# An Architecture for Integrated Online Analytical Mining

Muhammad Usman

Department of Computer and Informataion Sciences, Auckland University of Technology, New Zealand Email: usmanspak@gmail.com

Sohail Asghar

Center of Research in Data Engineering (CORDE), Muhammad Ali Jinnah University, Islamabad, Pakistan Email: sohail.asghar@jinnah.edu.pk

Abstract— Online Analytical Processing (OLAP) technology is an essential element of the decision support system and permits decision makers to visualize huge operational data for quick, consistent, interactive and meaningful analysis. More recently, data mining techniques are also used together with OLAP to analyze large data sets which makes OLAP more useful and easier to apply in decision support systems. Several works in the past proved the likelihood and interest of integrating OLAP with data mining and as a result a new promising direction of Online Analytical Mining (OLAM) has emerged. In this paper, a variety of OLAM architectures in the literature were reviewed and the limitations in the previously reported work have been identified. Literature review reveals the fact that none of the previously reported OLAM architectures have integrated enhanced OLAP with data mining. We enhanced the performance of OLAP in terms of cube construction time and visualization by providing interactive visual exploration of data cube. Furthermore, automatic OLAP schema generation has never been introduced as a component of OLAM architecture. The aim of this paper is to propose an integrated OLAM architecture that not only overcomes the existing limitations but also extends the architecture by adding an automation layer for the schema generation. In the proposed work, hierarchical clustering has been used as the data mining technique and three types of schemas namely star, snowflake and galaxy were automatically generated. A prototype of automatic schema generation has been developed and schema generation algorithms have been provided. In addition to this, we implemented and deployed the proposed architecture. Validation has been done by performing experiments on real life data set. Experimental results prove that the proposed architecture improves the cube construction time, empowers interactive data visualization, automate schema generation, and enable targeted and focused analysis at the front-end. By integrating enhanced OLAP with data mining system a higher degree of development is achieved which makes significant advancement in the modern OLAM architectures.

*Index Terms*—Automatic schema, cube, decision support system, data mining, data visualization, hierarchical clustering, OLAP

## I. INTRODUCTION

Online Analytical Processing (OLAP) technology refers to a set of data analysis techniques to view the data from all of the transactional systems in an interactive way in order to support the decision-making process. According to [1] and [2] OLAP systems have rapidly gained momentum in both the academic and research communities, mainly due to their allowance for quick, multi-dimensional analysis capabilities. It plays a vital role in data analysis to identify the key performance indicators of business and other application domains. In addition to this, it is an essential element of the decision support system and permits decision makers to visualize huge operational data for consistent and interactive analysis. Furthermore, it is becoming more accepted due to the recognition of the business value of exploring and querying huge amounts of data sets in a multidimensional way. In the last few years OLAP systems are the predominant front-end tools used in data warehousing environments and the OLAP system's market has developed rapidly. According to [3], the main reasons for this rapid development are as follows;

i)These systems are being used by knowledge workers to obtain quick answers of complex queries on their huge operational data.

ii) Allow users to perform quick, efficient and interactive analysis on organizational historical data.

iii) Specially designed with the aim of better supporting the retrieval of higher level summary information from detailed data.

iv) Offer better performance for aggregate queries

However, the growing complexity and volumes of the data to be analyzed impose new requirements on OLAP systems [4]. To fulfill these increasing requirements, a number of OLAP enhancement techniques have been proposed in the past [5,6,7,8,9,10,11,12,13,14,15,16]. Although some of the previously discussed research work proved the enhancement of OLAP systems but none of the work was intended towards focusing on the integrated enhancement of OLAP performance and visualization and there exist a strong need for it [17]. One of the main contributions in this paper is the integrated enhancement of OLAP which improves the performance in terms of cube construction time and intensifies visualization API.

Along with the integrated enhancement of OLAP, the other major contribution is the combination of OLAP and data mining. According to [18], data mining is a step in the overall concept of knowledge discovery in databases (KDD) and data mining techniques like Association [19], Classification [20], Clustering [21] and Trend analysis [22] can make OLAP more useful and easier to apply in decision support systems. More recently, data mining techniques have been applied in decision support applications in order to detect patterns and to extract knowledge form data. Several previous works [5, 7, 8 and 23] proved the likelihood and interest of coupling OLAP and data mining. However, [24] introduced a new concept of integrating OLAP and data mining and named it as Online Analytical Mining (OLAM).

OLAM is a promising direction for mining knowledge from multidimensional databases [23]. Significant works in this area include [24, 25, 26, 27 and 28]. These authors used different data mining techniques in their areas of interest and produced better results through the concept of OLAM. In this paper, a variety of OLAM architectures were reviewed and their limitations have been identified and highlighted. A major limitation revealed in literature review is the fact that none of the previously proposed architectures integrated enhanced OLAP with data mining. In the proposed architecture, we integrated enhanced OLAP with a data mining technique known as hierarchical clustering.

Hierarchical clustering is one of the dominant data mining techniques in the area of OLAM. In this paper work, we have also used the hierarchical clustering technique of data mining as a pre-processing step. Previously, hierarchical clustering has been used by [29, 30, 31, 32, and 33] in conjunction with OLAP systems for a number of reasons which include query optimization, physical data organization, fast data access and reduced storage cost. We intend to use the results of this hierarchical clustering technique in the form of data clusters to generate automatic schemas for enhanced OLAP system.

As far as the automatic generation of OLAP schema is concerned, [34, 35, 36, and 37] reported the work in this area. To our knowledge none of these works used hierarchically clustered data to generate OLAP schema. Furthermore, authors have used ER diagrams or conceptual graphical models to produce only one type of schema which is mostly star schema. Another major limitation in the automation work is that the tools and techniques used for schema generation only draws or identify the structure of schema and doesn't populate the actual data in the generated schema. In this paper work,

we are not only generating OLAP schema of three types namely; star, snowflake and galaxy but our automated schema builder tool provides automation in uploading the data from clustered file to relevant portions of the automatically generated schema. On the basis of these observations and limitations, we proposed an integrated OLAM architecture. The proposed architecture, firstly, enhances the power of OLAP in terms of its performance and visualization, secondly, integrates the enhanced OLAP with data mining technique of hierarchical clustering and lastly, extends the existing OLAM architecture with the addition of automatic schema generation layer. We have developed a prototype of automatic schema generation layer and also provided the schema generation algorithms. In addition to this, we implemented and deployed the proposed architecture.

Validation has been done by performing experiments on a real life data set. Experimental results prove that the proposed architecture improves the cube construction time, empowers interactive data visualization, automate schema generation, and enable targeted and focused analysis at the front-end. By integrating enhanced OLAP with data mining system a higher degree of development is achieved which makes significant advancement in the modern OLAM architectures. Finally, to the best of our knowledge, the proposed integrated OLAM architecture is superior to other OLAM architectures proposed prior or in course of carrying out this research.

The rest of the paper is organized as follows. Section II highlights the research contributions. Section III covers the literature review, in section IV the integrated OLAM architecture is proposed, Section V discusses the architectural implementation, Section VI elaborates the validation and evaluation of the proposed architecture. Finally, in section VII the summary of the work and the possible future direction from this research has been discussed.

#### II. RESEARCH CONTRIBUTION

The title of this paper reflects the overall goal of the work, which is to propose an integrated OLAM architecture. Although many research areas in the field of OLAM have reached the state of maturity, new challenges arise when applying the OLAM architecture to novel usage scenarios. In the following paper, we focus on overcoming the deficiencies of the conventional OLAM architecture in providing adequate and focused analysis in decision support applications. We have accomplished the following research contributions in this paper:

- Improving OLAP performance in terms of *cube* construction time and enhancing data visualization through the use of visualization API.
- Identifying a strong area of research that is the integration of enhanced OLAP with data mining to extend the OLAM architecture.
- Using hierarchical clustering as a pre-processing step of data mining.
- Proposing the integrated OLAM architecture.

- Extending the OLAM architecture with the addition of automatic schema generation layer.
- Supporting the automatic generation of three types of OLAP schemas namely *star*, *snowflake*, and *galaxy*.
- Automating the deployment of generated schema in the database server along with the population of data in the schema.
- Implementing the proposed architecture using market based technological tools.
- Developing prototypes of automatic schema generation layer and front-end targeted analysis.
- Performing experiments on real-life data sets using the prototype to compare and evaluate the results of the proposed architecture.
- Validating the experimental results and highlighting the benefits attained.

## III. LITERATURE REVIEW

In this section, literature review of integrated OLAM technology is presented. The purpose of this literature review is to study, analyze and identify the limitations of the OLAM technology. As described earlier, integrated OLAM technology is the combination of enhanced OLAP and data mining technique with automation in schema generation process. Additionally, the focus of this work is the use of hierarchical clustering to automate the process of OLAP schema generation. This diversifies the scope of literature review and comes under the larger umbrella of OLAM. Our overview of related topics focuses on four major themes: (1) OLAP enhancement, (2) integration of OLAP enhancement techniques, (3), use of hierarchical clustering with OLAP, and (4) automation in OLAP schema generation.

#### A. OLAP Enhancement

In this section, we review various significant OLAP enhancement techniques in the literature. The major focus of this section is performance improvement and visualization enhancement techniques. Firstly, we discuss some of the performance improvement techniques followed by the visualization enhancement techniques.

According to Goil et al. [5], parallel computing can be used as a way to enhance the performance of OLAP. It has been identified in the research work that typically OLAP operations are responsible for running complex queries on the underlying data, so these operations increase data processing time. The introduction of high performance parallel computers can lessen the computational time by distributing the tasks among various computer resources. The authors proposed an algorithm for Cube construction on parallel computers using distributed memory. The limitation of the work is the usage of a small and simple data set for the experiment. If the data set is large then a mismatch in the order of the dimensions can occur, as the

© 2011 ACADEMY PUBLISHER

multidimensional array doesn't allow the memory's use of their *sort* method to be efficient.

Similarly, Papadias et al. [6], reported on the efficient OLAP operations in Spatial Data Warehouses. Authors identified the problem that the computation of spatial data is expensive as online processing of such complex spatial data is inapplicable. They suggested a data structure, named aR-Tree, which combines a spatial index with the materialization technique. The strength of their work was the identification of the position of the objects in space in which the groupings and the hierarchies among the dimensions were unknown at design time. The work done is limited only to Spatial Data Warehouses and it also does not give any indication of the OLAP storage types such as ROLAP and MOLAP for Spatial data warehouses. The dataset used by the authors was found to be insufficient for the acceptance of this technique for all spatial cases.

In addition to the OLAP enhancement work, Asghar al. [7], proposed a functionality-enhancement et technique using Growing Self-organizing Neural Networks. This technique proposes the integration of data mining with OLAP by passing on mined data to the OLAP engine for a more focused analysis and hence adding intelligence to the OLAP system and thereby helping its users in sharp analysis of data. The major limitation is the manual schema generation of hierarchical clusters for the OLAP engine to perform operations. Furthermore, the data set has to be in a special format for the generation of hierarchical clusters using Growing Self-Organizing Map (GSOM) method. The enhancement architecture provided by the authors does not satisfactory answers for complex data sets. No indication was given as to how the hierarchical clusters generated from the dataset will be visualized.

To achieve better OLAP performance, а new contribution in the performance improvement area by associating OLAP and data mining was proposed. Messaoud et al. [8] suggested an enhanced OLAP operator based on the Agglomerative Hierarchical Clustering (AHC). The main problem identified is OLAP limitation of aggregation and summarization of complex objects like text, images, sounds and videos. The operator called Operator for Aggregation by Clustering (OpAC) is able to provide significant aggregates of facts referred to complex objects. The strength of the work is the combination of OLAP and data mining techniques as data mining can discover knowledge from both simple and complex data. The major limitation of the technique is that for n to classify, the AHC individuals generates n hierarchical partitions but it does not give guidance as to a best partition to choose. Data miner has to decide the number of clusters that corresponds to the context and to the goal of his/her analysis.

So far we have discussed the performance improvement techniques, now we shift our review focus to some of the

omprehensive and

major OLAP visualization enhancement work. For the purpose of visualization enhancement, Maniatis et al. [9], recommended the Advanced Visualization of OLAP. They suggested the Cube Presentation Model (CPM), which can be naturally mapped with an advanced visualization technique from the Human-Computer interaction area, called Table Lens. They formalized the simultaneous presence of more than one queries which was done in two layers (Presentation Layer and Logical Layer). The limitation of the work is that the visualization technique used for the screens is restricted only to one aspect of visualization enhancement. A comparative analysis is required in order to improve the visualization range of OLAP data.

Followed by the above advanced visualization technique, Sifer [10] introduced a visual interface technique for exploring OLAP data with coordinated-dimension hierarchies. They identified the problem that a number of text based and visual interfaces for querying multidimensional data exist but many of these interfaces are not applicable to OLAP. The reason is the lack of support for the use of dimensional hierarchies for selection and aggregation. The main contribution is the progressive view coordination interface which provides better support for query refinement than existing interfaces, by helping users decide the next query step with intermediate result overviews, and also helping the users to change a previous selection decision with retained selection context views. The strength of their work is the implementation of SGViewer visualization tool for their technique using a web log dataset. The limitation is that a very simple and small (only 3 dimensions) web log dataset was used for the illustration and, to use this technique the data has to be converted into Structured Graph Format Extensible Mark-up Language (XML) document. Therefore, it is a specific data format conversion limitation.

Voss et al. [11], adopted another way and suggested an extension in an advanced tool (CommonGIS) for the highly interactive visual exploration of spatial data. The identified problem was that the CommonGIS tool lacks the connection to OLAP warehouse to be a complete Business Intelligence (BI) application. They explored how to connect the tool to OLAP warehouses as another source of multi-dimensional data and designed architecture for the extension of CommonGIS. The strength of their work is that they allowed support of (Multi-Dimensional Expression) MDX for the manipulation of cube data using the application builder of the tool. The limitation of their work is the manual script writing process to convert client data into a compatible format so that the data is incrementally transmitted from the client source to the OLAP data cube.

In the same visualization enhancement area, Scotch et al. [12] proposed a tool for Spatial OLAP, called SOVAT (Spatial OLAP Visualization and Analysis Tool). The authors identified the problem that for community health

research a comprehensive and thorough analysis is needed for effective public health evaluation. The major contribution of their work is the development of tool for community health assessments. The strength of the tool is that it combines all necessary features of OLAP and GIS (Geospatial Information Systems) into one system to support comprehensive community health decisions. The limitation of this system is that it allows the visualization of spatial data and only permits the analysis on numeric data; however while performing analysis on spatial databases the researchers require tool support for not only number data but also other complex data types such as images, videos and maps. The community health researchers are more interested in the patterns or areas of an image for the evaluation purposes, rather than exploring numerical analysis of the data using OLAP.

Cuzzocrea et al. [13] introduced a hierarchy-driven compression technique for the advanced visualization of multi-dimensional cubes. This technique relied on the facility of generating a "semantic-aware" compressed representation of two-dimensional OLAP views. The strength of this particular technique is that it is most suitable for the handheld-devices where compression in the data cubes is required. The limitation is that it only supported two-dimensional OLAP views but, in real scenarios, there are multiple dimensions and facts in the dataset. The technique is more suitable for MOLAP scenarios and not very effective in the case of ROLAP, where multiple attributes exist in the relational dataset and the hierarchies are more complex within each dimension.

In 2005, [14] presented a new visual interactive exploration technique for OLAP. This work is similar to this paper work in terms of OLAP user facilitation. This technique and the proposed enhanced architecture, allows users who have less knowledge of OLAP technology to explore and analyze OLAP data cubes without generating sophisticated queries. The following year, [15] proposed a framework for querying complex multi-dimensional data with a major effort applied to transforming irregular hierarchies to make them navigable in a uniform manner. Lastly, in 2008, [16] introduced a comprehensive visual exploration framework which implements OLAP operations in form of powerful data navigation. Users can explore data using a variety of interactive visualization techniques using the proposed framework.

Research communities have often neglected the datavisualization problem in the past [13]. But we note that a considerable amount of work is recently being performed by various research communities on the visualization of OLAP data which indeed can play a very vital role in effective analysis of data. Our work in this paper is similar to the above mentioned contributions as it also improves the performance of OLAP in terms of cube construction time. Furthermore, we are using cube visualization API, which dramatically enhances the visual interactive exploration of cube data with the help of charts, graphs and data grids at the front end. In the next section, we present some noteworthy work done on the integration of OLAP enhancement techniques.

## B. Integrated enhancement of OLAP

In this section, we highlight the research done on the integrated enhancement of OLAP performance and its data visualization. Although some of the previously discussed research work proved the enhancement of OLAP systems but none of the work was intended towards focusing on the integrated enhancement of OLAP performance and visualization and there exists a strong need for it [17]. Asghar et al. [7] proposed a using functionality-enhancement technique Self-Organizing Neural Networks. To some extent this technique proposed the integrated enhancement of OLAP by passing the mined data to the OLAP engine for a more focused analysis and hence adding intelligence to the OLAP system. The major limitation of the work was the deficiency of enhancement on the visualization aspects of OLAP systems. Their enhancement architecture doesn't provide satisfactory answers for the visualization of complex data sets. During the last few years, a lot of work has been done on the enhancement of OLAP particularly on its performance and visualization.

Various architectures [7, 68, and 69] were proposed for the one dimensional enhancement of OLAP systems. But, the most relevant work in this integrated enhancement was done by [70]. The authors proposed architecture for the integrated enhancement of OLAP system. They have used visualization tools and developed a prototype which improved performance in terms of *cube* construction time. Both performance and visualization aspects were catered by the proposed architecture. The limitation of the work was again the dependency of GSOM to produce hierarchical clusters and the manual schema generation of the hierarchical cluster tables. The proposed architecture has a gap which is the transformation of cluster relationship table to OLAP schema. Authors only validated the prototype using only one type of OLAP schema that is the star schema. The paper work is different from the above mentioned literature as we are not using GSOM for hierarchical clustering. Furthermore, a schema builder component has been proposed which takes clustered data and generates three types of OLAP schemas automatically. Additionally, cube has been constructed which holds the clustered data to form an OLAM cube to support the process of online analytical mining which was missing. As this paper deals with the integration of OLAP and data mining, in the next section some important work on the combination of OLAP and data mining has been reviewed.

## C. Combining OLAP with Data Mining

Several previous studies [5, 7, 8 and 23] emphasized the likelihood and interest of coupling OLAP and Data Mining. Efficient and effective data mining integrates the

concept of [24], which is the integration of OLAP and OLAM systems. OLAM is a promising direction for mining knowledge from multidimensional databases [23]. Similarly, Hua [25] proposed and developed an interesting association rule mining approach called Online Analytical Mining of association rules. It integrated OLAP technology with association rule mining methods and leads to flexible multidimensional and multi-level association rule mining. Similarly, Josph et al. [26] presented a methodology that derives the association rule of web pages tick sequences according to the support level and confidence level of user requirements. This methodology was responsible for identifying a set of frequently accessed web pages on a website by a user. The result is the list of potential customers for a certain product or service on a target web page.

Dzeroski et al. [27] combined OLAP and Data Mining in a different way to discover patterns in a database of patients. Two data mining techniques, clustering and decision tree induction were used. Clustering was used to group patients according to the overall presence/absence of deletions at the tested markers. Decision trees and OLAP were used to inspect the resulting clustering and to look for correlations between deletion patterns, populations and the clinical picture of infertility.

Dehne et al. [28] studied the applicability of coarse grained parallel computing model (CGM) to OLAP for data mining. Authors presented a general framework for the CGM which allows for the efficient parallelization of the existing data cube construction algorithm for OLAP. Experimental data showed that this approach yield optimal speed up even when run on a simple processor cluster via a standard switch. The study shows that OLAP and data mining, if combined together, can produce greater benefits in a number of diverse research areas. Our proposed work is similar in terms of combining OLAP and data mining techniques but our main focus is on a particular data mining technique known as Hierarchical Clustering. Furthermore, we are using data mining as a pre-processing step to get better understanding of data before passing it to the automatic schema builder and which then generates schema for OLAP engine. In the next section, the related data mining technique which is hierarchal clustering and its usage with OLAP is the centre of attention.

#### D. Use of Hierarchical Clustering with OLAP

A number of data mining techniques which can be combined with the OLAP technology for the better solutions of research problems were described in the previous section. Hierarchical clustering of data is a well know mining technique and it has been used by a number of research communities in the past. In this paper hierarchal clustering has been used as a data mining technique with OLAP. In this section a review of hierarchical clustering in conjunction with OLAP technology has been performed. Markl et al. [29] suggested that OLAP performance can be improved by using the Multidimensional Hierarchical Clustering (MHC) technique. Clustering was introduced as a way to speed up query aggregation without additional storage cost for view materialization. The authors identified the problem with queries which either select a very small set of data or perform aggregations on a fairly large data set. The sole contribution of their work is an encoding scheme for hierarchical dimensions that enables clustering of data with respect to multiple, hierarchical dimensions. The major strength of the work lies in the comparison of their MHC technique with the traditional bitmap indexing approach on the real world data (7GB in size) and finding an increase in the performance up to the factor of 10.

Markl et al. [30], again utilized hierarchical clustering in the context of OLAP queries. Authors investigated the impacts on query processing in RDBMS when using UB-Trees and multidimensional hierarchical clustering for physical data organization. Furthermore, they illustrated the benefits achieved by performance measurements of queries using star schema for a real world application of a SAP business information warehouse.

Karaynnidis et al. [31], proposed a novel multidimensional file structure for organizing the most detailed data of the cube, the CUBE file. The CUBE file archives hierarchical clustering of data enabling fast access via hierarchical restrictions. It imposes low storage cost and adapts perfectly to extensive sparseness of data space. Results show that the CUBE File outperforms the most effective method proposed up to now for hierarchically clustering the cube, resulting in 7-9 times less I/Os on average for all workloads tested.

In the same context, Theodoratos et al. [32] claimed the Heuristic Optimization of OLAP in MHC (Multidimensionally Hierarchically Clustered) databases. They identified the problem that commercial relational database management systems use multiple onedimensional indexes to process OLAP queries that restrict multiple dimensions. They presented architecture for MHC databases based on the star schema called 'CSB' star. The focus was to facilitate the user to query the typical star schema and to formulate the given query for 'CSB' star schema using query processor. The limitation of their work was that they only tested a particular class of typical OLAP queries over the CSB star schema to check efficiency.

Tsois et al. [33], proposed cost based optimization of aggregation star queries on hierarchically clustered data warehouses. Hierarchically clustered data warehouses were used to apply a cost based method for the optimal application of the pre-grouping transformation. Authors identified the most suitable algorithms for the operations related to pre-grouping and derived detailed cost formulas for them. It is apparent from the review that various authors have used hierarchical clustering technique in different ways with OLAP to improve the functionality and to get better analytical results. The major difference of this paper work with the reviewed work of this section is that our use of hierarchical clustering is to generate hierarchal clusters to get an insight of the data before passing it further for OLAP operations. This mining technique has not been applied on the cube for archiving cube data or for the optimization of OLAP queries. Instead of query optimization, OLAP performance has been improved in terms of cube construction time through the use of hierarchically clustered data.

#### E. Automatic OLAP schema generation

In this section, we discuss the major research work on the automation process for the generation of OLAP schema. Hann et al. [34], proposed the generation of tool specific schemata for OLAP from conceptual graphical models. Their work described the design and implementation of the generation component in the context of their own *Bablefish* data warehouse environment. The principle issues of designing and implementing such a automatic schema generation component and the possible solutions have been discussed by the authors. Further topics are the use of graph grammars for specifying and parsing graphical multi-dimensional schema descriptions and the integration of the generation process into a metadata cantered modeling tool environment.

Peralta et al. [35], highlighted the existing work in the area of automation and claimed that automation is focused towards data models, data structures specifically designed for Data warehouse (DW), and criteria for defining table partitions and indexes. The major research contribution of the authors is a step forward towards the automation of DW relational design through a rule-based mechanism, which automatically generates the DW schema by applying existing DW design knowledge. The proposed rules embed design strategies which are triggered by conditions on requirements and source databases, and perform the schema generation through the application of predefined DW design oriented transformations.

For the sake of accomplishing the ease in the automation work, Tryfona et al. [36], built a conceptual model (*StarER*) for data warehouse design on the basis of user modeling requirements. The *StarER* model combines the star structure, which is dominant in data warehouses, with the semantically rich constructs of the ER model. Comparison of the proposed model with other existing models has been performed, pointing out differences and similarities. Examples from a mortgage data warehouse environment, in which *starER* is tested, that revealed the ease of understanding of the model, as well as the efficiency in representing complex information at the semantic level.

For the purpose of achieving the automatic generation, Song et al. [37] introduced an automatic tool for generating star schema entity-relationship diagram (ERD). A prototype named SAMSTAR was presented, which was used for the automatic generation of star schema from an ERD. The system takes an ERD drawn by *ERwin* Data Modeler as an input and generates star schemas. SAMSTAR displays the resulting star schemas on a computer screen graphically. With this automatic generation of star schema, this system helps designers reduce their efforts and time in building data warehouse schemas.

It is evident from the literature on automatic generation of schema that none of the previous works used hierarchically clustering to generate OLAP schema. Previous work on automation has a number of limitations as well such as support for only one schema type, dependency of ER diagram modeling and having to design rule based mechanism before the automation process.

This work is different as compared to existing automation work in the literature. Firstly, the results of hierarchical clustering are used to generate any of the three types of schemas. Furthermore, the work is not limited to designing the schema but it also populates the data from the clustered data source to the automatically generated schema tables. In addition to this, the schema builder also communicates with database server and deploys the generated schema in the server as a complete database. In section 3.6, comparison has been made of this paper work with the four themes of the literature review. The purpose of this comparison is to emphasize the significance of our work and to highlight the existing limitations in the related work.

## F. Comparison of the proposed work with the literature reviewed

In the previous sections the individual papers were discussed. Now, this section presents a comparison of this paper work with the four major themes of the literature reviewed. Firstly, a substantial amount of work has been done on the enhancement of OLAP in terms of its performance and visualization. According to [5, 6 and 7], performance of OLAP systems can be improved with the help of different techniques like parallel computing, indexing and materialization, neural networking and agglomerative hierarchical clustering. This paper work is a bit similar in terms of OLAP performance improvement but we have a broader scope as in this paper, we are proposing an architecture which includes the OLAP's data visualization enhancement as well.

In the area of OLAP's visualization, research communities has also paid attention and reported the work on visualization enhancement in [9, 10, 11, 12, 13, 14, 15 and 16]. Visualization enhancement techniques like Cube Presentation Model (CPM), visual interfacing, interactive exploration, hierarchy driven compression and complex multi-dimensional querying were introduced.

Previously it has been identified that there is a strong need of integrating performance improvement and visualization enhancement techniques [17]. Furthermore by utilizing the work of [7], which lacked in the areas of visualization work of OLAP data, schema generation and cube processing, this paper work moved a step forward and introduced an advanced architecture for the integrated enhancement of OLAP by combining OLAP with data mining techniques in [70].

Various researchers have contributed in the combination of OLAP and data mining in the past. Significant works in this area include [24, 25, 26, 27 and 28]. These authors used different data mining techniques in their areas of interest and produced better results by combining OLAP with data mining. The two major techniques of data mining used in this regard were Association Rule Mining and Hierarchical Clustering. In this paper we have used the hierarchical clustering technique for the integrated enhancement of OLAP's performance and visualization. Hierarchical clustering has been used by [9, 30, 31, 32 and 33] in conjunction with OLAP systems for a number of reasons. Some authors used this technique for query optimization, physical data organization, fast data access and reduced storage cost. The work in this paper is different from these as we used hierarchical clustering as a pre-processing step of data mining. Furthermore, we are utilizing it to produce hierarchical relationship data for the automatic generation of OLAP schema.

As far as the automatic generation of OLAP schema is concerned, [34, 35, 36 and 37] reported their work in this area. To our knowledge none of these works used hierarchically clustered data to generate OLAP schema. Most of the work is using ER diagrams or conceptual graphical models to produce only one type of schema which is mostly *star* schema. Another major limitation in the work is that the tools and techniques used for automatic generation of schema only draws of identify the structure of schema and doesn't populate the corresponding data in the generated schema.

This paper work is not only generating OLAP schema of three types namely; *star, snowflake* and *galaxy* but our automated schema builder tool provides automation in uploading the data from clustered file to relevant portions of the schema. This feature along with other specifications distinguishes the work done in this paper with others. But in a nutshell the paper work is intended to support the OLAM in a distinctive way. It is distinctive because none of the previous work dealt with all the four major aspects of our proposed research.

## IV. PROPOSED ARCHITECTURE

The architecture for integrated OLAM has been proposed in this section. But before moving forward towards the architecture itself, it is important to emphasize the necessity of the proposed architecture that *why exactly do we need to have architecture for the integrated OLAM?* To answer this important question we begin the explanation starting with the overview of the previous



Figure 1. Proposed architecture for integrated OLAM

section. As discussed earlier, it is evident from the literature review that none of the proposed work in the past had targeted towards the integration of enhanced OLAP with data mining along with the automation in the schema generation process. Integrated OLAM is the combination of data mining and enhanced OLAP with the incorporation of automatic schema generation from mined results. A number of authors [5, 7, 8, 23, 24 and 25], in the recent years, have emphasized on the concept of combination of OLAP and data mining. Although this previously reported work justified the advantages of such combination but the major limitation in their work is the absence of enhanced OLAP system combined with data mining. Typically, a standard OLAP system is integrated with data mining. In parallel to this, various authors [6, 9, 10, 11, 12, 13 and 14] reported their work on the enhancement of OLAP in terms of its performance or data visualization. In this paper, it is recommended that such an enhanced OLAP system should be integrated with data mining so that the benefits of enhancement achieved in OLAP are combined with data mining techniques to produce overall improvement in the OLAM system. In addition to this distinctive combination, the integrated OLAM architecture has been extended with the

additional layer for automatic schema generation. For the purpose of this transformation we suggest an independent automatic schema generation module that fulfils both the requirements of schema generation and transformation of data. Based on the above discussion, there is a strong requirement of a significant architecture that takes the data mining results to automatically generate schema, transform mined data in the schema, incorporate enhanced OLAP in it, support OLAM along with overcoming the existing limitations and lastly, produce improved, focused and intelligent results. Figure 1 illustrates the proposed architecture for integrated OLAM.

The goal of the proposed architecture is to integrate enhanced OLAP with data mining and to provide automatic schema generation from the mined data. We elaborate the four distinct layers of the proposed architecture which are 1) Data Mining: pre-processing layer, 2) Automatic schema generation layer, 3) Enhanced OLAP layer and 4) User interface or Front-end layer. Each layer along with its components is discussed in separate sub sections. To begin with the explanation, we start with the data mining laver.

## A. Data mining: Pre-Processing Layer

This layer is responsible for the data mining process. The three components in this layer are as follows:

- **Dataset:** This component represents the data set. A large number of data set exist and this component is responsible for the selection of a particular data set on which data mining is to be performed.
- Hierarchical Clustering: Once the data set is selected, the data mining technique which is hierarchical clustering is applied in this component on the data. This component also handles clustering parameters which include linkage method (Single, Complete or Average), clustering directions (rows, columns, both) and similarity/distance measure (Euclidean, Manhattan, Simple matching for nominal or Pearson correlation coefficient). On the given set of clustering parameters this component produce the hierarchical clusters.
- **Clustered Data:** The hierarchical clusters generated by the previous component are stored and saved in the form of clustered data. This component of the layer also saves the hierarchy relationship among the data along with the actual data in each cluster. The hierarchically clustered data become the input of the next layer which is responsible for the automation process of schema generation.

## B. Automatic schema generation layer

The purpose of this layer is to automate the process of schema generation. To achieve this purpose the data mining layer feeds the mined data or clustered data to this layer for the generation of schema from clustered data set. The layer has distinct components and the working of each component is explained as follows:

- Schema Builder: It is the major component of schema generation layer and the sole function of this component is to build a particular type of schema for the clustered data set. As this component has to perform numerous schema generation activities, therefore it has a number of sub-components to achieve the purpose of building the schema. The sub-components of the schema builder are as follows:
  - **Clustered Data Reader:** This subcomponent of the schema builder reads the data generated by the hierarchal clustering and save it temporarily for future operations.
  - **Clustered Data Viewer:** It allows viewing the clustered data read by the

reader in the grid view. All the cluster hierarchy and the number of records present in each can be viewed through it.

- Schema Generator: After the clustered data has been read and viewed the data goes to the component called schema generator. The sub-components of the schema generator are explained below.
  - Dimensions and Facts Identifier: This is the first sub-component of the schema generator. In the generation process of schema the first important step is the identification of dimensions and facts from the clustered data. For this important purpose the data in the schema generator goes to the Dimension and facts identifier, which identify the dimensions and facts present in the clustered data. It also identifies the cluster hierarchy and the numeric type of data which are the facts or key performance indicators of the data.
  - **Dimensions and Facts Separator:** The identifier component just identifies the various dimension and facts present in the clustered data. But in real the dimension and the facts are to be separated in order to build the schema. For this purpose the Dimension and facts separator splits the identified dimensions and facts to be used for different types of schema generation.
  - Schema Type Selector: At the stage when dimensions and facts are identified and separated the Schema type selector component handles the designing of a specific type of schema. This selector allows the selection of a particular type of schema which can be a star, snowflake or galaxy.
  - Warehouse Builder: After determining the schema type to be generated the dimensions and facts along with the specific schema type goes in to another major and core unit of schema builder which is called the warehouse builder. The warehouse builder as its name implies, builds a warehouse of a particular schema type and consists of the following main sub-

components. The functionality of each of the component in the warehouse builder is described below:

- Dimensional hierarchy handler: It is the first component of the warehouse builder and it serves the purpose of handling the hierarchy in the dimensions. As the clustered data has its own hierarchy and the dimensional data identified and separated previously have to be in some hierarchical order within each dimension. Because the dimensional hierarchy is different for different type of schema that is why this handler component handles the hierarchy present within each dimension.
- Dimension and fact table creator: Because the builder warehouse automatically creates a warehouse in the form of a database so the creation of the dimension and fact table is a must. For this reason the dimension and fact table creator takes dimensional hierarchy from the previous component and the facts separated by the separator component and create the tables according to the specific schema type. Within each table created for a dimension the hierarchy structure is determined and created using the hierarchy handler output, which is different for each type of schema.
- *Table relationship manager:* For each type of schema the referencing of dimension and fact table is different. For star the dimensions are connected directly, in

Snowflake schema there exists normalized а relationship between the dimensions whereas for Galaxy both types of relationships are being made. To handle this complexity of relationship among dimension and fact tables created by the warehouse builder for a specific schema type, the table relationship manager plays its part and manages all kind of relationships among the tables accordingly.

- Data population unit: When all the tables are created along with the hierarchy within each dimension and the relationship is being created among the dimensions and fact tables, the data population unit comes in to play. The role of this unit is to load the clustered data in the intelligently. schema Therefore, this unit picks the clustered data and inserts it in specific columns of the dimension and fact tables so that this data can be used later for the OLAP operations that are to be performed in the next layer of the architecture.
- Schema Viewer: This component takes the generated schema from the schema generator and gives the view of the automatically generated schema in the form of schema diagram.

## C. Enhanced OLAP Layer

As our proposed work is the integration of data mining with enhanced OLAP, so in this section we discuss the layer which is exclusively responsible for the OLAP enhancement in the proposed architecture. Furthermore, the construction of cube from clustered data in the form of automatically generated schema is presented. The various components of this layer along with their internal functionality are as follows:

- **Database Server:** The main purpose of the database server is to store the automatically generated databases from the warehouse builder of the schema generator. The components of a database server which are used for the communication of architectural layers and internal function are explained below:
  - **Generated Databases:** This component stores the automatically generated schemas in the form of databases in the database server. For each kind of schema a specific database is automatically created along with all the dimensions and fact tables.
  - **External Component API:** The external component API is responsible for the communication involved in the creation of specific type of database within the database server. It uses the requests from the schema builder component and responds accordingly.
  - **Storage Engine:** This component of the database server handles the storage allocation of the generated databases in the database server. It allocates memory type for the created database and handles the size of the particular database in the memory.
  - Query Processor: warehouse builder of the schema generator component discussed in the previous layer send requests using the connection string to the database server in the form of queries. The query processor component reads the request sent by the warehouse builder and processes the queries. These queries when processed create the databases and the tables with specific columns along with the clustered data.
- OLAP Server: Similar to the database server which stores and provides a database management system OLAP server is another important component of this layer that handles the cube construction, storage and manipulation. OLAP server's core purpose is to handle the cube related queries and manipulations. The component worth discussing in this are as follows:
  - **Data Source:** This component of the OLAP server manages various sources of data which are to be used for cube construction process. In the proposed architecture of integrated OLAM,

automatically generated schema databases become the source in the OLAP server for cube construction. It means that the generated databases are connected with data source using this component for future use. So the storage of data is in the database server and the connection between the two servers handles the communication involved form one server to another.

- **OLAP Engine:** This component allows the construction of cube on a specific data connected with the data source component. It is the most powerful component of the server which handles the processing of cubes which involve aggregation of data at various hierarchical levels.
- **Cube Storage:** Once data source makes a connection with the data for the cube construction and the OLAP engine construct the cube. The cube is stored in the OLAP server and the component responsible for this is the cube storage component. It also allows the type of cube storage which can be MOLAP, ROLAP or HOLAP.
- **Cube Visualization API:** After the construction and processing of OLAM cube the cube visualization API comes in to play. The purpose of this component is to apply the cube data visualization methods to enhance the power of OLAP visualization. This component of the layer provides rich interactive analysis functions and operations that enhance the capabilities of standard OLAP system.
- D. User Interface Layer

This layer represents the user interface of the proposed architecture. At this layer, the OLAM cube, which is constructed and processed in the previous layer, allows targeted and focused analysis. The constructed cube is named as OLAM cube because it consist of mined data results of the hierarchical clusters with the aggregation of facts present in each cluster. It allows all the OLAP operation to be performed on it such as *drill-down, rollup* etc. The OLAM cube data goes through the visualization API which enhances the visual exploration capabilities of its data. It allows interactive and visual analysis on the OLAM cube data. Therefore, the user gets a real insight of this mined data at this user-interface layer.

The user interface layer represents the targeted and focused analysis. It is targeted and focused analysis because at the first layer of data mining when the clusters are generated through hierarchical clustering, user had a view to determine the cluster of his/her interest and continue to the second layer of automatic schema generation for that specific cluster. Moving on with the architectural flow, the schema of user's choice is generated automatically and cube is constructed only for the schema of that particular cluster. This is the reason why the final cube itself has the targeted data, therefore There are a number of significant benefits achieved through the proposed architecture. The list of these benefits are as follows:

- Use of data mining as a pre-processing step for the targeted analysis of data.
- Allowance of various hierarchical clustering algorithms and data linkage methods.
- Automatic generation of three types of schema from the clustered data.
- Complete diagrammatical view of the generated schema using schema viewer.
- Use of distributed database and OLAP servers.
- Dynamic creation of separate databases for each type of generated schema.
- Performance improvement of cube in terms of its construction time.
- Visualization enhancement for the interactive and powerful visual exploration of cube data through the use of visualization API.
- Construction of OLAM cube instead of simple OLAP cube.
- Powerful front-end for the focused and interactive analysis on the OLAM cube.

#### V. ARCHITECTURAL IMPLEMENTATION

In this section, based on our implementation, we discuss in detail the steps involved in the implementation of the proposed architecture. Some tools and technologies have been used for the implementation of each layer of the the analysis or visual exploration performed at this stage is on a focused and targeted data and not on the complete data set.

#### E. Benefits Achieved

proposed architecture and divided the implementation phase into five major steps. Figure 2 depicts the major steps of architectural implementation.

Each step of implementation phase is discussed individually in the following sub sections.

#### A. Hierarchical Clustering of Data

The proposed architecture starts with the layer of data mining and the technique used for data mining is the clustering. hierarchical The first step of the implementation is the hierarchical clustering of data. To serve the purpose of step 1, we used Hierarchal Clustering Explorer (HCE) tool for generating the hierarchical clusters of data. This tool takes input data file and allows the hierarchical clustering of given data based on different clustering parameters. Figure 3 shows the interface of cluster parameter selection form, which has the different parameters for selecting the type of hierarchical clustering.

At this point, user can select the parameters of his/her choice to perform specific type of hierarchical clustering on the data. Upon the selection of clustering parameters, the data as a result, is hierarchically clustered and the hierarchy details are shown in the HCE tool. Furthermore, the tool allows exporting the clustered data results in the form of text files. At this stage, after the clustering has been performed the results are exported which serve as the input of the next step of automatic schema generation. We elaborate the next step of automatic schema generation in the following section.



Figure 2. Sequence of architectural implementation steps

Data Partitioning :	None	C by Bow C by Column
heck columns to cluste		
Column Name		1
	BEAL	
	REAL	
	REAL	
RACE	REAL 💌	
Uncheck All	Check All	1
C Average Group Link Complete Linkage Single Linkage Shneiderman's 1by1 Iode Arrangement Meth Keep Right Child Re	age ods dder	Load Similarity Matrix for Rows     Cluster Columns     Load Similarity Matrix for Columns
imilarity/Distance Meas Euclidean Distance	ure	Use P.Values as Weights

Figure 3. Clustering parameter selection in HCE tool.

## B. Automatic Schema Generation

In this step, the cluster results file which was exported using the HCE tool becomes the input of the automatic schema builder. We developed a prototype to automate the process of schema generation. The prototype takes the clustered data and generates the schema of any particular type such as star, snowflake or galaxy/constellation. The prototype has been developed using the C sharp (C#) programming language in Microsoft Visual Studio.net 2005. The implementation logic used for the development of automatic schema builder is explained in the form of a main algorithm 5.2.1.

#### Algorithm 5.2.1

```
In put : H CD (hierarchically _ Clustered _ Data)

O utput : Schema (star, snow flake or galaxy)

M ethod :

//*Step1 : Reading _ Hierarchically _ Clustered _ data(HCD)

w hile (R \neq \emptyset)

do

read Riintorj

let CH be column headear

if (i \in \emptyset) then

for (j=1 to r.len) /* where len is the length

C H k \leftarrow rj

else

for (j=1 to r.len)

C Dm \leftarrow rj
```

```
//*Step 2 : Viewing _ HCD _ in _ dataviewer _(DV)
while
            (r \neq \emptyset)
                 if (i \ni 0) then
                      for (j=1 \text{ to } CH.len)
                                             /*len is the lenght of the string
                                send CHj to DV
                  else
                        for (k=1 \text{ to } CD.len)
                               send CDk to DV
//*Step3:Dimension& fact_identificatin_and_Seperation
let D be the dimension storage variable and let F be the fact variable
             while (r \neq \emptyset)
                    if (r_{i+1} \in I) / * w here I is an interger
                      call procedue Dim_sep(r i, r i+1)
                                                              /* where (ri and ri+1) \in D
                      else
                       call Fact_sep(ri) where (ri) \in F
                                   i \leftarrow i + 2
procedure Dim_sep(var1,var2) /* var1 and var2 temporary variables
                   D[m,n] \leftarrow D[ri,ri+1]
procedure Fact_sep(var3)
                  F[p] \leftarrow F'[ri]
//*Step4:Schema_type_selection
let ST be the selected schema type
          if (ST \leftarrow "star")
             call procedure (Genereate_star)
         else
              if (ST ← "snowflake")
                    call procedure (Generarate _snow flake)
         else
              if (ST ← "galaxy")
                          call procedure (Generate_galaxy)
         end if
//*Step5:Viewing _ generated _ schema
let DT be the dimension tables and let FT be the fact table
                           if (ST ← "star")
                                       or if (ST ← "snow flake")
                            read dimension tables into DT
                               while (DTi \neq \emptyset)
                                       send D T i to T V */ S V is the table viewer
                                       read fact tables into FT
                                       send FT to TV
                          else
                                if (ST \leftarrow " g a l a x y ")
                                      read dimensions in to DT
                          end if
                  while (DTi \neq \emptyset)
                                 send D T i for table view
                                 read fact tables in FT
                                         while (FTi \neq \emptyset)
                                         send FTi for table view
                  end all.
```

In the above algorithm the basic steps of automatic schema generator are represented using an algorithm.

#### C. Database and OLAP server deployment

In the previous step, automatic schema has been generated and the schema builder actually creates database automatically in the database server to hold the schema along with dimension and facts tables. To store the automatically generated database by the schema builder in the server, we deployed *Microsoft SQL server 2000*. This database becomes the source data for the cube creation process. The cube is constructed in the OLAP server. We deployed Microsoft *analysis service 2000* for the purpose of OLAP operations from the clustered database.

## D. OLAM cube construction

The fourth step of the implementation is the construction of OLAM cube. The database server holds the automatically generated schema of clustered data. The OLAP server connects with the database server to get clustered data from the database. Cube wizard (of the *Microsoft analysis services 2000* was used to construct the OLAM cube. It is an OLAM cube because it has the mined data aggregates in it. User can perform the standard OLAP operations on this OLAM cube.

#### E. Embedding Cube Visualization API

In the previous section, *MS Analysis services 2000* was introduced to serve the purpose of OLAP server and to construct OLAM cube using its cube wizard option. The emphasis in this paper from the beginning is to integrate enhanced OLAP system with data mining technique and automation in schema generation. After the elaboration of implementation of mining technique and the automation process of schema generation but at this point, we embedded the cube visualization API called *Dundas OLAP services* [71] which provides user-friendly and powerful visual data exploration controls.

Using the above mentioned controls, we developed another prototype to view the constructed OLAM cube. Figure 4 shows the user interface of our prototype which has all the above mentioned controls to work on cube data. It can be seen from the Figure 4 that the controls allow users to perform rich analysis tasks on the data. Furthermore, with the help of these controls, interactive visual exploration of data in the form of drill-down and roll up charts, grids and selection menus is possible.



Figure 4. User interface of prototype with all the visualization controls

#### VI. VALIDATION AND EVALUATION

In this section, we validated and evaluated the proposed architecture with the help of experiments. We used a real life data set (Census [73]) to perform experiments and discussed the results of these experiments. A comparative analysis of the proposed architecture with previous work is discussed at the end of this section. Such analysis highlights the benefits gained through the proposed integrated Online Analytical Mining (OLAM) architecture. The experimental details are presented in the following sub section.

### A. Experiment

In this experiment, we use US Census dataset [73], The data was collected as part of the 1990 census. There are 68 categorical attributes. This data set was derived from the USCensus1990raw data set [74]. The attributes list and the coding for the values is described below. Many of the less useful attributes in the original data set have been dropped, the few continuous variables have been discretized and the few discrete variables that have a large number of possible values. More specifically the USCensus1990 data set was obtained from the USCensus1990 data set by the following sequence of operations;

- **Randomization:** The order of the cases in the original USCensus1990raw data set was randomly permuted.

**Selection of attributes**: The 68 attributes included in the data set are given below. In the USCensus1990 data set we have added a single letter prefix to the original name.

We add the letter 'i' to indicate that the original attribute values are used and 'd' to indicate that original attribute values for each case have been mapped to new values.

Hierarchies of values are provided in the file USCensus1990raw.coding.htm [75] and the mapping functions used to transform the USCensus1990raw to the USCensus1990 data sets are giving in the file USCensus1990.mapping.sql [76]. The data is contained in a file called USCensus1990.data.txt [77].

In order to do the first step of this experiment, we passed this data set to the HCE tool so that we can get the hierarchical clusters from this huge data set. Similar to the previous experiment we set the clustering parameter for the hierarchical cluster generation process and at the top level we got five distinct clusters having individual hierarchies as depicted in Figure 5.

It can be seen from the Figure 5 that there are five clusters in the dendogram view of HCE tool. At the bottom of the hierarchical representation we can also see the data present in each cluster. The actual data can be extracted for each cluster as well. The next step of the experiment is to transform this hierarchically clustered data which will be fed into the developed prototype called the Automatic Schema Builder.

At this stage of the experiment, individual clusters data is being analyzed and each of the cluster is given a name which classify its characteristics. The data set has records of the persons, for that reason we classified the clusters on the basis of person's industry and occupation. The hierarchical clusters were named accordingly.



Figure 5. Hierarchical clusters of census dataset

The naming convention used is similar to the previous experiment such as the first letter of the industry or occupation has been used as an abbreviation. These abbreviations are represented in the Table 1.

This hierarchical information along with the clustered data has been fed into the automatic schema builder. The same procedure follows in the schema builder; it displays the cluster data and further asks for the type of schema to be generated. In contrast with the previous experiment we chose snowflake schema for generation. The automatic schema builder ran the same sequence of steps for snowflake schema generation explained in the snowflake schema generation algorithm.

Cluster Names	Clus_abb	Cluster Names	Clus_abb
Electric, Electronic and Computing (EEC)	C1	Businessmen (BM)	C3-1
Electric, Electronic and Computing (EEC)	C1	Labours and Workers (LW)	C3-2
Electric, Electronic and Computing (EEC)	C1	Labours and Workers (LW)	C3-2
Electric, Electronic and Computing (EEC)	C1	Labours and Workers (LW)	C3-2
Electric, Electronic and Computing (EEC)	C1	Copiers and Engineers (CE)	C4-1
Electric, Electronic and Computing (EEC)	C1	Copiers and Engineers (CE)	C4-1
Electric, Electronic and Computing (EEC)	C1	Publishers and Designers (PD)	C4-2
Agriculture, Foresty and Fishery (AFF)	C2	Publishers and Designers (PD)	C4-3
Agriculture, Foresty and Fishery (AFF)	C2	Constructors and Architects (CA)	C5-1
Agriculture, Foresty and Fishery (AFF)	C2	Constructors and Architects (CA)	C5-1
Rubber, Leather and Furniture (RLF)	C3	E.Enginers	C1-1-1
Rubber, Leather and Furniture (RLF)	C3	Elec.Enginers	C1-2-1
Rubber, Leather and Furniture (RLF)	C3	Sr.Teachers	C1-3-1
Rubber, Leather and Furniture (RLF)	C3	Jr.Teachers	C1-3-2
Printing, Publishing and Textile (PPT)	C4	Retailers	C1-4-1
Printing, Publishing and Textile (PPT)	C4	Salesmen	C1-4-2
Printing, Publishing and Textile (PPT)	C4	Graduates	C1-5-1
Printing, Publishing and Textile (PPT)	C4	Scientists	C2-1-1
Machinery and Construction (MC)	C5	Policy makers	C2-1-2
Machinery and Construction (MC)	C5	Analysts	C2-2-1
Engineers and Specialists (ES)	C1-1	Owners	C3-1-1
Administrators and Managers (AM)	C1-2	Carpentars	C3-2-1
Teachers and Trainers (TT)	C1-3	Swievers	C3-2-2
Teachers and Trainers (TT)	C1-3	Taylors	C3-2-3
Retailers and Salesmen (RS)	C1-4	Copier	C4-1-1
Retailers and Salesmen (RS)	C1-4	txl.Engineers	C4-1-2
Students and Learners(SL)	C1-5	Publishers	C4-2-1
Scientists and Doctors (SD)	C2-1	Designers	C4-3-1
Scientists and Doctors (SD)	C2-1	Labours	C5-1-1
Analysts and Policymakers (AP)	C2-2	Architects	C5-1-2

## TABLE 1 ABBREVIATION OF CENSUS DATA CLUSTERS

91

After generating the snowflake schema, the schema visualizer displays the diagrammatical view of the schema in the form of dimension and fact tables along with the relationships among the tables. The snowflake schema view, generated for the census data, is shown in Figure 6.

In addition to this visualization of the schema in the schema viewer, an automated database is also created in *MS SQL Server 2000*. We named the database as "*snowflake\_db*". This database has a single fact table and three dimension tables. The dimension tables are normalized to form the snowflake schema. Unlike the star schema, which was generated for the previous experiment, the snowflake schema has only one dimension table that is connected directly to the fact table. All other dimension tables are connected to form a normalize relationship.

After the database has been created, a separated OLAP database is created; this database will be connected to the database server to get the source data. The source data is the actual clustered database which is generated automatically. We named the OLAP database as *"Census\_OLAP\_DB"* and proceeded to the next step of this experiment which is the cube construction process.

The same cube construction process is followed as done in the previous experiment using the cube wizard of *MS Analysis Services*. The difference in this experiment is that we selected "no\_of\_persons" as a measure and defined the dimensional hierarchy in a different way. For the snowflake schema, we selected the 3 dimension tables and created a dimensional hierarchy.

The dimension table named "dim\_parent" is connected to the "dim\_child" dimension table which is further connected with the "dim\_child\_child" table. This way we can achieve a single dimensional view of the data in which we selected the hierarchy of that single dimension from all these tables that are connected in a normalized way. Snowflake schema has this advantage of selecting the levels of a dimension form various dimension tables. Therefore, customized levels for a single dimension were selected to build a "dim\_cluster" for the cube. The dimension created for the cube can be visualized at different levels using the dimension browser window.

After defining the dimensions and measures, the cube storage type has been set as MOLAP in the same way as done in the previous experiment using the storage design wizard. Finally, the cube with all these settings is processed and constructed. We visualize the cube data using the front-end prototype which has an embedded visualization enhancement API that the *Dundas OLAP services*. This front-end data visualization prototype connects with the OLAP server as explained in the previous experiment and displays the cube data in a rich graphical manner. In the end, we discuss some of the results achieved through this experiment. The main objective of the experiment was to emphasize the targeted and focused analysis. We explain this using a few problems.



Figure 6. Automatically generated snowflake schema

• **Problem** : Find out which individual industrial group of people that is dominant in the census dataset, furthermore identify the occupation group which has the largest value and state the number of people and the exact occupation title.

To solve this, we can easily identify the cluster which has the maximum number of people as shown in Figure 7. It is evident from the above figure that C1 cluster has the maximum number of records, so the focus for the analysis is only C1 Cluster. We visually explore C1 and the results of exploration are given in Figure 8.

Again, it is easily recognized that C1 has a specific sub cluster called "Teachers and Trainers (TT)" which has the maximum number of people in it. We further drill down into the sub-cluster TT to find out the exact occupation which has the maximum number of people. Figure 9 displays the drill down change after clicking the "+" sign on the chart.

It is evident from the drill down result that in the cluster named Teachers and Trainers (TT), *Sr.Teachers* are the maximum in number; there are 4112 senior teachers in the dataset. This was a problem which has been easily solved with quick visual exploration of data. In order to retrieve such information without the presence of integrated OLAM architecture, it would have taken a lot of time, firstly, to construct a query to retrieve such result. Secondly, the query has to be processed against the whole data set which is very huge, so it will increase the query response time as well.



Figure 7. Chart view showing total record in each cluster



Figure 8. Chart view of sub cluster C1



Figure 9. Drill down operation on sub cluster of C1

Thirdly, such rich visual exploration of data using charts and graphs is also not possible with simple querying and retrieval of data.

It can be seen from the results of the above two experiments that integrated OLAM architecture can greatly enhance the analytical power of the existing system. It not only automates the schema generation process but also integrates the enhanced OLAP functionalities with data mining techniques. When all these aspects are combined in an efficient way, the problem solving ability of the OLAM systems is increased significantly. In the next section, we present a comparison of the proposed OLAM architecture with the previous work done in the past regarding OLAM technology to highlight the contribution and the benefits achieved through the experimentation.

#### B. Comparison and discussion

In this section, we compare the proposed architecture with a number of other OLAM architectures in order to highlight the significance of this paper work and to discuss the achievements of this paper. The proposed integrated OLAM architecture consists of three major aspects which are 1) Data mining (hierarchical clustering), 2) Automatic Schema generation (clustered data) and 3) Enhanced OLAP (performance improvement & visualization enhancement). Table 2 depicts the major contribution in the area of OLAM.

 TABLE 2

 OLAM ARCHITECUTE COMPARISON TABLE

AUTHORS	WORK TITLE	Data Mining (Hierarchical Clustering)	Enhanced OLAP (Performance & Visualization)	Automatic Schema Generation	
Goil et al.	High performance OLAP and Data Mining on parallel computers	X	×	X	
Asghar et al.	Enhancing OLAP functionality using self-organizing neural networks	~	×	×	
Messaoud et al.	A new OLAP aggregation based on AHC technique		X	X	
J. Han	Towards online analytical mining in large databases	X	×	X	
Z. Hua	Online Analytical Mining of Association Rules	X	X	X	
Joseph et al.	Online Analytical Mining Web Pages-Tick sequence	X	X	X	
Dzeroski et al.	Using Data Mining and OLAP to Discover Patterns in a Database of Patients with Y-Chromosome Deletions	×	×	×	
Dehne et al.	Coarse grained parallel Online Analytical Processing for data mining	×	×	×	
PROPOSED WORK	Integrated OLAM for enhanced visualization and targeted analysis	-	~	~	

It is evident from the table below that the previous work was more intended towards typical OLAM as compared to the proposed work which gives direction towards an integrated OLAM system.

Therefore, the proposed work is significant as the integrated OLAM architecture can be easily implemented and used for targeted and focused analysis. Now, we discuss the major achievements gained through the architectural implementation. We start with data mining.

- Data mining (hierarchical clustering): We used the HCE tool for the hierarchical clustering of data. The major benefit of using this clustering tool is that it allows users to perform different types of hierarchical clustering. It has almost complete support for different kinds of hierarchical clustering algorithms. It means that at the mining layer, user can select the algorithm of his/her choice among all the available algorithms. Furthermore, the linkage method and clustering direction can also be given according to user preference. The accessibility of these clustering algorithms and parameters supports the OLAM concept as it allows selection of various mining algorithms for hierarchical clustering.
- Automatic schema generation: In addition to the above mentioned powerful capability of selecting the hierarchal clustering algorithm, the proposed architectural implementation makes it more constructive and dominant by reading the clustering results and automatically generating OLAP schema from it. The automatic schema builder allows three types of schema generation which include star, snowflake and galaxy. The

implementation provides direction for actually creating an automated database which not only holds the schema structure but also uploads the data from clustered results into specific dimension and fact tables. This is a very major achievement, as it reduces all the time and effort which is required to do the extract, transform and load (ETL) process in the data warehouse environment. It means that once you have generated a schema of a particular type then you do not have to worry about the data population process. The automatically generated database is always ready to become the data source for the cube construction phase in the OLAP server.

Enhanced OLAP: As the proposed work emphasize the integration of enhanced OLAP system in an integrated OLAM architectural implementation. We achieved the OLAP enhancement in terms of its performance and visualization. Firstly, discuss we the performance improvement achieved through the proposed work. We have used ForestCoverType dataset in the first experiment. But, to calculate the cube construction time, we created a manual star schema using SQL queries from the dataset without passing it to the hierarchical clustering tool. This non-clustered data set became the source for the cube in the OLAP server. We further constructed the *ForestCube* have 3 dimension tables and 1 fact table. The ForestCube process time has been noted and it was observed that it took 1 second to process 60180 non clustered rows of data to construct the cube completely. The process time and the rows calculated are shown in Figure 9.



Figure 10. Cube processing time of the data cube



Figure 11. Cube processing time and data size comparison graph

Similarly, we noted the cube construction time for the clustered *ForestCoverType* data, which came as a result, through hierarchical clustering tool, and found that this clustered data cube was instantly processed and its construction time was less than a second. Figure 10 shows the cube construction time comparison of both clustered and non clustered dataset.

It is evident from the time comparison graph that the cube processing time can be decreased by using the clustered data and the user can perform the targeted and fast multidimensional analysis on the clustered cube. Hence, it validates the proposed architecture for the performance improvement of OLAP data cube in terms of this reduced computational time. The second aspect of the enhanced OLAP is the visualization capability; Figure 11 shows the outputs of the *ForestCube* in various visual chart formats.

With the development of the front-end analysis prototype and embedding the OLAP data visualization controls of *Dundas Software*, we provided the OLAP user with a rich user interface to perform targeted and interactive visual exploration of the data. User can view the same data in a number of charts and graphs just by selecting the chart type from the toolbar present in the developed prototype.

It is clear from the above figure that the visualization of cube data has been enhanced to facilitate the analyst to visually explore the hierarchical data and to perform the basic OLAP operation efficiently and quickly.

In this section, we presented the experiments done on two real life data sets. Furthermore, we discussed the experimental results and validated the proposed architecture by evaluating the results. Finally, we addressed the aims and objectives which are achieved through the proposed integrated OLAM architecture implementation. Experimental results prove that the proposed work in this paper intensify the focused and targeted analysis of data and supports the overall framework of OLAM in a very significant and efficient way.



Figure 12. Preview of various chart types in the developed prototype

#### VII. CONCLUSION AND FUTURE WORK

The research conducted in this paper was motivated by two observations. At one side, OLAM (combination of OLAP and Data mining), is gaining momentum as an imminent technology for decision support. On the other hand, integration of enhanced OLAP systems with any data mining technique is missing. This gap of integration limits the analysis capabilities of the standard OLAM architecture. The proposed work presents an attempt to reduce the gap of integration by proposing an integrated OLAM architecture. In addition to reduce the gap, the proposed architecture also allows automation in the process of OLAP schema generation. The purpose of this addition is to intensify the proposed architecture to deal with the new challenges imposed by emerging decision support applications.

The results presented in this paper improve the existing OLAM architectures by overcoming the existing limitations and introducing the automatic schema generation layer in it. Due to the limited timeframe of our work we deliberately excluded some properties of data mining techniques as they would have expanded the scope of the resulting architecture. Our work can be continued in research areas, such as integration of OLAP and data mining, automatic schema generation, enhanced OLAM systems or by intensifying the areas central to this paper.

Firstly, we have only considered a particular data mining technique which is hierarchical clustering. Future work can be the use of other mining techniques such as association rule mining, classification, predication and others with enhanced OLAP systems to benefit the existing OLAM architecture. Secondly, the automatic schema generation layer in our architectures is only useful for the hierarchal clustering results which have a parent-child relationship to generate schemas of different types. This area can also be improved for the different types of mined results as different data mining techniques produce different mining results. We are exploring other data mining techniques that can also be integrated with the enhanced OLAP system and also the mined results can be used to automatically build schema of different types. Another direction for future work is that we intend to enhance visualization capabilities of the hierarchical clustering results so that the clusters should be named properly for identification purpose.

#### REFERENCES

- [1]. S. Chaudhuri and U. Dayal, An overview of data warehousing and OLAP technology, *ACM SIGMOD Record*, Vol. 26 (1997), pp. 65-74.
- [2]. A. Cuzzocrea, D. Sacca and P. Serafino, A hierarchy driven compression technique for advanced OLAP visualization of multidimensional data cubes, in Proc. of 8th Int'l Conf. on Data Warehousing and Knowledge Discovery (DaWak), (Springer Verlag 2006), pp. 106-119

- [3]. S. Maniatis, P. Vassiliadis, S. Skiadopoulos and Y. Vassiliou, Advanced visualization for OLAP, in Proc. of 6th ACM Int'l Workshop on Data Warehousing and OLAP (DOLAP), (ACM New York, 2003), pp. 9-16.
- [4]. S. Mansmann and M. Scholl, Exploring OLAP aggregates with hierarchical visualization techniques, in Proc. of ACM Symposium on Applied Computing (2007), pp. 1067-1073.
- [5]. S. Goil and A. Choudhary, "High performance OLAP and data mining on parallel computers," Data Mining and Knowledge Discovery, vol. 1, no. 4, pp. 391-417, Dec. 1997.
- [6]. D. Papadias, P. Kalnis, J. Zhang and Y. Tao, "Efficient OLAP operations in spatial data warehouse," in Proc. of the 7th Int'l Symposium on Advances in Spatial and Temporal Databases (SSTD), Springer-Verlag, 2001, pp. 443-459
- [7]. S. Asghar, D. Alahakoon and A. Hsu, "Enhancing OLAP functionality using self-organizing neural networks," Neural, Parallel and Scientific Computations, vol. 12, no. 1, pp. 1-20, March 2004.
- [8]. R. B. Messaoud, O. Boussaid and S. Rabaseda, "A new OLAP aggregation based on the AHC technique," in Proc. of the 7th ACM Int'l Workshop on Data Warehousing and OLAP (DOLAP), ACM New York, 2004, pp. 65-72.
- [9]. S. Maniatis, P. Vassiliadis, S. Skiadopoulos and Y. Vassiliou, "Advanced visualization for OLAP," in Proc. of the 6th ACM Int'l Workshop on Data Warehousing and OLAP (DOLAP), ACM New York, 2003, pp. 9-16.
- [10].M. Sifer, "A visual interface technique for exploring OLAP data with coordinated dimension hierarchiches," in Proc. of the 12th ACM Int'l Conf. on Information and Knowledge Management (CIKM), ACM New York, 2003, pp. 532-535.
- [11].A. Voss, V. Hernandez, H. Voss and S. Scheider, "Interactive visual exploration of multidimensional data: requirements for common GIS with OLAP," in Proc. of the 15th Int'l Workshop on Database and Expert Systems Applications (DEXA), 2004, pp. 883-887.
- [12].M. Scotch and B. Paramanto, "SOVAT: Spatial OLAP visualization and analysis tool," in Proc. of the 38th Annual Hawaii Int'l Conf. on Systems Sciences (HICSS), 2005, p. 165.
- [13].A. Cuzzocrea, D. Sacca and P. Serafino, "A hierarchy driven compression technique for advanced OLAP visualization of multidimensional data cubes," in Proc. of the 8th Int'l Conf. on Data Warehousing and Knowledge Discovery (DaWak), Springer Verlag, 2006, pp. 106-119.
- [14].K. Techapichetvanich and A. Datta, "Interactive visualization for OLAP," in *Int'l Conf. on Computational Science and its Applications* (ICCSA), 2005, pp. 206-214.
- [15].S. Mansmann and M.Scholl, "Extending visual OLAP for handling irregular dimensional hierarchies," in Proc. of 8th Int'l Conf. on Data Warehousing and Knowledge Discovery (DaWaK'06), 2006, pp. 95-105.
- [16].S. Mansmann and M. Scholl, "Visual OLAP: A new paradigm for exploring multidimensional aggregates," in Proc. of IADIS Int'l Conf. on Computer Graphics and Visualization (CGV), 2008, pp. 59-66.
- [17].S. Asghar and M. Usman, "Enhancing OLAP performance and visualization," in *Proc. of 6th Int'l Conf. on*

Information Science Technology and Management (CISTM), 2008, pp. 59-70.

- [18].Fayyad U.M, Piatesky-Shapiro G., Smyth p. and Uthurusamy R., "From data mining to knowledge discovery: An overview,"in *Proc. of Advances in data mining and knowledge discovery*, MIT Press, 1996, pp. 1-34.
- [19].J. Hipp, U. Guentzer and G. Nakhaeizadeh, "Algorithms for association ruling mining – a general survey and comparison," ACM SIGKDD Explorations Newsletter, vol. 2, no. 1, pp. 58 – 64, June 2000.
- [20].R. C. Holte, "Very simple classification rules perform well on most commonly used datasets," *Machine Learning*, vol. 11, no. 1, pp. 63-90, April 1993.
- [21].A. K. Jain, M. N. Murty and P. J. Flynn, "Data clustering: A review," ACM Computing surveys (CSUR), vol. 31, no. 3, pp. 264-323, Sep. 1999.
- [22].M. J. Shaw, C. Subramaniam, G. W. Tan and M. E. Welge, "Knowledge management and data mining for marketing," *Decision Support Systems*, vol. 31, no. 1, pp. 127-137, May 2001.
- [23].J. Han, "Towards online analytical mining in large databases," ACM SIGMOD Record, vol. 27, no. 1, pp. 97-107, March 1998.
- [24].J. Han, S. H. S. Chee and J. Y. Chiang, "Issues for online analytical mining of data warehouses," in *Proc. of the SIGMOND Workshop on Research Issues on Data Mining and Knowledge Discovery* (DMKD), Seattle, 1998, pp. 2:1-2:5.
- [25].H. Zhu, "Online analytical mining of association rules," Master Thesis, Simon Fraser University, 1998, pp. 1-117.
- [26].J. Fong, H. K. Wong and A. Fong, "Online analytical mining Web-pages tick sequences," J. of Data Warehousing, vol. 5, no. 4, pp. 59-67, 2000.
- [27].S. Dzeroski, D. Hristovski and B. Peterlin, "Using data mining and OLAP to discover patterns in a database of patients with Y chromosome deletions," in *Proc. AMIA Symp.*, 2000, pp. 215–219.
- [28].F. Dehne, T. Eavis and A. Rau-Chaplin, "Coarse grained parallel on-line analytical processin (OLAP) for data mining, in *Proc. of the Int'l Conf. on Computational Science* (ICCS), 2001, 589-598.
- [29].V. Markl, F. Ramasak and R. Bayer, "Improving OLAP performance by multi-dimensional hierarchical clustering," in *Proc. of the 1999 Int'l Symposium on Database Engineering and Applications* (IDEAS), 1999, p. 165.
- [30]. V. Markl and R. Bayer, "Processing relational OLAP queries with UB-trees and multidimensional hierarchical clustering," in *Proc. of the Int'l. Workshop on Design and Management of Data Warehouses* (DMDW), 2000, pp. 1-10.
- [31].N. Karayannidis, T. Sellis and Y. Kouvaras, "CUBE file: A file structure for hierarchically clustered OLAP cubes," *Advances in Database Technology*, LNCS, Springer Verlag, Berlin-Heidelberg, pp. 621-638, 2004.
- [32].D. Theodoratos and A. Tsois, "Heuristic optimization of OLAP queries in multidimensionally hierarchically clustered databases," in *Proc. of the 4th ACM Int'l Workshop on Data Warehousing and OLAP* (DOLAP), ACM New York, 2001, pp. 48-55.
- [33].K. Hann, C. Sapia and M. Balaschka, "Automatically generating OLAP schemata from conceptual graphical models," in *Proc. of the 3<sup>rd</sup> ACM Int'l. Workshop on*

Data Warehousing and OLAP (DOLAP), ACM New York, 2000, pp. 9-16.

- [34]. V. Peralta, A. Marotta and R. Ruggia, "Towards the automation of data warehouse design," Technical Report TR-03-09, InCo, Universidad de la República, Montevideo, Uruguay, June 2003.
- [35].N. Tryfona, F. Busborg and J. G. B. Christiansen "StarER: A conceptual model for data warehouse design," *Proc. of the 2<sup>nd</sup> ACM Int'l. Workshop on Data Warehousing and OLAP (DOLAP)*, ACM New York, 1999, pp. 3-8.
- [36]. Y. Song, et. al., "SAMSTAR: An automatic tool for generating star schemas from an entity-relationship diagram," in *Proc. of the 27th Int'l. Conf. on Conceptual Modeling*, LNCS 2008,vol. 5231 pp. 522-523.
- [37].S. R. Maddi and V. Khan, "Comparative analysis of on-line analytical processing tools," University essay from IT-universitetet I Göteborg, Sweden 2007.
- [38].M. Reed, "A definition of data warehousing," [Onlline]. Available: http://www.intranetjournal.com/features/datawarehou sing.html. [Accessed: April 28, 2009].
- [39].R. Kimball and M. Ross, *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*, 2<sup>nd</sup> ed., New York: John Wiley, 1996.
- [40].J. Widom, "Research problems in data warehousing," in Proc. of the 4<sup>th</sup> Intl. Conf. on Information and Knowledge Management (CIKM), 1995, pp. 25-30.
- [41].S. Chaudhuri and U. Dayal, "Data warehousing and OLAP for decision support," in *Proc. of the ACM Intl. Conf. on Management of Data*, ACM New York, 1997, pp. 507-508.
- [42].G. Colliat, "OLAP, relational, and multidimensional database systems," ACM SIGMOND Record, vol. 25, no. 3, pp. 64-69, 1996.
- [43].H. Gupta, V. Harinarayan, A. Rajaraman and J. D. Ullman, "Index selection for OLAP," in *Proc. of the* 13<sup>th</sup> Intl. Conf. on Data Engineering (ICDE), 1997, pp. 208-219.
- [44]. W. H. Inmon, "The data warehouse and data mining," *Communications of the ACM*, vol. 39, no. 11, pp. 49-50, 1996.
- [45]. ADSS OLAP Help. [Online]. Available: http://www.agroict-olap.org/OLAP\_Help.aspx. [Accessed: April 28, 2009].
- [46]. T. J. Teorey, "Data warehousing, OLAP, and data mining, in *Database Modeling and Design*, 3<sup>rd</sup> ed., Morgan Kaufmann, 1999, pp. 211-250.
- [47]. IBM Journal of Research and Development. [Online]. Available: https://www.research.ibm.com/journal/sj/414/colossi.

https://www.research.toni.com/journal/s//414/colossi. html. [Accessed: April 28, 2009].

- [48].E. Thomsen, "OLAP Solutions: Building Multidimensional Information Systems," 2<sup>nd</sup>. New York: John Wiley, 2002.
- [49].B. Dinter and C. Sapia et. al., "OLAP marketing and research: Initiating the cooperation," *J. of Computer Science and Information Management*, vol. 2, no. 3, 1999.
- [50]. M S. Chen, J. Han and P. S. Yu, "Data mining: An overview from a database perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 866-883, Dec. 1996.
- [51].U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, "Knowledge discovery and data mining: Towards a unifying framework," in *Proc. of the 2<sup>nd</sup> Intl. Conf. on*

*Knowledge Discovery and Data Mining*, 1996, pp. 82-88.

- [52].D. Hand, H. Mannila and P. Smyth, "Principles of Data Mining," [Online]. Available: http://cat.inist.fr/?aModele=afficheN&cpsidt=189586 26. [Accessed: April 28, 2009].
- [53].U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, 1996.
- [54].P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data*, Springer Berlin-Heidelberg, 2006, pp. 25-71.
- [55].A. Berson, S. Smith and K. Thearling, "An Overview of Data Mining Techniques," [Online]. Available: http://www.thearling.com/text/dmtechniques/dmtechn iques.htm. [Accessed: April 28, 2009].
- [56].D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," in *Proc. of* the 6<sup>th</sup> Intl. Conf. on Knowledge Discovery and Data Mining ACM SIGKDD, 2000, pp. 407-416.
- [57].S. O. Deininger, et. al., "MALDI imaging combined with hierarchical clustering as a new tool for the interpretation of complex human cancers," J. Proteome Res., vol. 7, no. 12, pp. 5230-5236, 2008.
- [58].C. Li and W. H. Wong, "DNA-Chip Analyzer (dChip)," [Online]. Available: http://biosun1.harvard.edu/complab/dchip/Li%2003.p df. [Accessed: April 28, 2009].
- [59]. V. Hatzivassiloglou, et. al., "SIMFINDER: A flexible clustering tool for summarization," in NAACL Workshop on Automatic Summarization, 2001, pp. 41-49.
- [60].J. Seo, et. al., "Interactive color mosaic and dendrogram displays for signal/noise optimization in microarray data analysis," in *Proc. of the Intl. Conf. on Multimedia and Expo-Volume 3*, 2003, pp. 461-464.
- [61].R. Rak and W. Stach, "Evaluation of XLMiner demo version 2," report, 2004. [Online]. Available: http://www.cs.ualberta.ca/~zaiane/courses/cmput695-04/work/A2-reports/xlminer.pdf. [Accessed: April 28, 2009].
- [62].[Online]. Available: http://www.cs.umd.edu/hcil/hce/. [Accessed: April 28, 2009].
- [63].Netz, S. Chaudhuri, J. Bernhardt and U. Fayyad, "Integration of data mining and relational databases," in *Proc. of the 26<sup>th</sup> Intl. Conf. on Very Large Database*, 2000, pp. 719-722.
- [64].[Online]. Available:http://www.ibm.com/developerworks/ration al/library/07/0626\_johnson/index.html. [Accessed: April 28, 2009].
- [65].D. L. K. Johnson, "Reporting with rational portfolio manager version 7.1," June 26, 2007. [Online]. Available: http://www.ibm.com/developerworks/rational/library/ 07/0626\_johnson/index.html [Accessed: April 28, 2009].
- [66].Stanford Technology Group, "Designing the Data Warehouse on Relational Databases," 1996. [Online]. Available: http://www.informix.com.ua/articles/rolap/rolap.htm.

[Accessed: April 28, 2009].

[67].D. L. Moody and M. A.R. Kortink, "From enterprise models to dimensional models: A methodology for data warehouse and data mart design," in *Proc. of the*  Intl. Workshop on Design and Management of Data Warehouses (DMDW), 2000, pp. 5:1-5:12.

- [68]. "Data Warehouse Design," [Online]. Available: http://www.pafis.shh.fi/%7Eargarg03/SFIS/WS1/data w\_design.htm. [Accessed: April 28, 2009].
- [69].S. Asghar, D. Alahakoon and D. Taniar, "Enhancing OLAP functionality using association rules," in *Proc.* of 2nd Int'l Conf. on Computational Intelligence, Robotics and Autonomous Systems (CIRAS), 2003, pp. 1-6.
- [70].A. Rauber and P. Tomsich, "An architecture for modular OLAP Systems: Supporting distributed and parallel query processing using cooperating CORBA objects," in 10th Int'l Workshop on Database and Expert Systems Applications (DEXA), 1999, p. 45.
- [71].S. Asghar and M. Usman, "Integrated enhancement of OLAP performance and visualization using self growing organizing neural networks," in *Int'l J. of Information Tech. and Decision Making*, 2009, submitted.
- [72]. Available:http://support.dundas.com/OnlineDocument ation/WebOLAP/Overview%20of%20Dundas%20OL AP%20Architecture.html. [Online]. [Accessed: Apr. 2009].
- [73].J. A. Blackard, D. J. Dean and C. W. Anderson, Forest cover type, *The UCI KDD Archive* [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science (1998).
- [74].C. Meek, B. Thiesson, D.Heckerman,, US Census Data (1990), The UCI KDD Archive [http://archive.ics.uci.edu/ml/datasets/US+Census+Da ta 281990 29], US Department of Commerce.
- [75]. Available. http://archive.ics.uci.edu/ml/databases/census1990/US Census1990raw.html. Online [Accessed: Apr. 2009].
- [76]. Available. http://archive.ics.uci.edu/ml/databases/census1990/US Census1990raw.coding.htm Online [Accessed: Apr. 2009].
- [77]. Available.

http://archive.ics.uci.edu/ml/databases/census1990/US Census1990.mapping.sql Online [Accessed: Apr. 2009]



**Muhammad Usman** received BS computer science degree from Comsats institute of information technology, Pakistan in 2006 and MS software engineering degree from Shaheed Zulfikar Ali Bhutto institute of science and technology, Pakistan in 2009.

He has worked as a Data warehouse developer in the center of Agro-(CAID) Paligitzer He has multiched 5

informatics research (CAIR), Pakistan. He has published 5 conference paper and 1 journal article. He is currently researching in data mining, online analytical processing, decision support systems, knowledge discovery, neural networking, pattern recognition and soft computing.

Mr. Usman is a recipient of National Agriculture and IT advancement award in recognition of his work in Agriculture decision support system from FAST national university, Islamabad, Pakistan. Now he is studying a PhD program at Auckland University of Technology, New Zealand.



**Sohail Asghar** is an Associate Professor of Computing, Faculty of Computer Science, Mohammad Ali Jinnah University, Islamabad, Pakistan. In 1994, he graduated with honors in Computer Science from the University of Wales, United Kingdom. From 1994 to 2002, he worked as a Senior Software Engineer in a software company in Islamabad. He

then obtained his PhD in Information Technology at Monash University, Melbourne Australia 2006. in He has taught and researched in the Data Mining and Decision Support Systems areas, and he has published extensively in both Australian and international journals as well as conferences proceedings. He has also consulted widely on information Technology matters, especially within the framework of disaster management systems. In 2004 he won the Australian Post Award Graduate for Industry. Dr. Asghar is a member of, the Australian Computer Society (ACS), and IEEE.