Authors:	Niemi Timo, Hirvonen Lasse, Järvelin Kalervo
Name of article:	Multidimensional Data Model and Query Language for Informetrics
Year of publication:	2003
Name of journal:	Journal of the American Society for Information Science and Technology
Volume:	54
Number of issue:	10
Pages:	939-951
ISSN:	1532-2882
Discipline:	Natural sciences / Computer and information sciences
Language:	en
School/Other Unit:	School of Information Sciences

This is a preprint of an article published in Journal of the American Society for Information Science and Technology 54 (10), 939-951.

URN: http://urn.fi/urn:nbn:uta-3-759
DOI: http://dx.doi.org/10.1002/asi.10290

Multidimensional Data Model and Query Language for Informetrics

Timo Niemi[1] Lasse Hirvonen[1] and Kalervo Järvelin[2]
Departments of [1]Computer and Information Sciences and [2]Information Studies
University of Tampere, Finland

Preprint of:

Niemi, T. & Hirvonen, L. & Järvelin, K. (2003). Multidimensional Data Model and Query Language for Informetrics. Journal of the American Society for Information Science and Technology 54(xxx), to appear.

Addresses:

Department of Computer and Information Sciences
FIN-33014 University of Tampere
Finland

Department of Information Studies
FIN-33014 University of Tampere
Finland

Telephone numbers: +358 215 6782 (Niemi) +358 215 6953 (Jarvelin)
Fax number: +3582156070 (Niemi) +358 215 6560 (Jarvelin)
e-mail addresses: {Timo.Niemi; Lasse.Hirvonen; Kalervo.Jarvelin}@uta.fi

Multidimensional Data Model and Query Language for Informetrics

Timo Niemi[1], Lasse Hirvonen[1] and Kalervo Järvelin[2]

Departments of [1]Computer and Information Sciences and [2]Information Studies

University of Tampere, Finland

Abstract

Multidimensional data analysis or OLAP offers a single subject-oriented source for analyzing summary data based on various dimensions. We demonstrate that the OLAP approach gives a promising starting point for advanced analysis and comparison among summary data in informetrics applications. At the moment there is no single precise, commonly accepted logical/conceptual model for multidimensional analysis. This is because the requirements of applications vary considerably. We develop a conceptual/logical multidimensional model for supporting the complex and unpredictable needs of informetrics. Summary data are considered in respect of some dimensions. By changing dimensions the user may construct other views on the same summary data. We develop a multidimensional query language whose basic idea is to support the definition of views in a way, which is natural and intuitive for lay users in the informetrics area. We show that this view-oriented query language has a great expressive power and its degree of declarativity is greater than in contemporary operation-oriented or SQL-like OLAP query languages.

Keywords: Informetrics, OLAP, Multidimensional data model, Multidimensional query language, User Interface

1. Introduction

OLAP (On-Line Analytical Processing) or multidimensional data analysis seeks to support decision-making based on multi-dimensionally organized summary (aggregate) data. Conventional database systems have mainly been developed to support OLTP (On-Line Transaction Processing) applications, which usually are related to operational tasks in organizations. Conventional database management systems (briefly DBMSs) do not provide very powerful functions for data synthesis, analysis and consolidation, which are necessary in multidimensional data analysis (Codd, Codd & Salley, 1993). The OLAP approach offers a single source, a multidimensional database (briefly MDD), to support advanced decision-making. An MDD contains summary data pre-computed from a huge amount of raw data of operational databases. In practice the pre-computation of summary data is necessary because OLAP queries are complex and could require hours or days to run directly on the raw data of operational databases.

Sometimes OLAP and data warehousing are used as synonymous terms. Thomsen (1997) considers them complementary in that data warehousing makes raw data available to end users and ensures its accuracy and consistency (Inmon, 1992) whereas OLAP focuses on the end user's analytical requirements. Data warehousing is responsible for updating summary data to reflect the changing state of the OLAP application. Efficient updating of summary data of an OLAP application is challenging (see, e.g., Mumick, Quass & Mumick, 1997). In this paper we do not consider data warehousing and assume that accurate, consistent and timely summary data are available for OLAP processing.

It is assumed in multidimensional data analysis that a decision-maker needs `summary data` related to a specific `subject` and he must consider that data in respect of certain `factors`. Summary data are usually numerical and measurable. Therefore the attributes representing them are often called `measure attributes`. The factors on the basis of which summary data is analyzed are called `dimensions`, represented by `dimension attributes`. By selecting the specific dimensions through which summary data are analyzed one can obtain a `view` into summary data. By changing dimensions one may construct different views. This happens through OLAP queries, which specify new multidimensional views from the basic views provided by data warehousing.

Data analysts often need to group data, e.g., they want to consider dimensions at different levels of detail. Therefore it is important to represent the dimensions as `multilevel hierarchies`. For example, the dimensions `time` and `geography` could be represented as multilevel hierarchies (`days`, `weeks`, `months`, `quarters`, `years`) and (`cities`, `states`, `countries`), respectively.

The underlying data of an OLAP application are related to some domain. So far business applications have dominated (see e.g., Thomsen, 1997): financial reporting, portfolio analysis, cost/benefit analysis, marketing research and analysis, and quality management have been popular applications. However, OLAP is a general approach for decision support applications and has proven fruitful in medical (clinical) applications (Pedersen & Jensen, 1999) and in information retrieval (McCabe & al., 2000). McCabe and others (2000) combine traditional information retrieval with OLAP and show that typical multidimensional features are appropriate for text analysis. In this proposal users may in a natural way narrow or broaden their searches along hierarchical dimensions. Our aim (to our knowledge this is the first attempt) is to show that the multidimensional modeling approach gives new analyzing opportunities for informetrics as well.

Informetrics studies various statistical phenomena of literature, often based on bibliographic information provided by online databases. Among the statistical phenomena are productivity issues of authors, countries, or journals (Almind & Ingwersen, 1997; Persson, 2002) and generalized impact factors of journals or authors (Egghe & Rousseau, 1990; Hjortgaard Christensen, Ingwersen & Wormell, 1997). Also activity profiles of authors, organizations, and journals, or citation networks in the form of bibliographic coupling of authors or articles and author co-citation analysis (White, 1990) as well as literature growth and aging can be computed (Library, 1981). Informetric data seem well amenable to multidimensional modeling and OLAP-based analysis.

Several informetric measurements are produced by the ISI (Institute of Scientific Information), published in their reports, e.g., the Journal Citation Report. Informetric calculations can also be done online in the online databases. Hjortgaard Christensen and Ingwersen (1996; & Wormell, 1997) have described the methodology of various citation-based analyses using the OneSearch, RANK and TARGET commands of the Dialog Information Service. Very often ad hoc informetric measurements are needed for decision-making, e.g., for competitor information, science policy, research project funding, etc. Järvelin, Ingwersen and Niemi (2000) analyzed the requirements of informetric processing and found that contemporary systems for informetric processing fall short of advanced requirements. Multidimensional modeling and OLAP are general-purpose approaches, the potentials of which have not yet been explored for modeling and analysis of informetric data.

In this paper we pursue the following goals:

- to demonstrate that multidimensional modeling is a powerful analysis tool for informetrics;

- to present a conceptual/logical multidimensional model capable of taking into account typical requirements of informetrics;
- to develop a view-oriented query interface for MDD with a great expressive power and higher degree of declarativity than in existing operation-oriented OLAP query languages.

We have earlier proposed a high-level declarative query interface for the modeling and analysis of informetric data (Järvelin, Ingwersen & Niemi, 2000). While providing both methodological and conceptual advances, the proposed interface was based on a technology completely different from what is proposed here – that of data aggregation in the context of hierarchically modeled complex objects. The present paper shows ways of achieving these advances through current MDD and OLAP technology. Moreover, our OLAP approach allows clearly richer ways of defining dimension attributes than before. In particular, dimensions may flexibly be viewed through hierarchies defined apriori or on the fly.

The rest of the paper is organized as follows. In Section 2 we review the literature on multidimensional modeling and OLAP query languages/interfaces. Our logical model for multidimensional analysis in informetrics is given in Section 3 by way of a sample application. On the basis of this logical model a view-oriented multidimensional query language is developed in Section 4. In Section 5 several sample queries of different types are discussed. We believe that these queries show the advantages of the multidimensional approach to informetrics. Section 6 discusses the properties and the prototype implementation of our language. The conclusions are drawn in Section 7.

2. Related Work

MDDs were developed without any widely accepted formal model. Therefore there is no consensus on the primitives and the structuring among them which MDDs should contain. Another consequence is that there is no established terminology in multidimensional modeling (Vassiliadis & Sellis, 1999). A common feature of MDDs is that information is represented as multidimensional arrays.

Summary data are often modeled as a multidimensional data cube consisting of measure and dimension attributes (see e.g., Agrawal, Gupta & Sarawagi, 1997; Gray & al., 1997; Li & Wang, 1996; Kimball, 1996; Pedersen & Jensen, 1999). Thus multidimensional data cubes can be considered as the basic logical/conceptual model for OLAP while the operation set for data cube manipulation may vary considerably between proposals. At the instance level, the values of the dimension attributes are assumed uniquely to determine the values of all measure attributes. Niemi, Nummenmaa and Thanisch (2000) propose the use of functional dependencies for formal logical or conceptual data cube design. Pedersen and Jensen (1999) compare the modeling features of multi-dimensional approaches.

A very popular OLAP data model is the star schema (e.g., Inmon, 1992; Kimball, 1996; Chaudhuri & Dayal, 1997) although it is based on intuition rather than on precise formalism. In it a multidimensional data cube consisting of dimension and measure attributes is called a fact table. In addition, it contains a dimension table for each dimension attribute in the star schema. A dimension table describes the properties of the dimension at hand. The star schema is mainly a model for the logical structuring of multidimensional data. Baralis, Paraboshi and Teniente (1997) give the ER (Entity-Relationship Model) diagram for the star model. McCabe and others (2000) show that the star schema is suitable for information retrieval applications.

All multidimensional models containing fact and dimension tables are variants of the star schema, the snowflake model (e.g., Date, 2000) probably being its most famous variant. It is a star schema where the dimension tables are normalized (as in the relational model). Our multidimensional model for informetrics can also be seen as a variant of the star schema which, unlike the original star schema, consists of several fact tables. We believe that in informetric applications users often want to analyze summary data (measure attributes) with respect to dimension attribute values which satisfy specified conditions. Therefore, in order to support the specification of complex conditions among dimensions, our model allows semantic relationships between dimension tables.

Most of today's analysts do not master programming, database techniques, etc. Therefore it is important to develop high-level and intuitive declarative OLAP interfaces for them. Many existing multidimensional interfaces/query languages are operation-oriented. In these users specify textually or visually one operation at a time. In query specification (e.g., Thomsen, 1997) the user often invokes sequences of OLAP operations (slice-and-dice, drill-down, roll-up and pivot) interactively by starting from some basic cube. Each operation yields a resulting cube serving as the basis for the next operation. Obviously not all intermediate cubes are of interest to the user. Therefore an alternative for the stepwise specification style is needed.

The CUBE operation (Gray & al., 1996) is based on the most straightforward way to model multidimensional data. Its origin is in the relational framework and it has a single operand relation consisting only of dimension and measure attributes. Shukla and others (1996) extend the original CUBE operation by also allowing multilevel hierarchies of dimension attributes in the modeling of a multi-dimensional cube. Conceptually richer multidimensional models also describe properties (attributes) related to dimensions. The background assumption of the CUBE operation is that one powerful relational operation is sufficient for multidimensional manipulation. Therefore this operation is included as a standard feature in the relational query language SQL 3 (Date, 2000). The CUBE operation can be considered a generalization of the group-by operation of SQL.

So far the development of approaches and algorithms for computing the CUBE operation efficiently has dominated OLAP research. These algorithms have mainly been developed for ROLAP (Relational OLAP) implementation (see e.g., Agrawal & al., 1996; Harinarayan, Rajaraman & Ullman, 1996; Ross & Srivastava, 1997) – Zhao and others (1997) is an exception in which an algorithm for MOLAP (Multidimensional OLAP) implementation is developed. The algebras for specifying both some conventional OLAP operations and the CUBE operation, have been defined, e.g., by Gingras and Lakshmanan (1998) and Gyssens and Lakshmanan (1997). These algebras are based on the relational approach. Therefore it is not surprising that several SQL-like query languages have been proposed for multidimensional processing. The specification of the pure CUBE operation with SQL (e.g., Agrawal & al., 1996) is simple and declarative. However, the specification of queries dealing with hierarchy levels of dimensions among several OLAP cubes, is much more troublesome with an advanced SQL-like OLAP query language (e.g., MDX (Microsoft, 1998)).

SQL is originally a relational query language and not designed for multidimensional analysis. On the other hand, the origin of OLAP is rather in matrix algebra (mathematics) than in database technology. Thomsen (1997) showed that SQL has no simple way to rearrange views – especially those which need transposition between rows and columns. Due to the lack of a flexible view reorganization mechanism, SQL-like specification of views, which differs radically from those available, resembles algorithm design rather than declarative specification. Obviously such SQL-like specification is too demanding for many OLAP users. Therefore we propose an alternative approach to OLAP query specification.

In existing multidimensional approaches the user knows and gives either the dimensions or the exact values of those dimensions for analyzing summary data. In informetrics, typically, the information used in analyzing summary data is less predictable than in conventional OLAP applications. Users in informetrics do not necessarily know the exact values of those dimensions which should be used in OLAP queries. In order to enable advanced informetrics analysis, a mechanism is needed which, instead of exact dimension values, allows implicit specification of relevant dimension values. Therefore in our approach the relevant dimension values can also be produced by evaluating user-specified conditions on the properties of dimensions.

Our language is based on the notion of variable of deductive databases (see e.g., Sterling & Shapiro, 1994) and deductive databases (see e.g., Liu, 1999). This notion offers a straightforward, flexible and intuitive way to specify complex multidimensional queries. To the best of our knowledge this is the first attempt in which the notion of variable of this kind is used for specifying OLAP queries. Our multidimensional query language can be characterized as a view-oriented one in which the user only specifies the content of the result view without specifying the operations for its construction.

3. A Logical Model for Multidimensional Analysis in Informetrics

A logical model describes the available data from the perspective of the user. Therefore it is important to provide the user with a logical structuring of data that supports his or her intuitive interpretation of an MDD. From the user viewpoint, a logical structure of multidimensional data is necessary for organizing, managing and querying preprocessed data. Our logical model can be considered as a variant of the star model because it contains fact and dimension tables. However, we call fact tables (multidimensional) data cubes.

Next we introduce our multidimensional sample database, which will be used throughout the paper. At the same time, we introduce the primitives of our logical/conceptual model. Niemi, Hirvonen and Järvelin (2002) provide a more formal and detailed discussion of the model.

Our MDD consists of a collection of multidimensional *data cubes,* a collection of *dimension tables* and a collection of *hierarchy tables.* In addition, our model contains mechanisms for specifying how data in different tables are semantically associated with each other. Our sample MDD contains historical data on the publications of persons and their grants, in addition to their personal data. Further we have historical information on citations received by their articles. The corresponding instance level is given in Appendix A. In the visualization of a table we distinguish between the content (instance level) and its structural aspects (schema level). Due to space considerations, the sample database is only illustrative of the processing capabilities.

Data cubes. Data cubes contain the summary data on which multidimensional analysis is based. A complex multidimensional application (e.g., informetrics) contains data related to various contexts, which, however, may share some dimensions. It is therefore clearer to produce one data cube per context than to use one data cube involving all the contexts. A data cube consists of only those dimension attributes shared by all its measure attributes. This means that the dimension attributes form the key of a data cube. If information from several data cubes is needed they may be joined on one or more common dimensions. Navigation among data cubes is invisible to the user. Figure 1 presents the schema of our sample MDD data cubes.

Data Cubes:

authoring_table				
year	**person**	**domain**	**refereed**	**non_refereed**

grant_table			
year	**person**	**domain**	**grants**

citation_table		
year	**article**	**citation_count**

Figure 1. The schema of the sample MMD data cubes

In our sample MDD the data cube `authoring_table` expresses the total number of refereed (the measure attribute `refereed`) and non-refereed (the measure attribute `non_refereed`) publications grouped by two-year periods (the dimension attribute `year`), persons (the dimension attribute `person`) and domains (the dimension attribute `domain`). The first *row* of this table, <y1990-1, smith, ir, 3, 4> (see instance in Appendix A), is interpreted as follows: during the period 1990-1991 Smith wrote 3 refereed and 4 non-refereed publications on information retrieval. The values of measure attributes (i.e., the values 3 and 4) are called *cells* which are assumed to contain numerical summary data.

In the `grant_table` the total grants (the measure attribute `grants`) given for a person (the dimension attribute `person`), a specific research area (the dimension attribute `domain`) in a specific period (the dimension attribute `year`) are expressed. The data cube `citation_table` classifies the total number of citations (the measure attribute `citation_count`) per article (the dimension attribute `article`) and period (the dimension attribute `year`). Semantically, information in all the data cubes can only be connected on the basis of the dimension attribute `year` because it appears in each cube. Instead, the information in the data cubes `authoring_table` and `grant_table` can also be connected on the dimension attributes `person` or `domain`.

Dimension tables. Dimension tables represent dimension-specific properties. For each dimension attribute in the data cubes there may be at most one dimension table. Unlike the original star model, we allow data cube dimensionswith no dimension tables. The schema level of a dimension table consists of attribute names, whereas the instance level consists of the values of these attributes. Figure 2 presents the schema of our sample MDD dimension tables.

Dimension Tables:

person_info			
person	position	degree	yob

article_info					
article	author	title	forum	year	cs_class

forum_info			
forum	type	publisher	refereed

Figure 2. The schema of the sample MMD dimension tables

The dimension tables person_info, article_info and forum_info give properties related to persons, articles and their publication forums, respectively. In person_info we express the name (the attribute person), position, degree and year of birth (yob) of a person. The values of the attribute person uniquely identify the rows in this dimension table. These values also appear as values in the data cubes authoring_table and grant_table (see Appendix A). Through this shared attribute one can join data from authoring_table and grant_table and the dimension table person_info. This supports the implicit specification of dimension attribute values – e.g., through the position or degree of persons (say, senior scientists holding a Ph.D.) without having to list them by person.

The dimension table article_info expresses the author, title, forum, year and class in the ACM Computer Science Classification (the attribute cs_class) of an article (the dimension attribute article whose values appear in the data cube citation_table). In this case, too, the values of the dimension attribute uniquely identify the rows. The dimension table forum_info is associated indirectly with the dimension article by giving information on the publishing forums of the articles. The values of the attribute forum are used to connect information in the dimension tables article_info and forum_info (see Appendix A). The type (journal or conference), publisher name and an indication on the refereeing process (the values yes or no) of a forum are expressed by the attributes type, publisher and refereed, respectively.

In informetrics, summary data must often be analyzed on the basis of complex criteria related to properties of dimensions. Therefore the specification of these criteria may require the use of information in several dimension tables. Consequently, the semantic relationships among dimension tables are organized on the basis of the shared values of some attributes in the dimension tables. By navigating through the sample MDD, it is possible to find, e.g., the citation counts of articles published on refereed ACM forums.

Hierarchical tables. Hierarchy tables are always associated with dimension attributes, while measure attributes do not have hierarchy tables at all. Hierarchy tables support the analysis of information in data cubes at different levels of detail. Unlike some other approaches, we do not represent hierarchy levels in dimension tables – they would contain different types of information requiring different types of processing. Our model allows multiple hierarchies for each dimension, making this separation even more important. Figure 3 presents the schema of our sample MDD hierarchical tables. We trust that the information in our hierarchical table instances (see Appendix A) is self-explanatory.

Hierarchical Tables:

Dimension attribute	**Hierarchy level**	**Immediate Sub-hierarchy level**
domain	discipline	domains

Dimension attribute	**Hierarchy level**	**Immediate Sub-hierarchy level**
person	country	institution
person	institution	author

Dimension attribute	**Hierarchy level**	**Immediate Sub-hierarchy level**
year	all_times	years

Figure 3. Schema of the sample MMD hierarchical tables

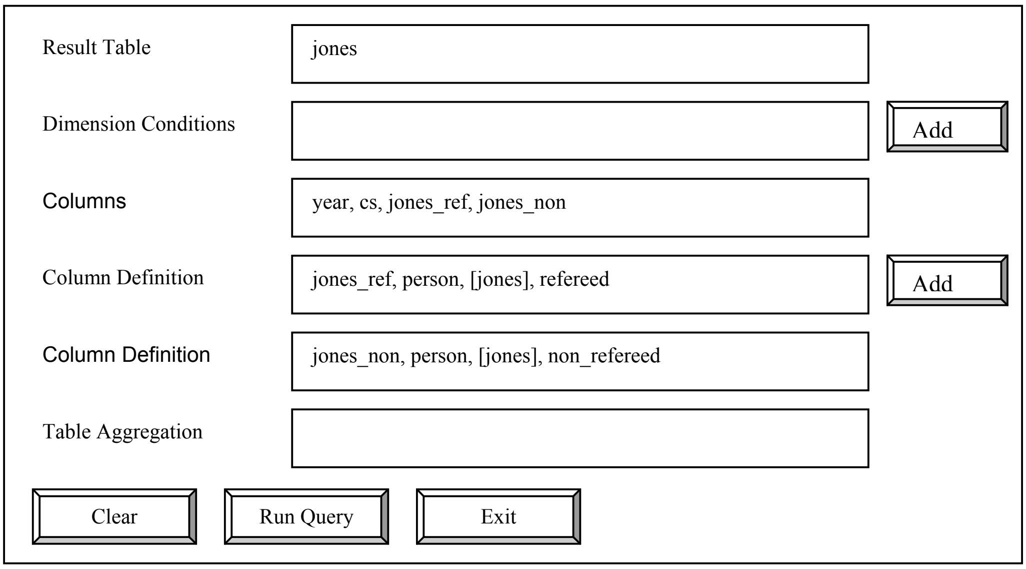
The values of the dimension attributes in the data cubes are represented on the basis of the lowest hierarchy levels defined for the dimensions (see Vassiliadis, 1998). The hierarchy levels of a hierarchy table are defined by grouping the values of the next lower level according to some principle. The construction of hierarchical levels starts by grouping values of some dimension attribute. Informetric analysis often employs ad hoc hierarchies, which may be difficult to predict. Therefore our approach offers a flexible mechanism for users to specify new hierarchies.

4. View-oriented Query Language for Multidimensional Analysis

The idea of our multidimensional query language is to offer an expressive, flexible and user-friendly tool for multidimensional analysis. Unlike most existing multidimensional query languages, our query language is not SQL-like but view-oriented. In our approach the user describes the multidimensional information in the result view without specifying the operations for its construction. Further, the user's navigation among data (tables) is minimal.

4.1. The Table Skeleton for Specifying the Result View

Let us assume that in our sample MDD the user wants to know how many refereed and non-refereed publications in computer science Jones has published per two-year period. The filled-out table skeleton specifying this query is given in Figure 4, and its result in Figure 5.



Result Table	jones
Dimension Conditions	
Columns	year, cs, jones_ref, jones_non
Column Definition	jones_ref, person, [jones], refereed
Column Definition	jones_non, person, [jones], non_refereed
Table Aggregation	

Figure 4. Sample Query 1

jones	year	cs	jones_ref	jones_non
	y1990_1	ir	5	2
	y1990_1	db	0	1
	y1992_3	ir	4	2
	y1992_3	db	0	1
	y1994_5	ir	3	2
	y1994_5	db	2	2
	y1996_7	ir	2	2
	y1996_7	db	5	1
	y1998_9	ir	2	3

y1998_9	db	1	3

Figure 5. Result of Sample Query 1

The fields in the table skeleton have the following meaning in the specification of Sample Query 1.

The field Result Table. The result view is named **jones**.

The field Dimension Conditions. If the user does not know the specific values of some dimension attribute needed in the query, he or she can give an expression in this field to evaluate them. This field is not needed in Sample Query 1 – we will discuss it later below.

The field Columns. In formulating the query, the user first has to select the appropriate dimensions or their hierarchy levels. The values of the measure attributes in the result view are computed on the basis of value combinations of these dimensions. In our sample query this means that the total numbers of refereed and non-refereed publications are computed on the basis of different value combinations of the dimension attribute **year** and the hierarchy level **cs** (a hierarchy level of the dimension attribute **domain** expressed in Appendix A). In Appendix A we can see that this hierarchy level consists of **ir** (information retrieval) and **db** (databases). Next the user specifies and names those columns which are associated with measure attributes, i.e., with the measure attributes **refereed** and **non_refereed**. Because the result view contains information only on the publications of Jones, the user names these new columns **jones_ref** and **jones_non**, which describe the total number of refereed (non-refereed) publications by Jones.

The field Column Definition. The new columns **jones_ref** and **jones_non** must be specified in the field Column Definition. In the first case the Column Definition field has the form: **jones_ref, person, [jones], refereed**. Here **jones_ref** expresses the new column name. The parameters **person** and **[jones]** express the dimension attribute and its values, respectively, with regard to which the summary data of the measure attribute **refereed** is viewed. The specification of the attribute **jones_non** is similar.

The other fields. By writing atoms col_sums, row_sums, col_avg, and row_avg in the field Table Aggregation, the user can specify respectively the total or average sums for columns or rowsbe computed. This field is not needed in Sample Query 1 – we will discuss it later below. The Add buttons are used for generating new fields of types Dimension Conditions and Column Definition. The buttons Clear, RunQuery and Exit are used for clearing the skeleton table of all text, for executing the query, and for finishing the query session.

Assume that we want to analyze how many refereed and non-refereed publications have been published in a specific domain in a specific period. We can formulate this kind of query by replacing the columns **jones_ref** and **jones_non** of Sample Query1 with the columns **all_ref** and **all_non** and by giving the following Column Definition fields for them:

- **all_ref, person, [smith, jones, hines, peters, wilks], refereed**
- **all_non, person, [smith, jones, hines, peters, wilks], non_refereed**.

The assumption in this query is that the user knows all persons in our sample MDD.

By replacing the column **jones_non** of Sample Query1 with the column **smith_ref** and by giving the Column Definition field **smith_ref, person, [smith], refereed** we can compare how the total numbers of the refereed publications of Jones and Smith have changed over time. We believe that these examples illustrate how in our approach multidimensional analysis is specified in an intuitive and declarative way.

4.2. The Notion of Variable in Query Formulation

In many existing multidimensional query languages the user has to know the dimensions and their values when formulating queries. In informetrics there is an obvious need to analyze summary data in an MDD based on ad hoc criteria concerning the properties of dimension attributes. We cannot assume in large MDDs that the user knows what dimension attribute values satisfy specific criteria. Nor can we assume that the user is willing to list them even if he or she knew. Therefore, powerful but simple means for specifying criteria for properties of dimensions are needed. In complex cases the user has to navigate among several dimension tables in specifying criteria related to properties of dimensions. Because the user is responsible for this navigation it is desirable that the user can express navigation in a natural way. We borrow the notion of variable from deductive databases (see e.g., Liu, 1999) and show that it allows intuitive navigation. To our knowledge, our query language is the first to use this kind of a notion of variable for specifying multidimensional queries.

We introduce the notion of variable without logic-based rules, in the same way as in QBE (Zloof, 1975). Intuitively, a variable is used to refer to an unknown value of some attribute in a dimension table. In our language it is sufficient that the user interprets correctly to which values a specific variable refers. In one Dimension Conditions field the user gives all criteria which the properties of a specific dimension must satisfy. In this field the individual sub-criteria are associated with each other by conjunctions. A variable begins with a capital letter whereas constants are numbers or strings, which begin in lower caseletters.

We refer to any dimension table by xyz(D1, D2, …, Dn) where xyz is a dimension table name and the variables Di stand for its attributes. For example, in our sample MDD the expression person_info(Per, Pos, Inst, Deg, Yob) refers to any row in the dimension table person_info (see Appendix A).

In addition to variables we can use constants in expressions referring to dimension tables. A constant indicates that we are interested only in rows in which the attribute has this constant value. For example, the expression person_info(Per, Pos, mit, Deg, 1950) only matches the first row (with the instantiations Per = hines, Pos = professor, Deg = dr) of this dimension table.

We also use shared variables of deductive databases and QBE. If the same variable appears in various sub-criteria connected by conjunctions then this variable is called a shared variable. A shared variable must be instantiated to the same value in all sub-criteria in which it occurs. For example, let us consider the following conjunction of two criteria: person_info(Per, Pos, Inst, Deg, Yob) and Yob > 1966. Intuitively, we want to find such persons whose year of birth is later than 1966. The third row in person_info is the only row which satisfies the criteria.

Navigation among two dimension tables is expressed simply by a shared variable. Because a shared variable must be initialized to the same value in both dimension tables, we can connect semantically related data from both dimension tables through such instantiations. If we want to know the articles written by scientists in UCLA, we need the dimension tables person_info and article_info (see Appendix A). By using a shared variable to refer to the attributes person and author in these dimension tables we connect information in them semantically to each other. We can express this in our sample MDD as follows: person_info(Per, Pos, ucla, Deg, Yob) and article_info(Art, Per, Title, For, Year, Cl). In the expression the shared variable is Per. The possible instantiations for Per in the first sub-criterion are jones and smith, while in the second sub-criterion the possible instantiations for the variable Art, which are related to these instantiations of Per, are art1, art3, art4, art6, art 10, art11, art12, art13 and art16. Shared variables can similarly be used to navigate among several dimension tables.

Conditions for the properties of dimensions are specified in the field Dimension Conditions of table skeletons. Their evaluation produces values related to variables in conditions. In the field Column Definition the user gives the values of some dimension attribute related to which the values of the given measure attribute are summed up in the result view. Therefore we need a mechanism for transferring the values of the dimension attribute from the field Dimension Conditions to the field Column Definition. Let *Expr* be a condition expression, which may consist of several sub-criteria connected by conjunctions, and *X* a variable occurring therein. The field Dimension Conditions then has the form: *X from Expr*. By using the expression *all(X)* in the third parameter of the field Column Definition (see above) we transfer the values of *X* which satisfy *Expr* to this parameter. Through this mechanism the user can analyze multidimensional data without knowing / listing exact values of dimension attributes. If we have the expression **Art from person_info(Per, Pos, ucla, Deg, Yob) and article_info(Art, Per, Title, For, Year, Cl) and forum_info(For, journal, Pub, yes)** in the field Dimension Conditions and the expression **ucla_cit, article, all(Art), citation_count** in a field Column Definition, then the column **ucla_cit** would contain the total number of the citations received by the refereed journal articles written by UCLA scientists.

5. Sample Queries

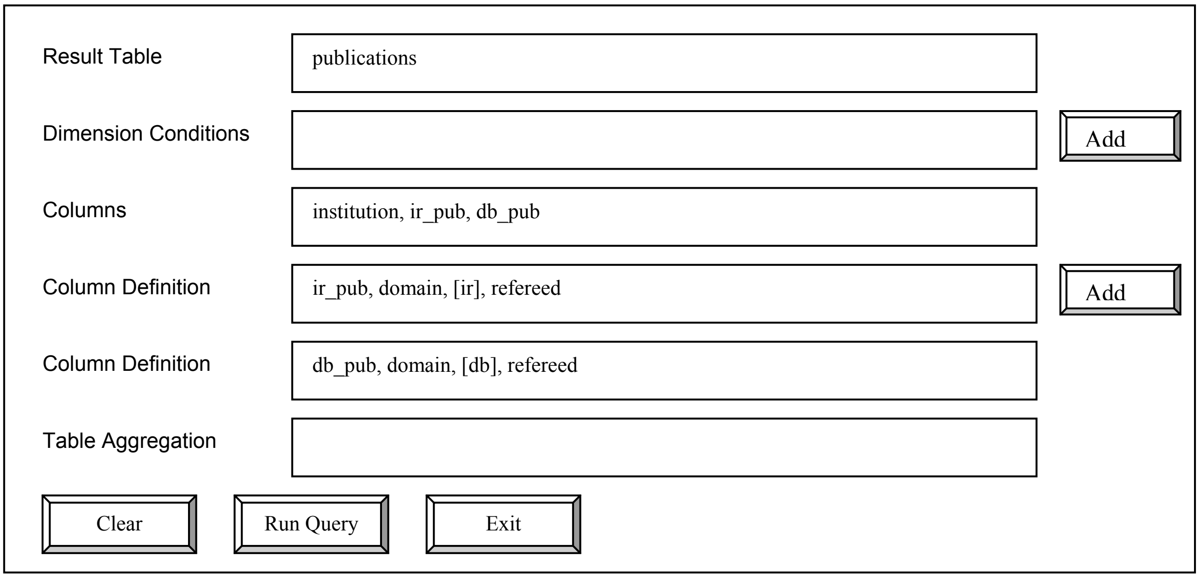
The sample queries aim at showing the usefulness of multidimensional analysis in informetrics. We shall consider them in two categories. In the first category (Sample Queries 1 to 3) the user knows the dimension attributes, their values and hierarchy levels. In the second category (Sample Queries 4 and 5) the user does not know which values of dimension attributes have certain properties. Most multidimensional query languages only support queries belonging to the first category. In informetric applications the dimension attribute values may be so numerous (e.g. lists of articles by an author set) that their exhaustive listing is very laborious if not impossible. This makes the second category a necessity for informetrics. Our sample queries show that the degree of declarativity is high in our query language. The reader can evaluate all the results (given in Appendix B) of our sample queries on the basis of the sample MDD given in Appendix A.

5.1 Sample Queries based on Known Values of Dimension Attributes

Sample Query 2 demonstrates that the summary data can be viewed easily at higher hierarchy levels defined for dimensions. Sample Query 2 accertains how many refereed information retrieval and database articles various institutions have published. In the sample MDD, **institution** is a hierarchical level defined for the dimension person. Therefore we can use it directly as a column in our result view. People working in a specific institution are expressed at the instance level (Appendix A). The publications of these individuals are regarded as publications of the institution. Figure 6 presents Sample Query 2. This query is an example of *institutional productivity analyses* in informetrics.

5.2 Sample Queries Based on the Use of Variables

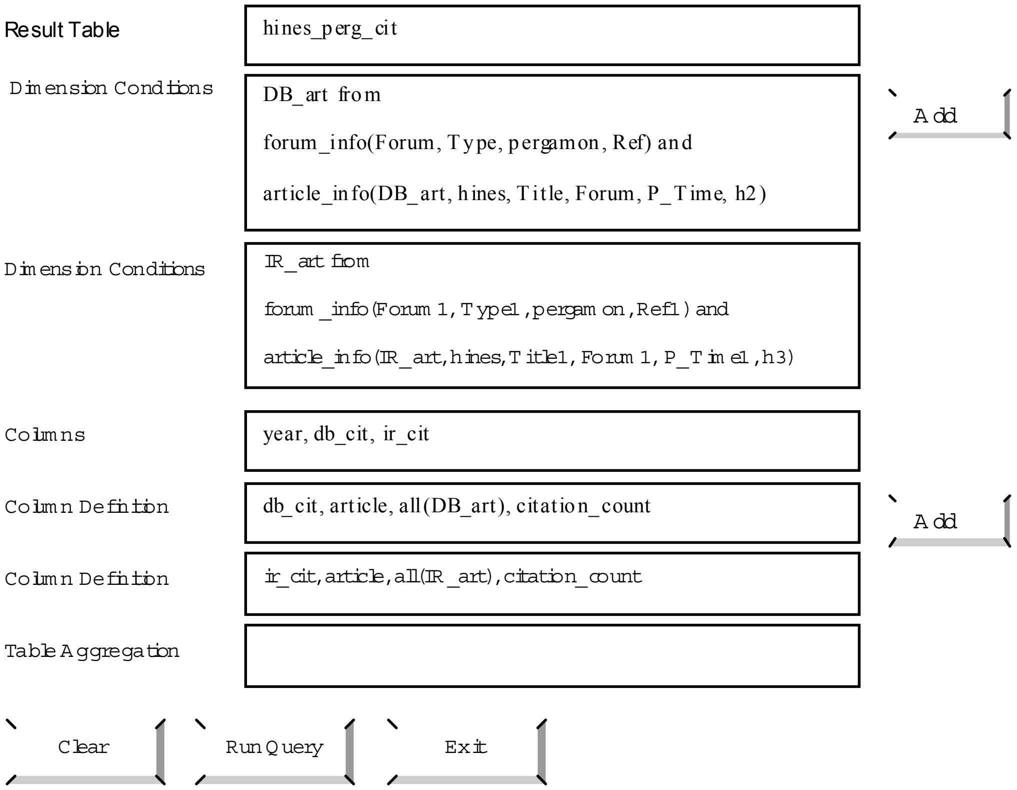
In informetrics the user does not always know, or would have difficulty listing, the relevant values of a dimension attribute. Assume that the user is interested in analyzing over time the number of citations given to articles published in refereed journals or conference proceedings. If the user knows that the articles art1 - art3 and art8 - art16 have been published in refereed journals and the articles art4 - art7 have been published in refereed conference proceedings then he or she can formulate the result view as shown above. However it is unrealistic to suppose that users generally possess such knowledge. Therefore we need the mechanism based on variables to determine these values.



Result Table	publications
Dimension Conditions	
Columns	institution, ir_pub, db_pub
Column Definition	ir_pub, domain, [ir], refereed
Column Definition	db_pub, domain, [db], refereed
Table Aggregation	

Figure 6. Sample Query 2

Sample Query 3 focuses on the total number of citations given to information retrieval and database articles written by Hines and published by Pergamon. To select these articles one has to fill in two Dimension Conditions fields, one for information retrieval articles and one for database articles. These fields are specified in the same way (we consider the specification of the latter). The variable **Forum** is used in the specification as the shared variable and the instantiations of this variable are forums of Pergamon in the expression **forum_info(Forum, Type, pergamon, Ref)**. By the expression **article_info(DB_art, hines, Title, Forum, P_time, h2)** we require that the articles deal with the database area (i.e. the attribute `cs_class` must have the value h2 according to the ACM Computer Science Classification). In addition we require that the author attribute has the value **hines**. Connecting these two expressions with the conjunction we obtain the complete specification and the instantiations of the variable **DB_art** contain only those articles that satisfy all conditions. This query, presented in Figure 7, is an example of *complex multi-criteria impact analyses* in informetrics.



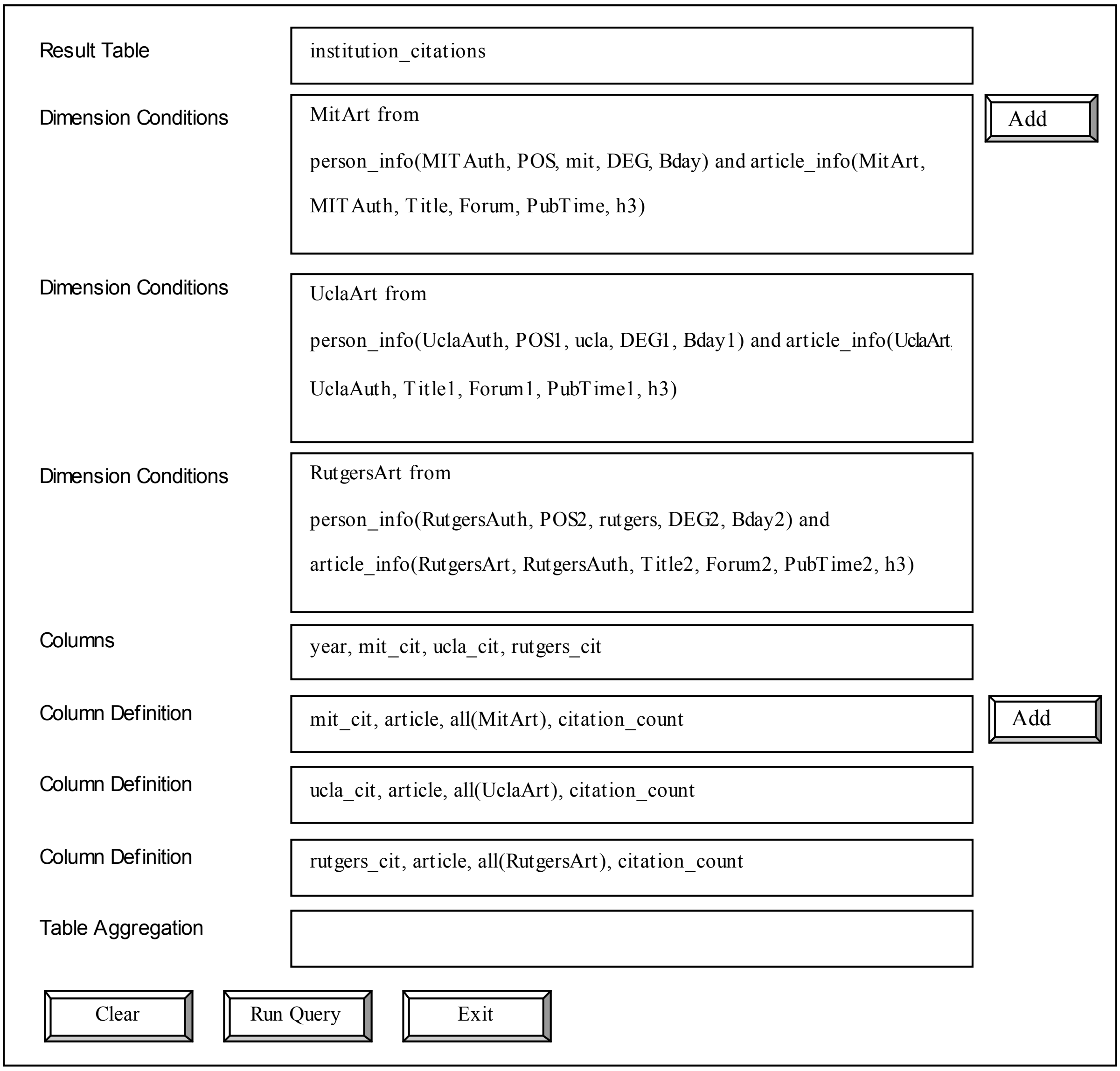
Result Table	hines_perg_cit
Dimension Conditions	DB_art from forum_info(Forum, Type, pergamon, Ref) and article_info(DB_art, hines, Title, Forum, P_Time, h2)
Dimension Conditions	IR_art from forum_info(Forum1,Type1,pergamon,Ref1) and article_info(IR_art,hines,Title1,Forum1,P_Time1,h3)
Columns	year, db_cit, ir_cit
Column Definition	db_cit, article, all(DB_art), citation_count
Column Definition	ir_cit,article,all(IR_art),citation_count
Table Aggregation	

Figure 7. Sample Query 3

If the user wantrd to compare, over time, the number of the citations given to these articles with the number of all the citations given to those articles by Hines addressing the above domains then he or she could define two new columns, say **all_db_cit** and **all_ir_cit**, for this purpose. These columns are defined in two Column Definition fields as follows: **all_db_cit, domain, [db], citation_count** and **all_ir_cit, domain, [ir], citation_count**. In addition, from the viewpoint of informetrics, this query represents *comparative impact analyses*.

Sample Query 4 analyzes in two-year periods the total number of citations that have been assigned to the information retrieval articles published by the authors belonging to a specific institution. This means that we have to define three Dimension Conditions fields for information retrieval articles published by people working in MIT, UCLA, and Rutgers, respectively. These fields are defined analogously. The instantiations of the variable **MITAuth** in the expression **person_info(MITAuth, POS, mit, DEG, BDAY)** are authors working in MIT whereas the instantiations of the variable **MitArt** in the expression **article_info(MitArt, MITAuth, Title, Forum, PubTime, h3)** are information retrieval articles written by these authors. Note that the value h3 of the attribute `cs_class` refers to information retrieval in the ACM Computer Science Classification. This query, presented in Figure 8, is another example of *complex multi-criteria impact analyses* in informetrics.

Let us assume that we expand Sample Query 4 so that also total grants given to people working in the same institutions are expressed by two-year periods. Thus we must define three new columns, called **mit_grants**, **ucla_grants** and **rutgers_grants**, respectively. In the Dimension Conditions field **MitPerson from person_info(MitPerson, P1, mit, D1, BD1)** the instantiations of the variable **MitPerson** express people working at MIT. Similar Dimension Conditions fields apply to **UclaPerson** and **RutgersPerson**. These variables are then used in the Column Definition fields as follows: **mit_grants, person, all(MitPerson), grants** and likewise for **ucla_grants** and **rutgers_grants**. In the expanded query the result view contains summary information from two data cubes citation_table and grant_table. This expanded query demonstrates that, unlike in the existing SQL-like query languages, in our approach the user is not responsible for navigating among data cubes. This feature makes query formulation both straightforward and compact.



Result Table: institution_citations

Dimension Conditions: MitArt from person_info(MITAuth, POS, mit, DEG, Bday) and article_info(MitArt, MITAuth, Title, Forum, PubTime, h3)

Dimension Conditions: UclaArt from person_info(UclaAuth, POS1, ucla, DEG1, Bday1) and article_info(UclaArt, UclaAuth, Title1, Forum1, PubTime1, h3)

Dimension Conditions: RutgersArt from person_info(RutgersAuth, POS2, rutgers, DEG2, Bday2) and article_info(RutgersArt, RutgersAuth, Title2, Forum2, PubTime2, h3)

Columns: year, mit_cit, ucla_cit, rutgers_cit

Column Definition: mit_cit, article, all(MitArt), citation_count

Column Definition: ucla_cit, article, all(UclaArt), citation_count

Column Definition: rutgers_cit, article, all(RutgersArt), citation_count

Table Aggregation:

Figure 8. Sample Query 4

6. Discussion

In the multidimensional analysis of informetrics we cannot always predefine all the hierarchical levels users need in their queries. Therefore one of the key ideas in our approach is to provide the user with a simple mechanism for defining new hierarchical levels for dimensions. Let us assume that in our sample MDD the user wants to analyze how many refereed publications appeared in various domains of computer science in the first and last half of the 1990's. Because the year dimension has only the hierarchical levels all_times and years the user has to define a new hierarchical level for half decades. The user may define the first and last half decade to consist of the years 1990-1995 and 1996-1999, respectively. In our approach the query can be represented with the columns **cs** (a predefined hierarchy level in Appendix A), **first_half_decade_ref** and **last_half_decade_ref**. In the Column Definition fields the user expresses the hierarchy levels related to the two last columns as follows: **first_half_decade_ref, year, [y1990_1, y1992_3, y1994_5], refereed** and **last_half_decade_ref, year, [y1996_7, y1998_9], refereed**. Thus both the predefined hierarchy levels and the hierarchy levels defined by the user can be expressed very declaratively in our language.

Our multidimensional data model has a rich modeling power consisting of data cubes, dimension tables and hierarchical tables. On the one hand this affords great expressive power but on the other hand this means a need for navigation among data. If the user were responsible for all navigation then query formulation would become complicated. Therefore our principle in language design has been to spare the user from any navigation which can be done automatically. The unavoidable remaining navigation must then be made as intuitive as possible.

Let us expand the above query so that the user is also interested in the total grants given to different domains of computer science in the first and last half of the decade. This expansion only requires the addition of two columns, called **first_half_decade_grants** and **last_half_decade_grants**, which are defined by the following Column Definition fields: **first_half_decade_grants, year, [y1990_1, y1992_3, y1994_5], grants** and **last_half_decade_grants, year, [y1996_7, y1998_9], grants**. In our sample MDD this query requires the manipulation of the data cubes authoring_table and grant_table. The navigation is based on shared values of the dimension attributes year and domain but remains invisible to the user.

Our sample queries above demonstrated that in informetrics it is also necessary to allow multidimensional analysis based on the values of dimension attributes, which the user cannot list. In this case the user stipulates conditions for dimension attributes which the relevant attribute values satisfy. We have shown (see e.g., Niemi, Christensen & Järvelin, 2000) that recursive rule-based query formulation typical of deductive databases is too complex for lay users. However, users can easily adopt the notion of the variable of deductive databases and QBE. It is easy to interpret the instantiations of a variable related to some attribute as the values of this attribute. Likewise, the use of shared variables is a very intuitive way to combine semantically related data from several dimension tables. We have demonstrated that the notion of variable supports the specification of powerful ad hoc conditions on dimension attributes. Therefore the same summary data may be analyzed on the basis of several criteria. For example, in our sample MDD the total number of citations given to articles can be analyzed over time on the basis of persons, institutions, domains, forums, forum types, publishers or whether articles have been refereed or not.

Further, each analytical basis contains a huge number of variations, which may be used in the classification of total numbers of citations. For example, by classifying the total number of citations to articles based on authors we can do it, e.g., on the basis of position, education or ages of authors. Of course one may combine these variations freely. For example, we can compare the total number of citations to articles by experienced doctors (the criterion being the birth year being before 1956) with the total number of citations given to articles by young non-doctors (born after 1955). We have demonstrated that in our approach multidimensional analysis based on complex conditions could be specified compactly and intuitively. These features meet the requirements for conceptual generalizations in informetrics presented by Järvelin, Ingwersen and Niemi (2000).

The result view may also contain information derivable from other information therein. In the field Aggregation we can express the total or average sums for columns or rows. In the same way we may express other conventional operations, such as the maximal or minimal values for columns or rows. Likewise, the query table skeleton may be extended with a new field where formulas would be defined between columns. This kind of extension can be easily added to our table skeleton so that each formula produces one column, which the user would name in the field Column. This kind of extension makes it possible to compute different ratios over columns, e.g., the impact factors, directly. Our aim in the future will be to enhance our approach so that the user need not know the columns of the result view. In the existing implementation the user must specify precisely each column in the result view. For example, in our Sample Query 5, in which we considered the total numbers of citations given for all articles produced by a specific institution, the user must know that the MDD contains information on MIT, UCLA and Rutgers. However, it is possible to analyze the institutions available and generate one column for each of them. This is a demanding task, but on the other hand this facility increases the declarativity of our query language even more.

We implemented the prototype of our existing multidimensional language in Prolog based on the constructors introduced by Niemi & Järvelin (1991).

7. Conclusions

We developed a multidimensional data model consisting of data cubes, dimension tables, hierarchy tables, and the relationships among them, for informetric applications. We showed that this model offers a powerful analyzing tool for informetrics. Likewise we demonstrated that the multidimensional structuring approach is a general approach for a complex analysis in the informetrics area. Analysts in informetrics often need to analyze summary data based on several dimensions at different levels of detail. A specific perspective for summary data is called a view. We therefore developed a view-oriented multidimensional query language. In our approach the user specifies the multidimensional information of the result view without having to specify the operations needed for its construction. In our query language the user only fills in fields of table skeletons.

Because many analysts in informetrics find techniques such as programming, database techniques etc. demanding, it is important to develop a high-level declarative multidimensional query language for them. Therefore our query language design provides the analysts with a query language whose degree of declarativity is greater than in contemporary operation-oriented or SQL-like OLAP query languages. This feature is especially important in informetrics because information is often analyzed by less predictable ad hoc criteria than in conventional OLAP applications. We demonstrated that many interesting multidimensional queries in informetrics are based on specifying criteria for the properties of dimensions. We showed that the notion of variable borrowed from deductive databases allows the user to specify these criteria in an intuitive and compact way. In order to support declarative query formulation, our language has been designed so that the user need not specify navigation among data cubes.

Acknowledgement

This research was supported by the Academy of Finland under the grant number 52894.

References

Agrawal, R., Gupta, A. & Sarawagi (1997). Modeling Multidimensional Databases, Proceedings of the 13th International Conference on Data Engineering (pp. 232-243), Birmingham.

Agrawal, S., Agrawal R., Deshpande, P. M., Gupta, A., Naughton, J. F., Ramakrinshnan, R. & Sarawagi, S. (1996). On the Computation of Multidimensional Aggregates, Proceedings of the 22nd VLDB Conference (pp. 506-521), Bombay.

Almind, T. & Ingwersen, P. (1997) Informetric analyses on the World Wide Web: Methodological approaches to "webometrics". *Journal of Documentation,* 53(4), (pp. 404-426).

Baralis, E., Paraboshi, S. & Teniente, E. (1997). Materialized View Selection in a Multidimensional Database, Proceedings of the 23rd VLDB Conference (pp. 156-165), Athens.

Codd, E. F., Codd, S. B. & Salley C. T. (1993). Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. Available at http:// www.hyperion.com/products/whitepapers/

Chaudhuri, S. & Dayal, U. (1997) An Overwiew of Data Warehousing and OLAP Technology. ACM Sigmond Record 26 (pp. 65-74).

Date, C. (2000) Introduction to Database Systems, 7th edition. Addison Wesley, Reading, MA.

Egghe, L. & Rousseau, R. (1990) Introduction to Informetrics: Quantitative methods in Library, Documentation and Information Science. Amsterdam: Elsevier.

Gingras, F. & Lakshmanan, L. (1998) nD-SQL: A Multi-dimensional Language for Interoperability and OLAP, Proceedings of the 24th VLDB Conference (pp. 134-145) New York.

Gray, J., Bosworth, A., Layman, A. & Pirahesh, H. (1996). Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-Tab, and Sub-Totals, Proceedings of the 12th International Conference on Data Engineering (pp. 152-159), New Orleans.

Gyssens, M. & Lakshamanan L. V. S. (1997). A Foundation for Multi-Dimensional Databases, Proceedings of the 23rd VLDB Conference (pp. 106-115), Athens.

Harinarayan, V., Rajaman, A. & Ullman, J. D. (1996). Implementing Data Cubes Efficiently, Proceedings of the ACM Sigmod Conference (pp. 205-216), Montreal.

Hjortgaard Christensen, F. & Ingwersen, P. (1996). Online citation analysis: a methodological approach. *Scientometrics,* 37(1), (pp. 39-62).

Hjortgaard Christensen, F., Ingwersen, P. & Wormell, I. (1997) Online determination of the journal impact factor and its international properties. *Scientometrics,* 40(3), (pp. 529-540).

Inmon, W. H. (1992). Building the Data Warehouse. QED Technical Publishing Group, Wellesley, Massachusetts.

Järvelin, K., Ingwersen, P. & Niemi, T. (2000). A User-oriented Interface for Generalized Informetric Analysis Based on Applying Advanced Data Modelling Techniques, Journal of Documentation 56 (pp. 250-278).

Kimball, R. (1996). The Data Warehouse Toolkit, John Wiley and Sons, USA.

Liu, M. (1999). Deductive Database Languages: Problems and Solutions. ACM Computing Surveys 31 (pp. 27-62).

Li, C. & Wang, X. S. (1996). A Data Model for Supporting On-Line Analytical Processing, Proceedings of Conference on Information and Knowledge Management (pp. 81-88), Baltimore, MD.

Library Trends. Special issue on bibliometrics, summer 1981. *Library Trends,* 30(1).

McCabe, M. C., Lee, J., Chowdhury, A., Grossmann, D. & Frieder, O. (2000). On the Design and Evalution of a Multi-dimensional Approach to Information Retrieval, Proceedings of the ACM SIGIR Conference (pp. 363 –365), Athens.

Microsoft (1998) Microsoft OLE DB for OLAP Programmer's Reference, Microsoft Corporation.

Mumick, I. S., Quass, D. & Mumick, B. S. (1997). Maintenance of Data Cubes and Summary Tables in a Warehouse, Proceedings of the ACM Sigmod Conference (pp. 100-111), Tucson.

Niemi, T. & Järvelin, K. (1991). Prolog-based Meta-rules for Relational Database Representation and Manipulation, IEEE Transactions on Software Engineering 17 (pp. 762-788)

Niemi, T., Nummenmaa, J. & Thanicsh, P. (2000). Functional Dependencies in Controlling Sparsity of OLAP Cubes, Proceedings of the 2nd Conference Data warehousing and Knowledge Discovery, Lecture Notes in Computer Science, Vol. 1874, (pp. 199-209), Springer-Verlag.

Niemi, T. Christensen, M. & Järvelin, K. (2000). Query Language Approach Based on the Deductive Object-Oriented Database Paradigm. Information and Software Technology 42 (pp. 777- 792).

Pedersen, T. B. & Jensen C. S. (1999). Multidimensional Data Modeling for Complex Data, Proceedings of the 15th International Conference on Data Engineering (pp. 336-345) Sydney.

Persson, O. *BibExcel.* http://www.umu.se/inforsk/Bibexcel/index.html (visited 21 August 2002).

Ross, K. A. & Srivastava, D. (1997). Fast Computation of Sparse Datacubes, Proceedings of the 23rd VLDB Conference (pp. 116-125), Athens.

Shukla, A., Deshpande, P. A., Naughton, J. F. & Ramasamy, K. (1996). Storage Estimation for Multidimensional Aggregates in the Presence of Hierarchies, Proceedings of the 22nd VLDB Conference (pp. 522-531), Bombay.

Sterling, L. & Shapiro, E. (1994). The Art of Prolog, second edition. The MIT Press.

Thomsen, E. (1997) OLAP Solutions Building Multidimensional Information Systems. John Wiley and Sons, USA.

Vassiliadis, P. (1998) Modeling Multidimensional Databases, Cubes and Cube Operations, Proceedings of the 10th International Conference on Scientific and Statistical Data Management.

Vassiliadis, P. & Sellis, T. (1999), A Survey of Logical Models for OLAP Databases. ACM Sigmod Record 28 (pp. 64-69).

White, H.D. ed. (1990). Perspectives on author co-citation analysis. *Journal of the American Society of Information Science,* 41(6) (pp. 430-468).

Zhao, Y., Deshpande, P. M. & Naughton, J. F. (1997) An Array-based Algorithm for Simultaneous Multidimensional Aggregates, Proceedings of the ACM Sigmod Conference (pp. 159-170), Tucson.

Zloof, M. Query-By-Example: Operations on transitive closure. Yorktown Heights, NY: IBM, RC 5526.

Appendix A The instance level of the sample MDD

Data Cubes:

grant_table	year	person	domain	grants
	y1990_1	smith	ir	400
	y1990_1	smith	db	0
	y1990_1	jones	ir	0
	y1990_1	jones	db	100
	y1990_1	hines	ir	20
	y1990_1	hines	db	0
	y1990_1	peters	ir	50
	y1990_1	peters	db	0
	y1990_1	wilks	ir	200
	y1990_1	wilks	db	0
	y1992_3	smith	ir	0
	y1992_3	smith	db	200
	y1992_3	jones	ir	20
	y1992_3	jones	db	100
	y1992_3	hines	ir	0
	y1992_3	hines	db	0
	y1992_3	peters	ir	30
	y1992_3	peters	db	0
	y1992_3	wilks	ir	100
	y1992_3	wilks	db	0
	y1994_5	smith	ir	0
	y1994_5	smith	db	100
	y1994_5	jones	ir	0
	y1994_5	jones	db	0
	y1994_5	hines	ir	100
	y1994_5	hines	db	0
	y1994_5	peters	ir	100
	y1994_5	peters	db	0
	y1994_5	wilks	ir	50
	y1994_5	wilks	db	0
	y1996_7	smith	ir	0
	y1996_7	smith	db	0
	y1996_7	jones	ir	30
	y1996_7	jones	db	0
	y1996_7	hines	ir	200
	y1996_7	hines	db	0
	y1996_7	peters	ir	0
	y1996_7	peters	db	200
	y1996_7	wilks	ir	0
	y1996_7	wilks	db	300
	y1998_9	smith	ir	200
	y1998_9	smith	db	0
	y1998_9	jones	ir	0
	y1998_9	jones	db	0
	y1998_9	hines	ir	50

authoring_table	year	person	domain	refereed	non_refereed
	y1990_1	smith	ir	3	4
	y1990_1	smith	db	1	4
	y1990_1	jones	ir	5	2
	y1990_1	jones	db	0	1
	y1990_1	hines	ir	0	1
	y1990_1	hines	db	3	4
	y1990_1	peters	ir	0	1
	y1990_1	peters	db	2	3
	y1990_1	wilks	ir	3	3
	y1990_1	wilks	db	0	2
	y1992_3	smith	ir	2	4
	y1992_3	smith	db	0	1
	y1992_3	jones	ir	4	2
	y1992_3	jones	db	0	1
	y1992_3	hines	ir	1	2
	y1992_3	hines	db	5	5
	y1992_3	peters	ir	0	1
	y1992_3	peters	db	3	0
	y1992_3	wilks	ir	2	3
	y1992_3	wilks	db	3	1
	y1994_5	smith	ir	3	4
	y1994_5	smith	db	1	4
	y1994_5	jones	ir	3	2
	y1994_5	jones	db	2	2
	y1994_5	hines	ir	0	2
	y1994_5	hines	db	4	6
	y1994_5	peters	ir	2	0
	y1994_5	peters	db	1	1
	y1994_5	wilks	ir	4	3
	y1994_5	wilks	db	1	2
	y1996_7	smith	ir	3	4
	y1996_7	smith	db	1	4
	y1996_7	jones	ir	2	2
	y1996_7	jones	db	5	1
	y1996_7	hines	ir	1	2
	y1996_7	hines	db	6	4
	y1996_7	peters	ir	1	2
	y1996_7	peters	db	0	3
	y1996_7	wilks	ir	2	3
	y1996_7	wilks	db	0	3
	y1998_9	smith	ir	3	2
	y1998_9	smith	db	2	2
	y1998_9	jones	ir	2	3
	y1998_9	jones	db	1	3
	y1998_9	hines	ir	1	0

y1998_9	hines	db	0
y1998_9	peters	ir	0
y1998_9	peters	db	500
y1998_9	wilks	ir	0
y1998_9	wilks	db	250

y1998_9	hines	db	0	1
y1998_9	peters	ir	0	0
y1998_9	peters	db	2	3
y1998_9	wilks	ir	3	2
y1998_9	wilks	db	0	1

citation_table	year	article	Citation_count
	y1990_1	art1	5
	y1990_1	art3	0
	y1990_1	art5	1
	y1990_1	art7	0
	y1990_1	art9	0
	y1990_1	art11	0
	y1990_1	art13	4
	y1990_1	art15	0
	y1990_1	art2	0
	y1990_1	art4	6
	y1990_1	art6	0
	y1990_1	art8	0
	y1990_1	art10	2
	y1990_1	art12	0
	y1990_1	art14	0
	y1990_1	art16	0
	y1992_3	art1	10
	y1992_3	art3	0
	y1992_3	art5	2
	y1992_3	art7	1
	y1992_3	art9	5
	y1992_3	art11	0
	y1992_3	art13	9
	y1992_3	art15	1
	y1992_3	art2	1
	y1992_3	art4	4
	y1992_3	art6	4
	y1992_3	art8	0
	y1992_3	art10	4
	y1992_3	art12	0
	y1992_3	art14	0
	y1992_3	art16	0
	y1994_5	art1	8
	y1994_5	art3	0
	y1994_5	art5	0
	y1994_5	art7	15
	y1994_5	art9	3
	y1994_5	art11	3
	y1994_5	art13	4
	y1994_5	art15	8
	y1994_5	art2	1
	y1994_5	art4	2
	y1994_5	art6	12
	y1994_5	art8	3
	y1994_5	art10	6
	y1994_5	art12	0
	y1994_5	art14	0
	y1994_5	art16	1
	y1996_7	art1	3
	y1996_7	art3	0
	y1996_7	art5	2
	y1996_7	art7	10
	y1996_7	art9	0
	y1996_7	art11	1
	y1996_7	art13	1
	y1996_7	art15	8
	y1996_7	art2	0
	y1996_7	art4	0
	y1996_7	art6	6
	y1996_7	art8	7
	y1996_7	art10	4
	y1996_7	art12	0
	y1996_7	art14	0
	y1996_7	art16	3
	y1998_9	art1	0
	y1998_9	art3	0
	y1998_9	art5	1
	y1998_9	art7	10
	y1998_9	art9	0
	y1998_9	art11	0
	y1998_9	art13	1
	y1998_9	art15	3
	y1998_9	art2	0
	y1998_9	art4	1
	y1998_9	art6	2
	y1998_9	art8	10
	y1998_9	art10	1
	y1998_9	art12	0
	y1998_9	art14	0
	y1998_9	art16	5

Dimension tables:

person_info	person	position	institution	degree	yob
	hines	professor	mit	dr	1950
	jones	senior_scientist	ucla	dr	1960

forum_info	forum	type	publisher	refereed
	jasis	journal	wiley	yes
	acm_sigir	conf	acm	yes

peters	engineer	mit	m_eng	1970
smith	scientist	ucla	m_sc	1965
wilks	senior_scientist	rutgers	dr	1955

inform_syst	journal	pergamon	yes
ipm	journal	pergamon	yes
isko_conf	conf	isko	no
isko_conf	conf	isko	no

article_info	article	author	title	forum	year	class
	art1	jones	vector_space_mod	jasis	y1990_1	h3
	art2	peters	expert_ir_system	jasis	y1992_3	h3
	art3	smith	ir_interface_for	jasis	y1996_7	h3
	art4	jones	ir_in_structured_d	acm_sigir	y1990_1	h2
	art5	hines	probalistic_ir	acm_sigir	y1992_3	h3
	art6	jones	intelligent_irs_for	acm_sigir	y1992_3	h3
	art7	peters	extended_boolean	acm_sigir	y1992_3	h3
	art8	hines	oo_dbms_query	inform_syst	y1994_5	h2
	art9	hines	probalistic_ra	inform_syst	y1992_3	h3
	art10	smith	distributed_rdbm	inform_syst	y1990_1	h2
	art11	jones	heterogenous_da	inform_syst	y1994_5	h2
	art12	jones	deductive_oodb	inform_syst	y1996_7	h2
	art13	smith	distributed_ir	ipm	y1990_1	h3
	art14	hines	oo_ir_system_fra	ipm	y1996_7	h3
	art15	wilks	vsm_and_probabil	ipm	y1992_3	h3
	art16	smith	distributed_ir_sys	ipm	y1994_5	h2

Hierarchical tables:

domain	dicipline	domains
	cs	ir
	cs	db

year	all_times	years
	alltimes	y1990_1
	alltimes	y1992_3
	alltimes	y1994_5
	alltimes	y1996_7
	alltimes	y1998_9

person	country	institution	author
	usa	ucla	smith
	usa	ucla	jones
	usa	mit	hines
	usa	mit	peters
	usa	rutgers	wilks

Appendix B Sample Query Results

The result of Sample Query 2:

publications	institution	ir_pub	db_pub
	rutgers	14	4
	mit	6	26
	ucla	30	13

The result of Sample Query 3:

hines_perg_cit	year	db_cit	ir_cit
	1998-9	10	0
	1996-7	7	0
	1994-5	3	3
	1992-3	0	5
	1990-1	0	0

The result of Sample Query 4:

institution_citations	year	mit_cit	ucla_cit	rutgers_cit
	1998-9	17	3	3
	1996-7	30	10	8
	1994-5	51	24	8
	1992-3	33	23	1
	1990-1	10	9	0