OLAP OF THE FUTURE

A Min Tjoa[1)], Roland Wagner[2)],
Philipp Tomsich[1)], Andreas Rauber[1)]

Abstract

On-line analytical processing (OLAP) is a powerful data analysis technology. However, current implementations bypass the 3-layer model of databases for performance reasons and create a redundant database

The clear separation between external views, conceptional model and internal physical representations in the three layer database architecture constitutes one of the key factors to the success of database technology for on-line transaction processing (OLTP). However, experience has shown that it is not feasible to construct an on-line analytical processing (OLAP) view onto the data within this model. As a compromise, materialized views were introduced. Yet, these limit the applicability of OLAP to fairly standard applications. Some application domains such as the analysis of stock market data or telecommunication systems with a diverse structure and unpredictable aggregation and query patterns can not be timely handled using pre-materialization. As a result, the benefits of OLAP are not available when building analysis and visualization tool for this kind of applications.

We envision OLAP-on-demand as a foundation for the re-unification of the transaction processing and analysis in databases. Instead of pre-materializing expected queries, analysts can integrate the available data sources on the fly where the analysis of information is not delayed by certain ``update windows''. Although current computing hardware does not provide the necessary data throughput, we expect future generation systems to cope with the high demands of this approach. In the meantime, parallel and distributed query processing can be used to achieve the required response times.

As an ultimate goal the realisation of Codd's formulaic model is envisioned.

[1)]Vienna University of Technology, Favoritenstraße 9–11/188, A–1040 Wien, Austria
[2)] Johannes Kepler University Linz, Altenbergerstraße 69, A–4040 Linz, Austria

1 Introduction

On-line analytical processing (OLAP) is a powerful data analysis technology. However, current implementations bypass the 3-layer model of databases for performance reasons and create a redundant database. In an extreme sense, we may consider OLAP in its current form to be harmful, as it increases the complexity of implementing and maintaining data analysis solutions, although it should simplify this. However, we expect future improvements to the implementation techniques and the increased performance of future generation computing hardware to reconcile the transaction processing and analysis components in databases. Instead of pre-materializing expected queries, analysts will integrate the available data sources on the fly such that the result of their analysis is not based on outdated information. Although current computing hardware does not provide the necessary data throughput, we expect future generation systems to cope with the high demands of such implementations. In the meantime, parallel and distributed query processing can be used to achieve the required response times. This will eventually turn OLAP into a paradigm for the formulation of all data manipulation and data analysis task. With the addition of a rule-language, even real-time control application could be expressed in the terms of multi-dimensional data transformation operators.

Section 2 describes the basic principles of OLAP. Section 3 discusses in some detail how the current implementations of OLAP systems contradict the principles of database systems. OLAP on-demand, the expected reconciliation of data management and analysis is put forth.

Section 4 discusses the bottlenecks leading to the compromises taken in current applications and presents methods expected to overcome these bottlenecks. A conclusion is given in Section 5, which recognizes the full potential of OLAP.

2 On-Line Analytical Processing

After the publication of the relational model by E.F.Codd [9] and the introduction of the three-layer concept by the ANSI Study Group on Database Systems [6, 26], databases became a ubiquitous technology in transaction processing systems. The key factors to this success were the flexibility to define multiple views onto the same data, the integration of data in a central database and the independence of the physical implementation for the conceptual model. With these prerequisites, increasingly powerful on-line transactional processing (OLTP) systems appeared.

The success of database technology has led to a rapid growth of data stored in databases, accelerated by the increasing volumes of electronic commerce. Today, large organizations are manipulating data in the range of terabytes with petabytes around the corner. With availability of such large amounts of data a sophisticated and intuitive data analysis solutions become a necessity. An emphasis was put on the development of on-line analytical processing (OLAP) [1, 7, 8, 10] tools to empower the user-analyst to extract useful information without experiencing the complexities of the underlying infrastructure. With the diverse query and analysis tools, one of the pronounced goals of OLAP was a mediator-role for data access and data analysis. It provides a set of intuitively understood, multidimensional data access and transformation operators as a foundation for the navigation and analysis of data. OLAP is changing the way end-user's interface with databases much in the same way as relational database technology replaced the complex, heterogeneous pre-relational systems by introducing a homogeneous interface using the relational language to form ad-hoc-queries. Conceptionally an evolutionary improvement to the external views of the three layer architecture, OLAP could turn into the data access and manipulation paradigm of the 21st century.

Examples for necessity and power of OLAP range from scientific applications to the analysis of business data. Data Warehousing can be regarded as a continuation of statistical databases used in the domain of life sciences and medicine by Vinek as one of the pioneers in the area [27].

However, OLAP has one, undeniable drawback: it cannot be represented as an external view in existing databases [1, 10, 13, 14]. The reason for this lies in the purpose and design of database management systems, which were never intended to provide the sophisticated data synthesis and analysis functions necessary. These functions were always intended to be provided by separate end-user tools. *This contradicts the paradigm of the three layer database architecture*. In his data modelling book [29] one can follow his main intend to defend the 3-layer architecture where the Conceptual Model represents the core of every software or database implementation. With the growing complexity and volumes of data to be stored and analysed the necessity of a proper semantic modelling of the conceptual model becomes more and more evident. The ontological aspects of the conceptual modelling were always the core of database modelling. The recent debate on the ontological foundations of conceptual modelling [32, 33, 37] attests the necessity of a precise mapping of terminology concepts. Günther Vinek has always paid huge attention on this issue by precisely defining questions like “what is a customer of a bank”, “what is an agricultural business entity-what if this entity is located on the border of 2 provinces and covering an area on both provinces – a real world question for the design of the LFBIS-data model of the Austrian Federal Ministry for Agriculture”.

Nowadays the US-retailer Wal-Mart operates one of the world’s largest data warehouses and conducts OLAP-based analysis on it, the European particle research lab CERN analyses the data gathered during its collider experiments using OLAP technology. In the latter case, terabytes to petabytes of data will be acquired. The activities of these tools against the enterprise database constitute a transaction. However, these transactions usually take multiple times that of their OLTP counterparts. While OLTP transactions are simple insertions or updates which execute instantaneously, OLAP transactions may require minutes or even hours to complete. Due to the availability requirements of operational systems, queries cannot be executed using the main database. As a result, OLAP systems compromise in their adherence to the 3-layer model to improve their query response times by duplicating the data in a specialized database, which includes synthesized summary information.

The core functionality of OLAP stems from its ability to dynamically and flexibly aggregate information, providing summary information. This is implemented within the conceptual framework of a multidimensional data space, where information can be viewed and analyzed according to multiple dimensions.

The consolidation of the data has been characterized as “the process of synthesizing pieces of information into single blocks of essential knowledge” [10]. However, this simultaneous analysis of multiple data dimensions may lead to arbitrarily complex aggregation functions. As data sets may be aggregated in a vast number of ways, with every consolidation path reflecting a unique perspective, the queries posed by multiple users are non-deterministic, as every user places a different emphasis according to his or her background. In addition, the required consolidation paths evolve over time with the needs of the users. In contrast, the design of a database implies certain, preferred consolidation paths which dictate the physical design of the database and heavily influence the efficiency of the database. Moreover, the definition of consolidation paths limits the analyst's choices of consolidation paths, reducing the extended flexibility offered by the tool.

In order to achieve interactive response time to queries, pre-materialization of aggregated data is used to construct an independent, redundant database from de-normalized data [13, 14]. However, as not all consolidation paths can be anticipated, the data can only be navigated interactively along predefined paths, which severely limit the flexibility of OLAP and contradict its original goal. Another major drawback resulting from the pre-materialization of aggregated values is the emergence of a so-called "update window". This is the time delay between updates to the operational database and the next time the analytical database is rebuilt. Admittedly, the average analytical applications implemented using OLAP operates exclusively on historic data. However, a wide variety of application domains requires up-to-date information and exhibit unpredictable aggregation and query patterns.

Financial information processing, the telecommunications sector and electronic procurement applications are just a few examples for this. On top of these limitations, the fact that OLAP tools interact with an analytical database gives them a read-only semantics. In order to modify the operational database, conventional tools need to be employed.

While the basic concepts of OLAP put it to the forefront of database access technologies, unifying data management, access and analysis, integrating legacy databases transparently, the inadequacies of current implementations keep it from fulfilling its full potential of providing a common framework for the implementation of data access and analysis tools. The necessity to bypass the 3-layer architecture to provide satisfactory performance and the dependence on redundant, de-normalized data eliminates many of the benefits gained from the the relational model. Pre-materialization introduces undue delays until data updates propagate onto the user's desktop. Additionally, it relies on the definition of "typical queries", which are expressed in terms of the consolidation paths that are materialized. If the user is able to pose arbitrary queries, the timeliness of an answer cannot be guaranteed. The duplication of databases increases the maintenance cost. As a final consequence, the scalability of OLAP systems suffers, as the number of possible consolidation paths expands exponentially, as the structure of the source data increases in complexity.

3 OLAP and the 3-layer database reference model

E.F.Codd already wrote in his introduction of OLAP, that "the need which exists is not for yet another database technology, but rather for robust data analysis tools which complement the existing data management system" [10]. It is to be expected, that the continuing growth of database systems will eventually lead to situations where the limited scalability and the increasing update windows will eventually lead to a rethinking of the attitude towards OLAP. In the long run, a data analysis solution can only succeed if it integrates well with the data management solution.

In order to re-conciliate OLAP with the available tools, it has to be fit into the 3-layer model on the same level as external views: **OLAP is just another view.**

Conventional wisdom 1: Preaggregations are the foundation of OLAP.

In order to achieve interactive response time to queries, pre-materialization of aggregated data is used to construct an independent, redundant database from denormalized data [13, 14]. Pre-materialization attempts to do as much work in advance as possible. All data sources are integrated into a single database during the pre-materialization. Aggregated values are then added to the resulting database, creating an analytical database optimized for OLAP. As long as no unanticipated queries are posed to a system that uses pre-materialization, instantaneous results are guaranteed.

Proposition 1: Aggregations can be done on the fly.

The OLAP capabilities have to be provided on-the-demand in such a system. Since multiple data source may need to be integrated to successfully analyze data, a multi-database system will result. An aggregation engine in such a system needs to process the queries by retrieving the raw data from the available sources and integrate them dynamically, thus actually creating virtual data cubes.

The benefits of using virtual data-cubes are apparent. The elimination of redundant storage structures simplifies the setup and maintenance, while the lack of a dedicated feeding and pre-materialization phase eliminates the update window. Queries are always evaluated against the source databases. In addition, the scalability of the system is improved, as the total amount of processing is reduced. Only those aggregates are computed that are actually needed. This will dramatically improve the capability to operate on very large data sets, where only small subsets are ever used during analysis.

Current computing systems lack the I/O bandwidth and processing power to perform OLAP-on-demand, except for trivial cases. Yet, it may be expected that future hardware generations will soon satisfy the performance requirements. However, even with current computing system real on-the-fly aggregation can be provided as outlined in Section 4.

Conventional wisdom 2: Intelligence resides in the feeding layer.

Today, a great part of the intelligence of an OLAP system can be found in the feeding and cleaning process of the data, as well as in its de-normalization. After these steps are completed, the structure of the source data does not play a role. The de-normalization improves the browsing performance, as no indirections are required to access the data. In exchange for the performance gains, updates to the data become very costly operations. This is one of the reasons for the read-only semantics of the analytical databases that are created during pre-materialization.

Proposition 2: Metadata-driven integration and cleansing.

This will have to change, when multiple data sources are to be considered during each query, and the metadata will be alleviated to the level of a data transformation language [15]. One of Vinek’s foci of scientific work is the statistical analysis of huge amount of measurement and experimental data. This includes data cleansing and plausibility control in a generic way as it is preformed by the ETL process. The NUMAN language introduced for this purpose can still be regarded as pioneering work because of its general coverage [28, 30].

This puts OLAP into the position of transforming legacy data into a multi-dimensional representation using one level of metadata and operating on the resulting data cubes using a second level of metadata. The original definition of OLAP as a data access, transformation and analysis language will finally become a reality.

Unlike in current OLAP systems, metadata will play a paramount role in systems that integrate data on-demand into virtual data cubes.

Conventional wisdom 3: Update windows are a necessary evil.

A major drawback resulting from the pre-materialization of aggregated values, is the emergence of a so-called “update window”. Both de-normalization and pre-materialization contradict the principles of database systems, as a second database is created. Consequently, the end-user applications do not operate on a central data repository anymore. As a consequence, OLAP-based tools cannot directly modify the database. Although they provide a transparent interface to multiple, integrated data sources, this interface can be used for viewing only. The use of pre-materialization also causes an update window which bars the use of OLAP in applications that require accurate and up-to-date results. Due to the high cost of pre-materializing the required aggregates, updates to the analytical database will usually occur outside of business hours, creating an update window of at least a full workday with update windows of a week not being uncommon. Even if versioning or similar techniques are employed, a considerable update window remains.

Proposition 3: OLAP data is not outdated.

Real-time and near-time data analysis are a necessity in a wide variety of applications, ranging from decision support systems which analyze stock market data to emerging e-commerce application.

E-commerce applications, that perform up-to-date pricing, deal with perishable goods, or deal with offers that have a limited lifespan need short update windows. Additionally, customer relationship applications require up-to-date information to avoid embarrassing miscommunication. However, as decisions have to be based on an up-to-date view of a dynamically changing marketplace. We thus need to provide an up-to-date view of the underlying data [21].

Conventional wisdom 4: Limited freedom is the price for performance.

The design of a database implies certain, preferred consolidation paths which dictate the physical design of the database and heavily influence the efficiency of the database. Moreover, the definition of consolidation paths limits the analyst’s choices of consolidation paths, reducing the extend of flexibility offered by the tool. Thus the promise of flexible querying using ad-hoc consolidation is true only within the framework of predefined consolidation paths.

Proposition 4: OLAP can provide absolute freedom.

Integrating data sources on-the-fly into virtual data cubes actually provides the freedom promised by the definition of OLAP. The user is no longer restricted by the framework implemented by the database designer, but can freely navigate through the entire data.

The de-normalization of the stored data is another trade-off between space and performance. While accessing data from the database is considerably simplified, the resulting database is highly redundant and far harder to modify.
In summary, the current state of OLAP is such that the promise of providing a universal data access paradigm cannot be fulfilled, if the use of pre-materialized and de-normalized databases remains dominant. As long, as OLAP requires the bypassing of proven database principles and creates additional hard-to-maintain and badly scalable databases, it will remain a limited to a small number of application areas.

4 Approaching a solution

The implementation of OLAP-based interactive data analysis for large data sets poses a number of challenges resulting from the overheads incurred when moving large amounts of raw data and evaluating the aggregation functions defined on the fly. Even when the possible aggregations are limited to trivial functions such as summation and averaging, the sheer number of basic records that need to be processed and the sheer number of possible aggregation paths makes the implementation of OLAP tools an intimidating prospect. However, emerging technologies promise to overcome these problems.

Conventional wisdom 5: Pre-aggregation is a prerequisite for interactive response times.

As technology intended for interactive data exploration, timely responses are a critical success factor for the success of any OLAP implementation. However, the performance requirements for OLAP on demand exceed the performance characteristics of modern systems. This leads to the question, whether advanced data analysis is really limited by hardware performance and whether an integration of OLAP into the 3-layer architecture is feasible at all. The main problem is posed by the computations necessary in the synthesis of aggregated values.

Proposition 5: Parallel computing may provide one solution.

Parallel and distributed computing offer an intermediate solution to achieve acceptable performance by using the virtual cube concept.. Together with techniques similar to those used for instruction scheduling in modern compilers, parallel I/O and cluster computing may overcome the major bottlenecks.

However, the use of parallelism does not offer sufficient guarantees that queries can be answers in a timely fashion, particularly in the presence of multiple users or frequent updates to the source data.
Additional optimizations have to be employed in such an environment.

Evaluating queries in OLAP systems fortunately involves a considerable amount of parallelism.
All dimensions of a data cube are independent of each other in OLAP. Since all data can be indexed through these, they perform a similar function as indices on tables in relational databases. However, no primary index exists. Instead, all indices have an equal priority. As a consequence, they do not impose any additional constraints on the ordering of the query evaluation, but allow for a concurrent application of the query evaluation algorithm in each dimension. Surprisingly, this comes very close to

the original promise of equal access in databases [13]. This type of parallelism takes place between multiple instances of the query evaluation algorithm that run concurrently. Additionally, the query evaluation algorithm can be parallelized as well. Furthermore, multiple queries to the same data warehouse are also independent. For this reason they can be executed in parallel, adding inter-query parallelism. Especially in distributed computing resources, inter-query parallelism can be harnessed efficiently.

Conventional wisdom 6: Data access is a bottleneck.

When examining the bottlenecks that hinder the adoption of virtual data cubes, lacks in I/O throughput and in processing power become apparent. As some researchers [16] already warn, bus bandwidths are likely become the next bottleneck in data management systems.

Proposition 6: Distributed storage, caching and pre-fetching deliver the data in time.

Parallel I/O [31, 11] and pre-fetching will have to be used to maintain and move the data sets which form the base of the information synthesis. Using distributed storage and replication, it becomes possible to scale to larger data sets while minimizing the performance degradation. Storage area networks increase both the available storage, fault tolerance and the throughput achievable. Like parallel I/O systems, they can provide data at a much faster rate.

One of the major optimizations possible is the use of caches to retain the most recently used sections of the virtual data cube in memory. Due to the spatial locality of successive queries, most requested should overlap with their predecessors. For this reason, caching appears to be one of the most important optimizations possible. Research on the best possible caching strategies may improve the performance of virtual data cubes to a level close to pre-materialization. Particularly if caching is combined with a evaluation of the query history and a prediction of the most likely queries to follow, selective pre-fetching can provide a limited pre-materialization. However, the distinction that the pre-fetched results may be invalidated by an update remains, providing exact query results in the presence of frequent and unexpected updates.

Conventional wisdom 7: Fast response times require expensive supercomputers.

Up to now all major solution providers have exclusively used expensive supercomputing hardware to implement large-scale OLAP solutions.

Proposition 7: Grid computing lowers the tab on high computational performance.

As a new emerging paradigm in parallel computing — the exclusive and rare supercomputer has been replaced with clusters and grids built from commodity hardware [2, 3, 19, 4], interconnected with high-speed networking technologies. Faster processors have found their way onto the desktop. Coupled with larger main memories, an exciting new era for data analysis solutions is dawning, which employs clusters and grids as its main platform and uses the large available memory to cache intermediate results. In the meantime, parallel and distributed computing — the two staple crops of high performance computing — serve well to improve the responsiveness of OLAP solutions.

Furthermore data mining capabilities as one of the main user applications of data warehousing should be integrated using service oriented approaches. The basic principles that motivate the design of a service-oriented grid-aware data mining system are as follows [5, 12, 24, 25, 35, 36]:

- Open architecture. In the context of the OGSA documents, openness is being used to communicate extensibility, vendor neutrality, and commitment to a community standardization process.
- Data distribution, complexity, heterogeneity, and large data size. One must be able to cope with very large and high dimensional data sets that are geographically distributed and stored in different types of repositories.
- Applying different kinds of analysis strategies. Because the first effort to service-oriented Grid-enabled data mining, a wide spectrum of data mining strategies has to be considered and accommodated in the system, and their impact on performance has to be investigated.
- Compatibility with existing Grid infrastructure. The higher levels of the architecture should use the basic Grid services for implementing wide area communication, cooperation and resource management.
- Openness to tools and algorithms. The architecture must be open to the integration of new data mining tools and algorithms.
- Scalability. Architecture scalability is needed both in terms of number of nodes used for performing the distributed knowledge discovery tasks and in terms of performance achieved by using parallel computers to speed up the data mining task.
- Grid, network, and location transparency. Users should be able to run their data mining applications on the Grid in an easy and transparent way, without needing to know details of the Grid structure and operation, network features and physical location of data sources. This is supported by a layer of Grid data virtualization services.
- Security and data privacy. Security and privacy issues are vital features in a lot of data mining applications. The Grid services offer a valid support to the system to cope with user authentication, security and privacy of data. However, knowledge discovery specific security services, such as filtering of sensitive data, are not provided by Grid middleware, thus they are implemented as Grid data mining services.

Conventional wisdom 8: Active database systems are distinct from OLAP solutions.

Rule-based trigger mechanisms are exclusively based on the relational source database. OLAP systems do neither interact with active database technology nor are aware of it.

Proposition 8: OLAP can incorporate active database technology.

Furthermore the conceptual model should incorporate the workflows leading to the feeding and use of the statistical database or OLAP data warehouse [17, 18].The addition of rule-based trigger mechanisms—as they are known from active databases—does not only add turn OLAP into a very powerful environment to express data-oriented control systems, but also permits provisions for efficient update-tracking rules [20, 23]. These rules associate regions of the logical data cube with cache objects, permitting selective cache object invalidation upon updates.

This is particularly important, if large caches are used (i.e., with multiple users). In this context an approach using stateful aggregation functions, which contain an internal state and provide the capability to adapt their value to insertions, updates and deletes accordingly by updating their value, is to be considered. This will reduce the data traffic to the aggregating processing nodes and open the possibility to track changes within the fact base from any in-memory aggregate cache.

Conventional wisdom 9: Real ad-hoc aggregations are unfeasibly slow.

The large number of possible consolidation paths poses unsurmountable obstacles to optimizing the synthesis of aggregated values. For this reason, real ad-hoc queries will always remain too slow, justifying the restrictions posed by using pre-materialized consolidation paths.

Proposition 9: A wealth of further optimizations is possible and being worked on.

The adaptation of techniques for compiler engineering will give rise to efficient aggregation engines.
These will distribute the consolidation function across distributed resources according to additional metadata. This is similar to register allocation and scheduling for instruction level parallelism. Analogous to memory loads and register spilling, disk accesses and network transfers incur large performance penalties. It is therefore important to minimize the adverse impact by exploiting local caches and local disk accesses where ever possible. Similarly, the reordering and rewriting of the aggregation function will exploit the same strategies as are used in scheduling for instruction level parallelism. The resource matrices used in compilers are equal to the additional metadata which describes the available computing resources.

Query optimization strategies for different architectures will use dynamic system information to adapt the execution plans to the system state. This is of particular concern, if a network of non-dedicated workstations will be used. Load balancing can provide a close to optimal resource utilization, if the system background load can change during the execution of a query.

In many application scenarios close approximations of aggregated values are entirely satisfactory answers. Thus approximations of aggregated values provide another approach to the optimizing of query response times. Particularly the use of wavelets [31] appears promising, as it compresses the data. The computation of some common aggregation primitives is far easier using wavelets, which could leave them as an attractive alternative in cases, where obtaining an approximate query results quickly is more desirable than obtaining an exact result slowly.

Conventional wisdom 10: Formulaic Models are only a vision.

In his seminal paper [10] Codd defines the ultimate goal as follows: “The formulaic model is the most dynamic and requires the highest degree of user interaction and associated variable data consolidation. This data model indicates which values or behaviours across multiple dimensions must be introduced into the model to effect a specific outcome. Within this dynamic model, the analyst specifies an outcome, and a starting point, animates the model, and gains insight as to what variables of which dimensions must be manipulated and to what extent, if the specified outcome is to be attained”. Codd adds: “No product of which we are aware provides this capability today”

Proposition 10: Innovative Computational Science Methods will path the way towards the Formulaic Model.

The formulaic model requires that model parameters must be obtained from the observed data. In OLAP terms measurement data should be derivable from the dimensional data. The transformation from data to model parameters in Earth sciences and other physical systems are the substance of the so called "inverse problem" in mathematics where huge progress has been achieved in the last years [22]. The use of such methods will be necessary to achieve Codd's requirements of a formulaic model.
OLAP should ultimately strive its goal in formulating the results to be achieved by the user or decision maker and deliver the different ways to achieve the goals formulated. Therefore we will need a combination of the right formal models for the information delivery requested by decision makers which will go far beyond nowadays data mining methods.

5 Conclusions

While OLAP unarguably remains one of the most exciting developments in the sector of data analysis, current implementation can be seen as harmful, representing a dead end in the general applicability of the concept. Pre-materialization and de-normalized data bypass the most basic and widely accepted database principles available, while limiting the possible applications. In order for OLAP to live up to its promise of becoming the ultimate data access and analysis interface, a re-conciliation of data analysis and data storage are overdue. The most likely form of such reconciliation is the integration of OLAP as an external view on the data, which integrates multiple data sources on-demand and synthesis summary information. This would enable the creation of non-traditional applications on top of an OLAP layer, even positioning OLAP as a paradigm for expressing data-oriented computations and control systems. Additionally, the elimination of the update window provides support for using the technology in real-time and near-time contexts, such as required for many emerging e-commerce applications.

All logical views for analysis of measurement data are OLAP

The multidimensional approach implemented in OLAP provides superior power and elegance, due to its reliance on closed multidimensional operators. Just as the relational database model every operation on a table results in result table, OLAP operators always map from an input cube to an output cube.
Taking the abovementioned one step further, we may consider the OLAP model as a generic paradigm for data-centric computing: As all measurement data are inherently multidimensional, it can be represented within an OLAP system and processed using OLAP operators. Thus, every computation can be seen as a data manipulation within a high-dimensional data space. As a final consequence, OLAP is a generic paradigm for the specification of any processing.

To achieve the ultimate goal of OLAP, i.e. the realisation of the formulaic model we will need besides the necessary tools people who possess an integrative view toward real world problems and the courage to solve them. One of Günther Vinek's worth remembering statements is: "A computer scientist should be able to immerse into what ever problem and find an optimal possible solution even in domains where he has no expertise given the necessary time to solve it. This is the essence of "Computation thinking" as it was recently propagated by Jeannette Wing [34] as the universally applicable attitude and skill set required for computer scientists. Günther Vinek has exemplary shown this attitude in his many complex problem solving activities in medicine, agriculture, chemistry, and information systems.

References

[1] AGRAWAL, R., GUPTA, A., SARAWAGI, S. Modeling multidimensional databases. Technical report, IBM Almaden Research Center, 1996.

[2] BARAK, A., GUDAY, S., WHEELER, R. The MOSIX Distributed Operating System, Load Balancing for UNIX, volume 672 of Lecture Notes in Computer Science. Springer Verlag, 1993.

[3] BARAK, A., LA'ADAN, O. The MOSIX Multicomputer Operating System for High Performance Cluster Computing. Journal of Future Generation Computer Systems, 13(4–5): 361–372, March 1998.

[4] BECKER, D. J., STERLING, T., SAVARESE, D., DORBAND, J. E., RANAWAK, U. A., PACKER, C. V. Beowulf: A parallel workstation for scientific computation. In Proceedings International Conference on Parallel Processing, 1995.

[5] BREZANY, P., WÖHRER, A., TJOA, A M. The Grid: vision, technology development and applications. Elektrotechnik und Informationstechnik 123(6): (2006).

[6] BURNS, T., FONG, E. N., JEFFERSON, D., KNOX, R., MARK, L., REEDY, C., REICH, L., ROUSSOPOULOS, N., TRUSZKOWSKI, W. Reference model for dbms standardization, database architecture framework task group (DAFTG) of ANSI/X3/SPARC database system study group. SIGMOD Record, 15(1):19–58, Jan 1986.

[7] CHAUDHURI, S., DAYAL, U. Data Warehousing and OLAP for Decision Support. In Proceedings SIGMOD'97, pages 507–508, 1997.

[8] CHAUDHURI, S., DAYAL, U. An overview of Data Warehousing and OLAP technology. In SIGMOD Record, March 1997.

[9] CODD, E. F.. A relational model of data for large shared data banks. Communications of the ACM, 13(6):377–387, June 1970.

[10] CODD, E. F., CODD, S. B., SALLEY, C. T. Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT-Mandate. Technical report, E. F. Codd and Associates, 1993.

[11] DOGAC, A., DENGI, C., OSZU, M. T. Distributed object computing platforms. In: Communications of the ACM, 41(9):95–103, September 1998.

[12] HOFER, J., BREZANY, P., Distributed Classifier Construction in the Grid Data Mining Framework GridMiner-Core. In Proceedings of the Data Mining and the Grid (GM-Grid 2004), 4th IEEE International Conference on Data Mining (ICDM'04), 2004:

[13] KIMBALL, R.. The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses. John Wiley & Sons, Inc., 1996.

[14] KIMBALL, R., STREHLO, K. Why decision support fails and how to fix it. SIGMOD Record, 24(3):92–97, September 1995.

[15] MANGISENGI, O., TJOA, A M. Modelling OLAP with extended relational models. J. of Computer Science and Information Management, 1999.

[16] MESSINA, P., CULLER, D., PFEIFFER, W., MARTIN, W., ODEN, J. T., SMITH, G.. Architecture. Communications of the ACM, 41(11):36–44, November 1998.

[17] MUECK, T., VINEK, G.. Modelling Dynamic Constraints Using Augmented Place/Transition Networks. Information Systems, Vol. 14, No. 4. Pergamon Press,1989.

[18] OLIVÉ, A.: Conceptual Schema-Centric Development,A Grand Challenge for Information Systems Research, CAiSE 2005, LNCS 3520, pp. 1 – 15, 2005.

[19] RIDGE, D., BECKER, D., MERKEY, P., STERLING, T.. Beowulf: Harnessing the power of parallelism in a Pile-of-PCs. In Proceedings IEEE Aerospace, 1997.

[20] SCHREFL, M., THALHAMMER, T.: On Making Data Warehouses Active. Proc. DaWaK Lecture Notes in Computer Science Vol. 1874, Springer Verlag, 2000.

[21] STEFANOV, V., LIST, B., SCHIEFER, J. Bridging the Gap between Data Warehouses and Business Processes: A Business Intelligence Perspective for Event-Driven Process Chains, Proc. EDOC, 2005.

[22] TARANTOLA, J., Inverse problem theory - methods for data fitting and model parameter estimation, Society of Industrial and Applied Mathematics, 2005.

[23] THALHAMMER, T., SCHREFL, M., MOHANIA, M. K.: Active data warehouses: complementing OLAP with analysis rules. Data Knowl. Eng. 39(3): 241-269 (2001).

[24] TJOA, A M., BREZANY, P., JANCIAK, I., Towards Grid Based Intelligent Information Systems. In Proc. 2003 International Conference on Parallel Processing, 2003.

[25] TJOA, A M., JANCIAK, I., WOEHRER, A., BREZANY, P. Providing an Integrated Framework for Knowledge Discovery on Computational Grids. In Proc. 5th International Conference on Knowledge Management 2005.

[26] TSISCHRITZIS, D., KLUG, A. C. The ANSI/x3/SPARC DBMS framework report of the studygroup, Information Systems, 3(3):173–191, March 1978.

[27] VINEK, G. Ein Softwaresystem zur Speicherung und integrierten Auswertung von Versuchsdaten. EDV in Medizin und Biologie 3 (1972), S. 82:

[28] VINEK, G., RENNERT, F. Die Problemsprache NUMAN zur Vorbereitung von Daten für statistische Analysen. In: EDV in Medizin und Biologie 4 (1973), S. 1.

[29] VINEK, G., RENNERT, F., TJOA, A M. Datenmodellierung: Theorie und Praxis des Datenbankentwurfes, Physika-Verlag 1982.

[30] VINEK, G., RENNERT. F., TJOA, A M., WAGNER, R. R. A Data Manipulation Language Based on a Relational Data Model for the Preparation of Statistical Analysis Proceedings in Computational Statistics. In: J. Gordesch, P. Naeve (Hrsg.), COMPSTAT 1976, Physika-Verlag, S. 379-387.

[31] VITTER, J. S., WANG, M.. Approximate computation of multidimensional aggregates of sparse data using wavelets. In Proceedings SIGMOD'99, 1999.

[32] WAND, Y., WEBER, R. "Information systems and conceptual modelling: a research Agenda," Information Systems Research, (13:4), 2002, pp.363–376.

[33] WAND, Y., WEBER, R. On Ontological Foundations of Conceptual Modeling: A Response to Wyssusek, Scandinavian Journal of Information Systems, 2006, 18(1):127-138.

[34] WING, J. Computational Thinking, CACM vol. 49, no. 3, March 2006.

[35] WOEHRER, A., BREZANY, P., JANCIAK, I., Tjoa, A M., Virtualizing Heterogeneous Data Sources on the Grid - Design Concepts and Implementation Knowledge Grid and Grid Intelligence, 2004.

[36] WOEHRER, A., BREZANY, P., TJOA, A M. Novel mediator architectures for Grid information systems. In Journal for Future Generation Computer Systems , 2005.

[37] WYSSUSEK, B., "On ontological foundations of conceptual modelling," Scandinavian Journal of Information Systems, (18:1), 2006, pp. 63-80.