total query evaluation and cost	Dynamic DW design not addressed	Relational Model	Experimental Analysis
6.	Mokrane Bouzeghoub, Zoubida Kedad	View Maintenance	Search space for selection of views materialization	Physical design framework	View synthesis graph	Better performance	Design/VM	Ordering of search space,Quality factors driven exploration / reduction of search space	Need to address quality factors related to physical design	Hierarchial Model	---
7.	Xin Zhang , Elke A. Rundensteiner	View Maintenance	Concurrent view evolution and maintenance over distributed sets	SDCC framework	Local compensation strategy	Concurrency control	VM	Semi-concurrency level of consistency, addresses VM problems	More concurrency level can be achieved	Relational Model	Experimental Analysis
8.	Darja Solodovnikova	View Maintenance	Handle changes in data sources/ DW schema	DW Evolution framework	Versioning approach	Concurrency control	Design/VM	Addressed evolution problems	Transform adaptation framework to evolution framework	Relational Model	---
9.	José A. Rodero, José A.Toval, Mario G. Piattini	Design	Controlling objectives, auditing and risk assessment	COBIT framework	COBIT related technology	Effective, efficient, confidentiality, Integrity, Availability, Reliability,	DSS	Refinement of proposed control objective, Proposal of set of metrics, Tool development	---	Relational and OO Model	JAVA, JDBC, SQL, ODBC
10.	I. Anica-Popa, G. Cucui	View Evolution	Improve CI process	DSS framework based on web mining	Web minig, CI, BI	Efficient	DSS	Competitive Advantageous	Implementation required	---	---
11.	Matte0 Golfarelli, Stefano Rizzi	Design	Conceptual design and defining workload	DFM based framework	DFM	Complete and consistent design	Design/VM	Conceptual design, semi-automatically workload characterization	Design algorithm for logical and physical design	Relational Model	SQL
12.	Matthias Jarke, M. A. Eusfeld	Incremental V Maint	Enrichment of metadata, mathemati	Generic conceptual	GQM approach	Quality Design, Data and Software	DSS	Incremental model formation, Open	Validation against case	Relational Model	ConceptBase

	C. Quix, P. Vassilidis	enhance	cal techniques for measurement and optimization	framework				quality management environment	studies Development of techniques		
13.	Anton Yrjönen and Janne Merilinna	Design	Extension with measurable NFR	NFR+ framework	Modelling language including a softgoal interdependency graph	Better Performance	Design/VM	Serve as connecting point and guide to designers, Refinement/ Traceability of requirement, Feasible production process	---	Relational Model	Meta Edit+ + language
14.	Joseph O. Chan	Design	Information roadmap coordinating source data & other warehouses	Enterprise modeling framework	Methodology driven enterprise requirements, conceptual contructs	---	Design/VM	Enhances operational and analytical capabilities, Real Time management, Adaptive enterprise wide information technology architecture	---	Relational Model	Case study
15.	J. Akoka, L. Berti-Équille, O. Boucelma, M. Bouzeghoub, I. Comyn-Wattiau, M. Cosquer V. Goasdoué-Thion, Z. Kedad, S. Nugier, V. Peralta, S. Sisaid-Cherfi	Design	Adopt Quality improvement strategy	QUADRIS	Indentify interdependencies between Quality dimensions: QoD, QoM	Data Quality	Design/VM	Quality-aware information system, Cost optimal strategies for data Quality prevention & improvement	More Study Required	Relational Model	Case Study
16.	Dimitri Theodoratos, Mokrane Bouzegh	View selection	Minimization of cost function or constraints of	VS Frame work	AND/OR dags, mapping techniques	Consistency	Design/VM	Search space for VS, DW design goals	Experimental Implementation	Relational Model	---

	oub		different types, Design goal problems					achieved, Static/ dynamic considerati on, broader class of queries			
17.	Markus Blaschka	View Evolu tion	Methodolo gy for evolution of multidime nsional schema	FIEST A	Schema evolution algebra, specialize d graphical notation	Consisten cy	Desi gn/V M	Conceptual schema evolution operations, Graphical representat ion of multidimen sional schema	Specific conceptu al part, Increase perform ance combini ng physical design	Relatio nal Model	E/R – Graph appro ach
18.	Anisoara Nica, Elke A. Rundenst einer	VM	Evolve view under IS capabilitie s changes	EVE framew ork, MISD	Designing view specificati on language, Meta knowledg e base	Consisten cy	VM	Approach for view synchroniz ation, solution to capability change operators	MKB evolutio n required Non-Cost effective model	Relatio nal Model	E-SQL
19.	Jian Yang, Kamalak ar Karlapale m, Qing Li	VM	Determine set of shared views to be materialize d	Frame work based on shared views	Shared view, MVPP	Better performa nce, low maintena nce	VM	Common intermediat e result can be shared among multiple global queries, MVPP technique	Experim ental Impleme ntation	Relatio nal Model	Case study
20.	Songting Chen, Xin Zhang, Elke A. Rundenst einer	VM	Handling of source data updates & schema changes in distributed environme nt	DYNA framew ork	Dependen cy detection & correction , rescheduli ng maintenan ce order	Concurre nt, Data preservin g	VM	Handles maintenanc e anomalies caused by source data updates, data preserving schema changes or non-data-preserving schema changes	---	Relatio nal Model	Exper iment al Analy sis
21.	Chuan Zhang, Jian Yang	View Evolu tion	Determine how existing materialize d will be affected by requireme nt changes	Materia lized view evoluti on design framew ork	MVPP	Efficient	Desi gn/V M	Addressing -designing algorithm and rules for DW schema evolution problem, Reduced space and recomputat ion	Develop ing analytica l model	Relatio nal Model	Exam ple base study
22.	Amy J.	View	Adopt	EVE	Replacem	Better	Desi	E-SQL	Impleme	Relatio	E-

	Lee, Anisoara Nica, Elke A. Rundensteiner,	Synchronization	views in evolving environment	framework	ent strategies for view components, view rewriting/ algorithm	performance, synchronization	gn/VM	language, solved view synchronization	nt model for capturing quality and cost by view synchronization	nal Model	SQL, ORACLE, JAVA, JDBC, MS ACCESS
23.	Xin Zhang, Elke A. Rundensteiner	VM	Concurrent view evolution & view extent maintenance over distributed IS	SDCC framework	Algorithm /protocol, Local compensation	Better Performance	VM	Addressed VS, VSy, VA problems, Analyzed cost and traffic factors	More factors needs to be analyzed	Relational Model	Example based study
24.	George Papastefanatos, Panos Vassiliadis, Alkis Simitsis, Konstantinos Aggistalis, Fotini Pechlivani, Yannis Vassiliou	VE	To deal with problem of database evolution	Graph based framework	Generic annotation policy for database evolution, sql extension	Efficient, feasible, coherent	Design/M	SQL Language extension, Framework for evolution change	Need to identify patterns of evolution sequences	Relational Model	Case study
25.	B. Ashadevi, Dr. P. Navaneetham	VS	To select set of views that minimizes total cost associated with materialized views	VS Frame work (Optimized version)	Optimization	Low cost & maintenance, Better query response	VS	Efficient DW system	Further factor analysis required	Relational Model	Experimental Analysis
26.	Garima Thakur, Anjana Gosain	Design	Designed framework	DWEVOLVE	---	———	Design/M	Enchanced framework by incorporating user requirements	Implementation required	Relational Model	---
27.	Resmi Nair, Campbell Wilson, Bala Srinivasan	Design	Generate schema based on set of queries	Query driven design framework	Multidimensionality, optimization approach	Better query response	Design/M	Conceptual representation of queries	Require further optimization and implementation	Relational Model	---
28.	Jose-Norberto Maz´on, Juan Trujillo	Design	Develop hybrid model at conceptual level, automatic derivation of its logical	MDA model	Requirement driven approach, Data driven approach	Less complexity	Design/M	Proposed a hybrid model	Require implementation, Not included heterogeneous dimension	Relational Model	Example based study

			representation						hierarchies		
29.	Ernest Teniente, Toni Urpí	VM	View updating, Materialized VM, Integrity/Inconsistency problems or condition monitoring	Frame work for classifying & specifying deductive databases	Interpretation of event rules i.e. upwards & downward problems	Integrity	VM	Provided common framework	Require Implementation	Relational Model	---
30.	Richard Hull, Gang Zhou	VM	Data integration, Integrated views	Frame work based on squirrel integration mediators VDP	Material all auxiliary data or leave some/all of them (VDP)	consistency	VM	Support materialized, virtual & hybrid relation, optimized support for integrated views	Require Implementation	Relational Model	---
31.	Solodovnikova D.	Design	Multiversion DW Evolution	Formal model	ETL	---	D/M	Tracks evolution process	Require further work	---	---
32.	D. Agrawal, A. El Abbadi, A. Singh, T. Yurek	IVM	Efficient VM at multiple distributed autonomous data sources	Frame work for analyzing VM	Algorithms	---	IVM	Consistency of view at DW concurrent updates	Need more algorithms	Relational Model	Experimental Analysis
33.	Matthias Jarke, Yannis Vassiliou	Design	Match categories of query language interfaces with required certain user types	Structured framework DQL	Model of query language development	---	D/VM	Classification scheme for query language, Criterion hierarchy for query evaluation, classification scheme & recommendation	Need more exploration	Relational Model	Experimental Analysis
34.	A. Crespo Márquez, P. Moreu de León, J.F. Gómez Fernández, C. Parra Márquez, V. González	VM	Classify maintenance engineering techniques	Generic model for maintenance management framework	Integration of other models	Maintenance management	VM	Classification & use of different maintenance engg. tools	Further study	Relational Model	---
35.	Zhongmei Shu, Yayao Zuo, Yong Tang	Design	Generating IS designs from requirement specificati	IRIS framework	Mapping process	Accuracy, security	D/VM	Proposed mapping framework	Tool Development needed	Relational Model	Prototype Implementation Telos

			on								
36.	Yannis Vassiliou, Manolis Marakakis, Panagiotis Katalagarianos, Lawrence Chung, Michalis Mertikas, John Mylopoulos	VM	Overcome the problem of overload & crash at end of DW	P2P based collaborative model	Exploits merits of P2P & CSCW	Better Performance	VM	Proposed strategy shows better performance both on message no. and total execution time	---	Relational Model	Experimental Analysis
37.	E. Levrat, B. Iung, A. Crespo Marquez	VM	E-maintenance	Conceptual e-maintenance framework	zachman framework	Clarity	VM	Better understanding & deployment of e-maintenance	Implementation Required	Relational Model	---
38.	Dallan Quass	IVM	Maintaining view with aggregation	Frame work view maintenance	Aggregation operators	Better Performance	IVM	Enhanced maintenance expression to guarantee strong minimality	Explore efficient algorithm	Relational Model	Theoretical Analysis
39.	Mahesh B. Chaudhari, Suzanne W. Dietrich	IVM	Maintaining view over heterogeneous data sources	Materialized view framework over heterogeneous data sources	Prototype using commercial off-shelf components	Better Performance	IVM	Discussed research challenges for IVM	To be Explored more	Relational Model	Explore LINQ XML
40.	Bin Liu and Elke A. Rundensteiner	VM	Maintain & optimize general join views	Cost-driven VM framework	Maintenance strategies	Optimality	VM	Better maintenance over distributed systems	Explore distributed query optimization	Relational Model	Experimental Analysis
41.	Lawrence Chung, Panagiotis Katalagarianos, Manolis Marakakis, Michalis Mertikas, John Mylopoulos, Yannis Vassiliou	Design	Requirement-Design framework	Mapping framework	Method based ondependency & NFR	NFR achievement	D/VM	Prototype - IRIS	Further work required	Relational Model	---
42.	Anton P.	VM	Support	Frame	Based on	Database	VM	User can	Need to	Relatio	Exam

	Karadimce Susan D. Urban		ad-hoc declarative update requests	work for declarative updates & constraint maintenance	emulation of classic update methods	consistency, Integrity, User transparency		freely pose declarative adhoc updates without jeopardizing db consistency	Validate tool	nal Model	ple based study
43.	K.Y. Kutucuoglu, J. Hamali, Z. Irani, J.M. Sharp	VM	Maintenance operations	PMS model	QFD	---	VM	A practical implication based framework	Implementation required	---	---
44.	Nikiforova, O.	VM	Reduce complexity	MDA framework	CASE tools	Complexity	VM	Proposed the framework briefly	Implementation required	---	---
45.	Bin Liu, Elke A. Rundensteiner	VM	Optimizing cyclic join view moves	Cost-driven architecture	Maintenance strategy	---	VM	Improvement of performance in distributed systems	More work required	Relational Model	TxnWrap System
46.	Crespo Márquez, P. Moreu de León, J.F. Gómez Fernández, C. Parra Márquez, M. López Campos	Design	Maintenance Management	Maintenance Management framework	Integration of generic models	Effective & Efficient system	D/M	Provided practical vision	More work required	---	---
47.	Jose-Norberto Maz´on, Juan Trujillo	Design	Develop MD model	MDA fraework	Hybrid based approach	Time & effort saving	D/M	Proposed framework decreases the inherent complexity	Require consideration for heterogeneous dimension hierarchies	Relational Model	Example-based analysis
48.	Howard Ong	Design	Deploy technology	Framework of DWA concepts	Custom-built solution	---	D	Optimal deployment of tools & technology	Require further study	Relational Model	---
49.	Panos Vassiliadis, Christoph Quix, Yannis Vassiliou, Matthias	Design	Modeling dynamic DW parts	Metamodel of DW operational process	Complementing existing architectures & models	Quality factors	D/M	DW efficient process management Used global-as view	Employ local-as view approach	Relational Model	Experimental study Telos

	Jarke							approach			

V. REVIEW AND RESULTS

Following sub-sections provide a review of materialized view evolution in a data warehousing framework based dimension using various aspects:

5.1 Technique

With respect to technique, summarizing materialized view evolution i.e. which methods involved in evolution of materialized views in a data warehousing framework dimension, is most focused *viz*. View Evolution, Basic View Maintenance, Incremental VM, Self Maintainable Maintenance, Not self Maintainable Maintenance, View selection, View Synchronization, View Adaptation. The study found 4 authors discussing view evolution [10,17,21,24], 19 authors discussing basic VM [6,7], 16 authors discussing design [9,11], 7 authors discussing incremental VM [12,32], 1 author discussing view synchronization, 2 authors discussing View selection [16,25], but none on self Maintainable Maintenance, Not self Maintainable Maintenance, and View Adaptation, giving a total of 49 papers under the stated methods in a data warehousing framework dimension (Fig.2).

5.2 Tool supported

In order to be effective and useful, methods involved in evolution of materialized views in a data warehousing framework dimension must be implemented or analyzed effectively. Theoretical analysis or experimental analysis of methods involved in evolution of materialized views can also be done. Authors in [09,22] provided the implementation of materialized view evolution methods using front-end or back-end languages. Theoretical analysis has been carried out for the materialized view evolution methods in [19,21] by the respective authors. While, experimental analysis has been carried out in [7,12] by the respective authors. Although the methods in [8] have not been analyzed nor implemented at all by the authors. As can be inferred, theoretical analysis have been used more than any other analysis technique for methods in a data warehousing framework dimension (Fig.3).

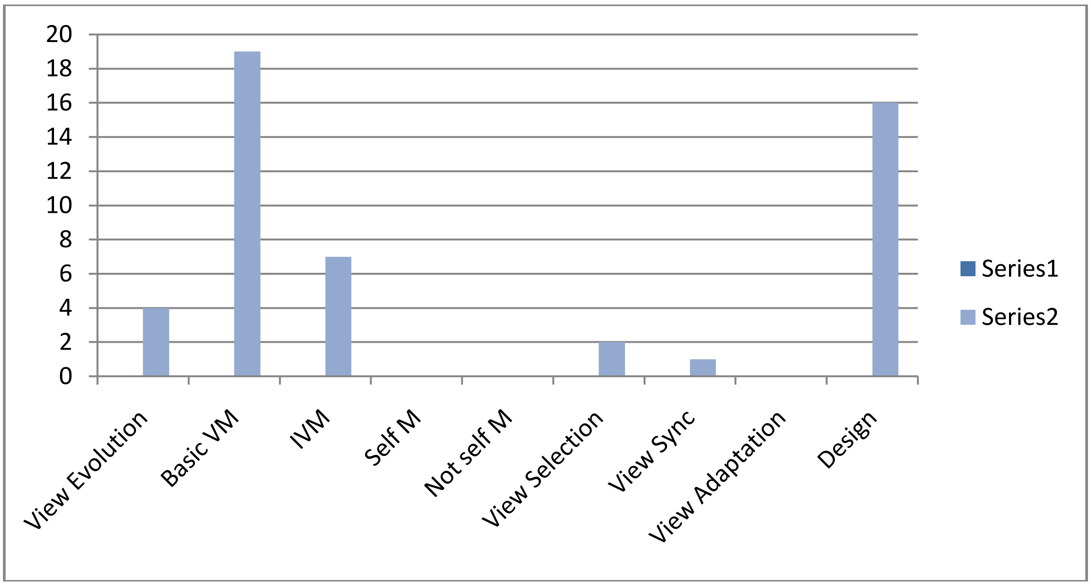

Figure.2. Classification of Technique Supported

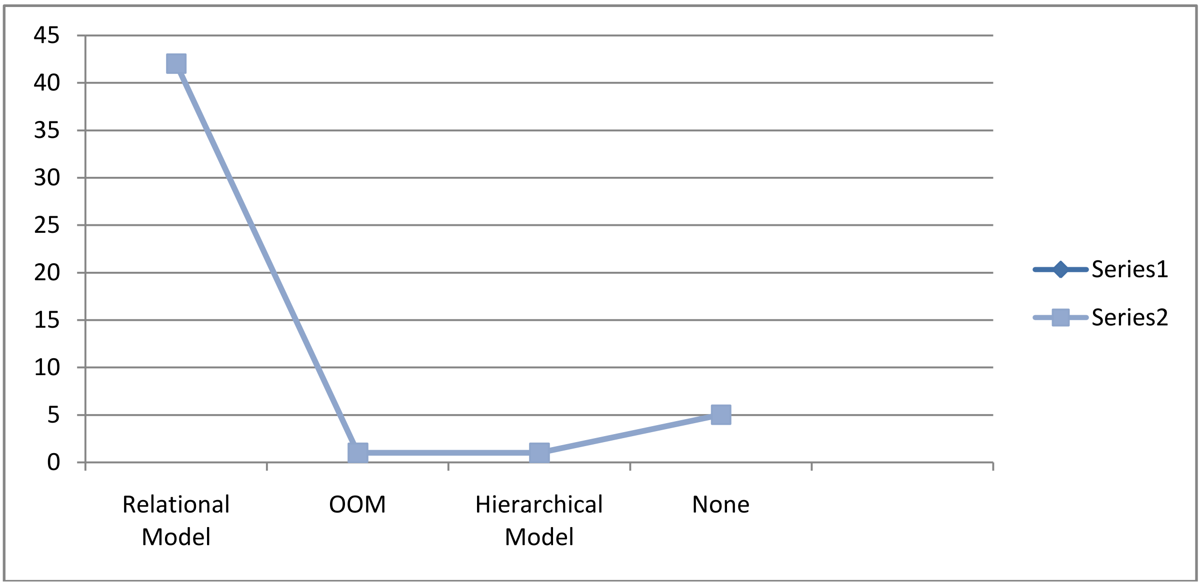

Figure.3. Tool Supported Classification

5.3. *Attributes Addressed*

The materialized view evolution methods considered in this study focus on various external quality attributes like accessibility [9], scalability[2],consistency[1], effectiveness [21,29], etc . But most of papers lacked implementation results for validating their claims. There are also some framework based materialized view evolution methods which have not been associated with any external quality attribute.

5.4. *Types of queries*

Based on the types of queries, the relational model [1] have been most frequently used by the authors for addressing materialized view evolution methods, while others used Object oriented [9] and Hierarchal [6] model. There are some materialized view evolution methods which have not been associated with any particular model by the authors. (Fig.4).

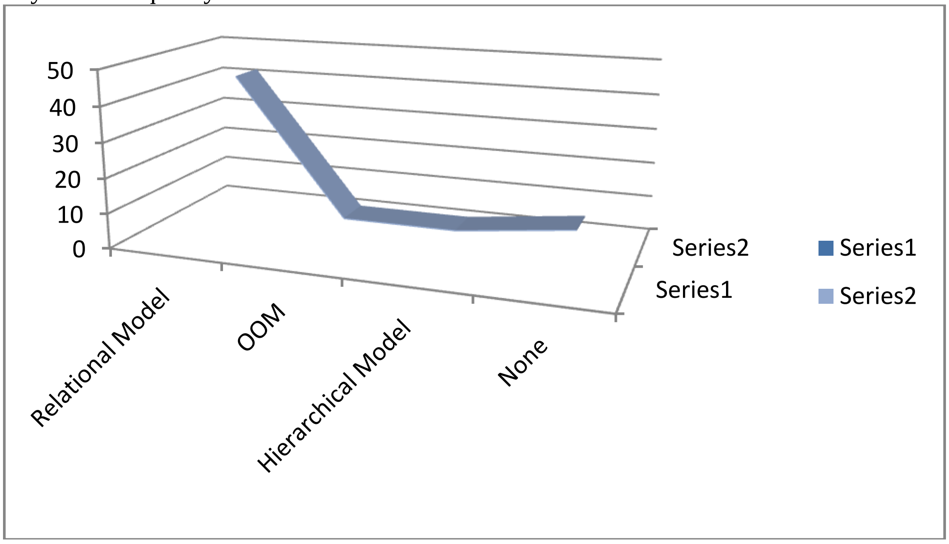

Figure.4. Classification of types of queries

5.5. *Applicable framework*

The materialized view evolution methods considered in this study also focus on applicable framework stage. The study found more than 30 authors [18,19] discussing view maintenance as applicable framework stage, while other authors [6,8] discussing design as applicable framework stage.

5.6. *Issues addressed*

In general, there are two type of algorithms for imposing constraints i.e. - *view maintenance algorithms [11]* and *view update [18]* algorithms. Some authors have addressed issues using *immediate* and *deferred* data synchronization algorithms [7, 12, 39], but many authors have mentioned other techniques [6] for imposing constraints in order to handle the required changes in the distributed database environment.

5.7. *Method Supported*

I*mmediate* and *deferred* data synchronization algorithms can be further classified on the basis of method's activities as optimization [8], Data integration [16] or others [6].

5.8. *Process supported*

Different authors have proposed different types of frameworks for materialized view evolution. Based on the various perspectives on the role of process, they are differently named as Eve [18], Quadris [15], SDCC framework [7] and so on. These frameworks are all useful in different situations, however they quantify for different perspective or support.

5.9. *Advantages & Disadvantages*

Mostly all the authors are handling maintenance anomalies, provided a framework for managing structural changes and have addressed the designing of algorithm for the materialized view evolution problem in order to reduce recomputation. They have provided a practical implication based framework for materialized view evolution [9, 11]. But some of the authors have not designed the algorithm nor have provided the validation / implementation studies [15] in regard to framework based materialized view evolution.

VI. CONCLUSION & FUTURE SCOPE

This study provides a critical survey of different approaches in which the materialized view evolution has been studied in relational databases and data warehouses as well as in a distributed setting. We have defined formally the materialized view evolution problem and identified the main materialized view evolution dimensions along with materialized view evolution methods have been classified. Based on the classification, we have discussed framework based materialized view evolution methods. Analysis of state of the art of materialized view evolution has shown that there is very few work on materialized view evolution in distributed databases and data warehouses[Bauer et. al 2003, Chaves et. al 2009, Yang et. al 2005] and no effective solution for peer to peer systems. Indeed, [Gribble et. al 2001] seems to be the only paper which deals with the view evolution problem in peer to peer environment. In fact, it is provided a full definition of the problem but without providing any algorithm or detail on how to select an effective set of views to materialize and place them at appropriate peers. Thus, one of challenging directions of future work aims at addressing the materialized view evolution problem more efficiently in a distributed setting and semantic web databases.

VII. REFERENCES

[1]. Kevin O'Gorman, Divyakant Agrawal, and Amr El Abbadi, "Posse: A Framework for Optimizing Incremental View Maintenance at Data Warehouses", Data Warehousing and Knowledge Discovery. First International Conference, Dawak'99. Proceedings (lecture Notes in Computer Science Vol.1676), 1999.

[2]. Yousry Taha, Arsany S. Sawiros, Noha Adly, "AN EFFICIENT DATA WAREHOUSING FRAMEWORK"

Computing - Technology and engineering, e-Publisher: CiteSeerX, 2009.

[3]. Yannis Kotidis, Nick Roussopoulos, “DynaMat: A Dynamic View Management System for Data Warehouses”, SIGMOD '99 Proceedings of the 1999 ACM SIGMOD international conference on Management of data, Pages 371-382, ACM New York, NY, USA 1999

[4]. Aristides Triantafillakis, Panagiotis Kanellis, Drakoulis Martakos, “An agent-based Framework for materialized view maintenance in Collaborative Electronic Commerce Environments”, Materialized view maintenance in Electronic Commerce environments, 2002

[5]. Dimitri Theodoratos, Timos Sellis, “Designing Data Warehouses” Elsevier Preprint , 1999

[6]. Mokrane Bouzeghoub, Zoubida Kedad, “A Quality-Based Framework for Physical Data Warehouse Design” Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000) Stockholm, Sweden, June 5-6, 2000

[7]. X. Zhang, E.A. Rundensteiner “The SDCC Framework for Integrating Existing Algorithms for Diverse Data Warehouse Maintenance Tasks”. Proceedings International Database Engineering and Applications Symposium (IDEAS), August 1999, Montreal, Canada.

[8]. Darja Solodovnikova, “Data Warehouse Evolution Framework”, Proceedings of the Spring Young Researcher's Colloquium on Database and Information Systems SYRCoDIS, Moscow, Russia, 2007

[9]. José A. Rodero, José A.Toval, Mario G. Piattini, “The audit of the Data Warehouse Framework” Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'99) Heidelberg, Germany, 14. - 15. 6. 1999

[10]. I. Anica-Popa, G. Cucui, “A Framework for Enhancing Competitive Intelligence Capabilities using Decision Support System based on Web Mining Techniques”, Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 Vol. IV (2009), No. 4, pp. 326-334

[11]. M. Golfarelli, S. Rizzi, “A Methodological Framework for Data Warehouse Design”, Proceedings of First International Workshop on Data Warehousing and OLAP (DOLAP, in connection with CIKM'98), Washington, D.C., USA, November 1998.

[12]. M.Jarke, M. A. Eusfeld, C. Quix, P. Vassiliadis,” "Architecture and Quality in Data Warehouses", Proceeding of the 10th International Conference on Advanced Information Systems Engineering (CAiSE'98), Pisa, Italy, June 1998.

[13]. Anton Yrjönen and Janne Merilinna, “Extending the NFR Framework with measurable Non Functional Requirements” The 2nd international workshop on Non-functional System Properties and Domain Specific Modeling Languages NFP in DSML2009.

[14]. Joseph O. Chan, “ Building Data Warehouses Using The Enterprise Modeling Framework”, Journal of International technology and Information Management 2004 Volume 13, Number 2.

[15]. J. Akoka, L. Berti-Équille, O. Boucelma, M. Bouzeghoub, I. Comyn-Wattiau, M. Cosquer V. Goasdoué-Thion, Z. Kedad, S. Nugier, V. Peralta, S. Sisaid-Cherfi,” A FRAMEWORK FOR QUALITY EVALUATION IN DATA INTEGRATION SYSTEMS” SBBD 2004: 134-147

[16]. Dimitri Theodoratos, Mokrane Bouzeghoub,” A General Framework for the View Selection Problem for Data Warehouse Design and Evolution” DOLAP '00 Proceedings of the 3rd ACM international workshop on Data warehousing and OLAP, Pages 1 – 8, ACM New York, NY, USA 2000.

[17]. M. Blaschka. “FIESTA: A Framework for Schema Evolutionin Multidimensional Information Systems”. In 6thCAiSE Doctoral Consortium, Heidelberg, 1999.

[18]. Anisoara Nica, Elke A. Rundensteiner,” Using Containment Information for View Evolution in Dynamic Distributed Environments” DEXA '98 Proceedings of the 9th International Workshop on Database and Expert Systems Applications, Page 212 , IEEE Computer Society Washington, DC, USA 1998.

[19]. Jian Yang, Kamalakar Karlapalem, Qing Li, “A Framework for Designing Materialized Views in Data Warehousing Environment “Proceedings of the 17th International Conference on Distributed Computing Systems (ICDCS '97), IEEE 1997.

[20].S.Chen, X. Zhang, E.A. Rundensteiner. “A Compensation-based Approach for Materialized View Maintenance in Distributed Environments”. In Computer Science Technical Report, Worcester Polytechnic Institute, Worcester, MA, USA (2004).

[21]. Chuan Zhang, Jian Yang,” Materialized View Evolution Support in Data Warehouse Environment” Sixth International Conference on Database Systems for Advanced Applications (DASFAA’99), 1999.

[22]. Amy J. Lee, Anisoara Nica, Elke A. Rundensteiner,” The EVE Approach: View Synchronization in Dynamic Distributed Environments”, IEEE TRANSACTIONS ON KNOWLEDGE ANDDATA ENGINEERING, VOL. 14, NO. 5, SEPTEMBER / OCTOBER 2002.

[23]. Xin Zhang, Elke A. Rundensteiner,” The SDCC Framework For Integrating Existing Algorithms for Diverse Data Warehouse Maintenance Tasks”, Database Engineering and Applications, 1999. IDEAS '99. International Symposium Proceedings , Aug 1999 Page 206 – 214.

[24]. George Papastefanatos, Panos Vassiliadis, Alkis Simitsis, Konstantinos Aggistalis, Fotini Pechlivani, Yannis Vassiliou, “LANGUAGE EXTENSIONS FOR THE AUTOMATION OF DATABASE SCHEMA EVOLUTION” ICEIS (1) 2008: 74-81.

[25]. B. Ashadevi, Dr. P. Navaneetham,” A Framework for the View Selection Problem in Data Warehousing Environment” International Journal on Computer Science and Engineering Vol. 02, No. 09, 2010.

[26]. Garima Thakur, Anjana Gosain,” DWEVOLVE: A Requirement Based Framework for Data Warehouse

Evolution" ACM SIGSOFT Software Engineering Notes, Page 1, November 2011 Volume 36, No.6.

[27]. Resmi Nair, Campbell Wilson, Bala Srinivasan," A Conceptual Query-Driven Design Framework for Data Warehouse", World Academy of Science, Engineering and Technology 25, 2007.

[28]. Jose-Norberto Maz´on, Juan Trujillo," A Hybrid Model Driven Development Framework for the Multidimensional Modeling of Data Warehouses", SIGMOD Record, June 2009 (Vol. 38, No. 2)

[29]. Ernest Teniente, Toni Urpí, "A Common Framework for Classifying and Specifying Deductive Database Updating Problems" Proc. Int. Conf. on Data Engineering (ICDE '95).

[30]. Richard Hull, Gang Zhou. "A Framework for Supporting Data Integration Using the Materialized and Virtual Approaches". In Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 481-492, June 1996

[31]. Solodovņikova D. "The Formal Model for Multisession Data Warehouse Evolution", Post conference proceedings of the 8th International Baltic Conference on Databases and Information Systems, Frontiers in Artificial Intelligence and Applications, IOS Press, 2008.

[32]. D. Agrawal, A. El Abbadi, A. Singh, T. Yurek., "Efficient View Maintenance in Data Warehouses". In Proceedings of the 1997 ACM International Conference on Management of Data, pages 417-427, May 1997

[33]. Matthias Jarke, Yannis Vassiliou, "A Framework for Choosing a Database Query Language". ACM Comput. Surv. 17(3): 313-340 (1985)

[34]. A. Crespo Márquez, P. Moreu de León, J.F. Gómez Fernández, C. Parra Márquez & V. González, "The maintenance management framework: A practical view to maintenance management", Safety, Reliability and Risk Analysis: Theory, Methods and Applications – Martorell et al. (eds), 2009 Taylor & Francis Group, London.

[35]. Yannis Vassiliou, Manolis Marakakis, Panagiotis Katalagarianos, Lawrence Chung, Michalis Mertikas, John Mylopoulos, "IRIS — A mapping assistant for generating designs from requirements", Lecture Notes in Computer Science, 1990, Volume 436/1990, 307-338.

[36]. Zhongmei Shu, Yayao Zuo, Yong Tang, "A collaborative framework for parallel view maintenance", 12th International Conference on Computer Supported Cooperative Work in Design, 2008. CSCWD 2008, Pages 81 – 86.

[37]. E. Levrat, B. Iung, A. Crespo Marquez, "E-maintenance: review and conceptual framework", Production Planning & Control: The Management of Operations Volume 19, Issue 4, pages 408-429 , 2008 .

[38]. Dallan Quass, "Maintenance Expressions for Views with Aggregation", In Proceedings of VIEWS. 1996, 110-118.

[39]. Mahesh B. Chaudhari, Suzanne W. Dietrich, "A Distributed Event Stream Processing Framework for Materialized Views over Heterogeneous Data Sources", VLDB 2010, Singapore.

[40]. Bin Liu and Elke A. Rundensteiner, "Cost-Driven General Join View Maintenance over Distributed Data Sources", Proceeding ICDE '05 Proceedings of the 21st International Conference on Data Engineering Pages 578-579.

[41]. Lawrence Chung, Panagiotis Katalagarianos, Manolis Marakakis, Michalis Mertikas, John Mylopoulos, Yannis Vassiliou, "From information system requirements to designs: A mapping framework", Published by Elsevier Information Systems Volume 16, Issue 4, 1991, Pages 429–461.

[42]. Anton P. Karadimce Susan D. Urban, "A Framework for Declarative Updates and Constraint Maintenance in Object-Oriented Databases", IEEE, 1993.

[43]. K.Y. Kutucuoglu, J. Hamali, Z. Irani, J.M. Sharp, (2001) "A framework for managing maintenance using performance measurement systems", International Journal of Operations & Production Management, Vol. 21 Iss: 1/2, pp.173 – 195.

[44]. Nikiforova, O., "Discussing the Difference between Model Driven Architecture and Model Driven Development in the Context of Supporting Tools", Fourth International Conference on Software Engineering Advances, 2009. ICSEA '09, pp 446 – 451.

[45]. Bin Liu, Elke A. Rundensteiner, "Optimizing Cyclic Join View Maintenance over Distributed Data Sources", IEEE transaction on Knowledge and Data Engineering, March 2006 (vol. 18 no. 3) pp. 363-376

[46]. A. Crespo Márquez, P. Moreu de León, J.F. Gómez Fernández, C. Parra Márquez, M. López Campos, (2009) "The maintenance management framework: A practical view to maintenance management", Journal of Quality in Maintenance Engineering, Vol. 15 Iss: 2, pp.167 – 178

[47]. Jose-Norberto Maz´on, Juan Trujillo," A Hybrid Model Driven Development Framework for the Multidimensional Modeling of Data Warehouses", SIGMOD Record, June 2009 (Vol. 38, No. 2).

[48]. Howard Ong, "Data Warehousing - An End-To-End Solution", Oracle Open World 1999, Brisbane.

[49]. Panos Vassiliadis, Christoph Quix, Yannis Vassiliou, Matthias Jarke, "DATA WAREHOUSE PROCESS MANAGEMENT", 12th international conference on advanced information systems engineering (CAiSE 00) Volume 26 Issue 3, May 2001 Pages 205 – 236.