South African Computer Journal
Number 18
December 1996

Suid-Afrikaanse Rekenaartydskrif
Nommer 18
Desember 1996

Computer Science and Information Systems

Special Edition: Computer Security

Rekenaarwetenskap en Inligtingstelsels

The South African Computer Journal
An official publication of the Computer Society of South Africa and the South African Institute of Computer Scientists

Die Suid-Afrikaanse Rekenaartydskrif
'n Amptelike publikasie van die Rekenaarverenging van Suid-Afrika en die Suid-Afrikaanse Instituut vir Rekenaarwetenskaplikes

Subscriptions

	Annual	Single copy
Southern Africa:	R50,00	R25,00
Elsewhere:	$30,00	$15,00

An additional $15 per year is charged for airmail outside Southern Africa

to be sent to:
Computer Society of South Africa
Box 1714 Halfway House 1685

Phone: +27 (11) 315-1319 Fax: +27 (11) 315-2276

UP Drukkers Printers

Guest Contribution

A Pragmatic Approach to Development Information to Provide Service on a Wide Scale

Stephen S Mncube, PhD
Divisional Manager: Development Information Services, Centre for Policy and Information, Development Bank of Southern Africa

1 Introduction

The rapid technological growth in our time has produced an explosion of information. This, in turn, has spawned information systems based on the use of computers and automated systems. These mechanised devices with their seemingly infinite capacity to store and retrieve knowledge on command have myriad applications. But the use of computer and automated information devices pose serious problems to individuals, groups and societies on an international scale in disseminating the available information. This is even more true in information flow between regions with high information capabilities than in those that have little or none. The information flow between these regions has been varied and frequently haphazard whenever it has existed.

In this context the philosophy for development information speaks to interdependence and humanitarian concern in information flow to various regions of the earth. Information must be perceived as a universal entity. The initial failure of one region and the ability of another to acquire information should not dictate a permanent global demarcation into 'developed and less developed' categories.

2 Objectives

The objective of development information is for it to function as an interlinking mechanism between a complex industry of information resources and the users of information. In addition, development information should be a catalytic agent that aims at providing objective clarification in information needs between regions with high information capabilities and those with little.

3 Mission

The mission is to strive for a move in all regional, national and international agencies and organisations concerned with information to give full co-operation and assistance in setting criteria and standards, formulating policy, and assessing the information needs of a given region of the world. Development information is capable of realistically assessing information related to socio-economic development in the light of the unique requirements of world regions in need of information and information technology.

4 The Dynamics of Information Processing

The proliferation of information technology production has resulted in the growth and development of an ever expanding information packaging industry. This area has become so large in size and scope that it is necessary to discuss some of the important developments that are taking place in the area.

The newest phrases now being used in information packaging range from electronic archives, compact discs (CDs), computer tapes, microfiche, teletexts, video discs, magnetic tapes and interactive imaging systems (optical systems) to word processing and the use of laser technology. Developing countries will not escape this new wave of information packaging. It soon will be bombarded with vendors of these products, to a point where some adaptation will be inevitable.

Assuring the quality of technical processes and the accuracy of packaging information is becoming an increasingly difficult task. Rapid increases in the volume of information, the sophistication of information uses, and the complexity of material flows and processes are characteristic of most modern technical environments. As complexity increases, the risk of introducing significant errors into material processes increases. The very complexity of such systems makes the detection of error itself a complex task. With increasing frequency, public and private organisations are seeking help from corporations with experience in quality control and information validation to ensure that technical process and information packaging meet performance and accuracy standards. The problems association with quality control and validation can be minimised by following these guidelines:

- Establishing ways of aiding in planning, organisation and control of software purchasing and development

through
  - creating a directory of software suppliers
  - evaluating the quality of software supplied
  - keeping abreast of the state-of-the-art in software production
- Providing leadership in the innovative use of software materials and the utilisation of extensive market research on software before making a major purchase
- Establishing cost-efficient ways of packaging and designing your own software by learning how to design and evaluate software for your own use

Preparation for these new technologies for packaging information ought to be made in institutions of higher learning. Perhaps it would be timely to introduce some of these concepts in technical institutes in order that future demands imposed by the new information technology may be met.

The successful growth of developing countries information technology will ultimately depend upon the commitment of substantial resources, especially financial resources. The successful application of this information technology will require more than the mere receiving and storing of it. In addition to the tasks of acquiring and organising informational materials, channels must be established to analyse incoming information. Too much of the information technology that does get transferred out of the industrialised nations is never utilised because it is unsuited for the consumption of users in developing countries. A great deal more effort must be made to analyse, package and disseminate materials on existing and forthcoming information in all vital areas of work and study so that these technologies will be accessible to the developing countries' information-user communities.

5 The Need to Establish a Consortium

Information technology specialists need to establish a consortium of regional, national and international information networks and associations. The consortium could be an open structure inviting any institutions, organisations and agencies existing for the purpose of forming a network or documentation clearinghouse and of providing information technology not as an end-product but as a means for human change.

The element common to all membership is an interest in and dedication to providing useful and accurate information that can bring about humanistic change. Equally important is a commitment to the development of relevant information resources to meet the needs of regions with low information capacity.

The philosophical outlook of the consortium would therefore be to crystalise and emphasise broad knowledge, deep understanding, and imaginative efforts, including a dedication to great ideas in providing accurate solutions to the information needs of various regions on an international scale.

6 The Organisation of Services for Members

The ideas constituting a conceptual framework for a service-oriented consortium are as multitudinous as the Kalahari sands but in this instance the consortium could function to:

- support creative change within its membership
- facilitate and support new educational enterprises and programs addressed to meet the needs of previously disenfranchised persons
- develop and implement co-operative programs and projects among it members
- provide a meeting ground for a diversity of persons, institutions, and agencies with common values and purposes
- provide a forum for the exchange of ideas among its member associates
- encourage methods of solving social problems
- influence public policy to be consistent with its mission and purpose.

7 An Appraisal of Internet

While casual observers have the leisure to observe unobtrusively the growth and development in Internet to be a world-wide phenomenon in information sharing, they do so at no cost. On the other hand, information specialists have to judge and weigh the work of an ongoing Internet program and estimate its usefulness as a network or networks to their daily operations. Information specialists are, therefore, still more sceptical about the scope and magnitude of the Internet. They alone are facing challenges of adding another performance task of being evaluation researchers of Internet in order to provide objective clarification of incorporating Internet as an integral component of their information system. This can be a tedious undertaking because it entails not only knowing how to navigate the Internet network but also cognisance of the following key factors:

- how appropriate the Internet is to your information environments
- to what extent the databanks provided through Internet are relevant to the mission and objective of your environment
- what the relationship is between costs and benefits of having Internet at your disposal

All these factors need to be addressed to determine the effectiveness of Internet in any given information environment, be it in a government setting or in other work environments.

8 Training in Information Networks

A seminar for the network should be designed to launch the co-operative exchange of knowledge and experience with information accessibility and utility of the participant's respective information holdings. To succeed in this effort, the organisers will marshal appropriate interdisciplinary experts and technical resources from within the regions involved. The content of the seminar will consist of information related to formalising and establishing a plan for information networking. It is important that information ministries achieve a high level of knowledge and sensitivity to the information needs of their individual country, region, and ultimately the world, in order to assess, prescribe, design, manage and evaluate the most appropriate uses of information technology for enhancing the advancement of their world countries.

Although the organisers will determine who will be invited to participate in this seminar, special effort will be made to ensure that representatives come from a diversity of backgrounds, and have some knowledge or experience relating to information systems. Additionally, consideration will be given to the level of information technology which is currently utilised by the representative's country.

The content of the seminar will be tailored to meet the specific needs and issues designated by the participants through a pre-seminar survey and needs assessment, which will be administered by the organisers.

9 Summary

The above aims at presenting some possible scenario and does not pretend to be exhaustive. The issues, however vital to development information, are given cursory treatment here. It remains important for the luminaries in this area to expand on some of the thoughts contained above.

Bibliography

1. A D Burnett. *Technology for information in development: proceedings of the sixth conference of the International and Comparative Librarianship Group of the Library Association.* International and Comparative Librarianship Group of the Library Association, Brighton Polytechnic, Falmer, Sussex, August 21–23 1987.
2. M Cook. Guidelines on curriculum development in information technology for librarians, documentalists, and archivists. Paris Unesco, 1986.
3. O Fairfax, J Durham, and W W. Audio-visual materials: development of a national cataloguing and information service: Report of a joint feasability study for the british library and the council for educational technology. Distributed by Councils and Education Press, 1976.
4. M Fransman. *1948 – The market and beyond: Cooperation and competition in information technology development in the Japanese system.* Cambridge University Press, New York, 1990.
5. N K Hanna. The information technology revolution and economic development. World Bank Discussion Papers, World Bank Washington DC., 1991.
6. N Heaton and L W MacDonald. *Human factors in information technology product design and development.* Ellis Horwood, 1991.
7. T Koizumi and G E Lasker, eds. *Advances in education and human development: new trends in learning and teaching evolution of human systems and institutions, individual freedom and sociopolitical controls in the high technology environment of the advanced information society.* International Institute for Advanced Studies, Baden-Baden, West Germany, 1990.
8. V Sethi. The development of measures to assess the extent to which an information technology application provides competitive advantage. Ann Arbor: MI, University Microfilm International, University of Pittsburgh, 1988.

SACJ is produced with kind support from
Mosaic Software (Pty) Ltd.

An Integrated Classification of Multiple Database Systems

Karen Renaud and Paula Kotze
Department of Computer Science and Information Systems, University of South Africa, P.O. Box 392, Pretoria, 0001.
{renaukv,kotzep}@riscl.unisa.ac.za

Abstract

In many applications today multiple pre-existing database systems are integrated into one multiple database system. There are various characteristics of these systems which distinguish them from other types of multiple database systems. These distinguishing characteristics are autonomy, heterogeneity and distribution. Based on this, each multiple database system can be classified as a certain type of multiple database system. In this paper, the various characteristics of multiple database systems are discussed, and a classification taxonomy is introduced which can be used to distinguish different types of multiple database systems from one another. The multidatabase concept is also defined in terms of the given classification.

Keywords: *Multiple database system, autonomy, heterogeneity, distribution, multidatabase, schema, database management system*
Computing Review Categories: *H.2.4,H.2.1*

1 Introduction

Applications increasingly require access to data residing in multiple, geographically distributed data stores. As a result, the integration of these stores has assumed some importance in recent years and the rapid development in networking technology has made this integration tenable. In many instances, the data sources are pre-existing database systems operating in heterogeneous hardware and software environments, and following different protocols for concurrency control and recovery. Multiple database systems (MDBSs) are an important research area and the research into this area is expected to gain momentum with more and more enterprise mergers, takeovers and restructuring taking place worldwide. [10, 12]

MDBSs are an ideal means of sharing data and resources without affecting the autonomy of the individual database systems. MDBSs, which are built in a bottom-up fashion from pre-existing database systems, can be loosely defined as an interconnection of autonomous database systems [1].

The MDBS concept can be best illustrated by means of an example. Figure 1 illustrates the assumed architecture of a typical MDBS — namely, various MDBSs with an MDBS software layer incorporated on top of these systems. This software layer could either be located on a separate computer or on the same computer as one of the component database systems. This MDBS represents the computer setup in a particular company which has decided to purchase three pharmacies, located in Pretoria, Cape Town and Port Elizabeth, South Africa. The new company needs to access the data records of these three pharmacies each with their own database systems:

1. Tonic Pharmacy, situated in Pretoria, which currently uses a relational database system;
2. Medilots Pharmacy, situated in Cape Town, which currently uses a network database system, and
3. Harbour Pharmacy, situated in Port Elizabeth, currently using an object-oriented database system.

Each individual pharmacy's database system must still continue to function as it always has, but in addition, the new owner needs to do global queries which combine data from all three sites, as well as update and retrieval type transactions on the data in all the databases.

For instance, the new company needs to access all three databases to check sales; to do stock control; to keep tabs on the Schedule 6 and 7 medicines, and to generate monthly accounts. The individual database systems must still be able to execute local transactions to register sales, issue prescriptions and enter purchases on customer accounts. Because the database systems at each site are autonomous and pre-existing, the best option is *not* to standardize them to all use the same database management system (DBMS) and underlying database structure.

Each pharmacy database system is independent, and continues to function as it always has, with local users submitting transactions, and the local database administrator (DBA) admitting users, and managing and administering the local database system.

2 Characteristics of Multiple Database Systems

There are three features which characterize MDBSs: *distribution, heterogeneity* and *autonomy* [9].

2.1 Distribution

The distribution characteristic deals with the location of data. Two cases can be identified. The data is either physically distributed over multiple sites or is stored at one site [9].

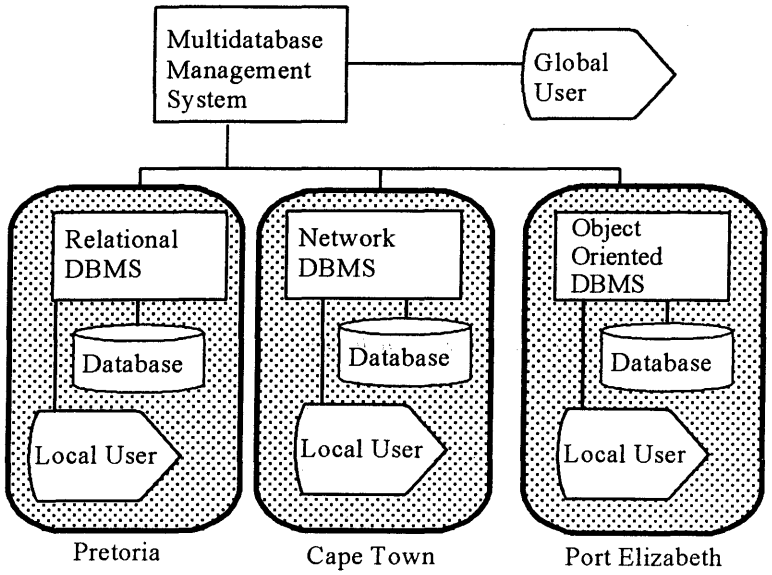

Figure 1. Overall architecture of the example multiple database system

2.2 Heterogeneity

Heterogeneity refers to the diversity with respect to the multiple databases which make up the component databases of the MDBS. This diversity can present as one of the following [6, 11]:

1. Diversity of hardware: configuration, instruction sets, data formats and representation (e.g., IBM mainframes, DEC VAXes, UNIX or PC Power machines such as Pentiums running Windows NT).
2. Operating system diversity: file system, naming, file types, operations, interprocess communication and transaction support (e.g. IBM/VM, DEC VAX/VMS, Microsoft DOS, Microsoft Windows 3.1, Microsoft Windows 95 or UNIX).
3. Diversity in networking protocols: the networks connecting the various databases to the MDBS may have different protocols (e.g. TCP/IP, UDP, DECnet, SNA or remote procedure calls (RPCs)).
4. Variations in the DBMS:
   (a) Differences in data managers where perhaps two component databases both use a relational database, but while one uses Oracle as the DBMS, another uses Informix.
   (b) Differences in underlying data models: network, hierarchical, relational or object oriented.
   (c) Transaction management primitives and related error detection facilities available through the local database interfaces.
   (d) Concurrency control, global commitment and recovery schemes used by the local database system's DBMS.
5. Semantic heterogeneity: this occurs when there may be a disagreement about the meaning, interpretation, or intended use of the same data found in different data stores forming part of the MDBS.

2.3 Autonomy

Autonomy refers to the distribution of control. It indicates the degree to which individual component databases in an MDBS can operate independently. Breitbart *et al* [4] and Özsu & Barker [9] each cite three levels of autonomy. They use different terminology, but the autonomy levels are basically the following:

1. *Design autonomy* — The local operations of the individual DBMSs are not affected by their participation in the MDBS and no modifications are made to local DBMS software.
2. *Execution autonomy* — Each local DBMS at the database sites retains complete control over the execution of transactions at its site. This type of autonomy is especially important if participating database systems belong to different, competing organizations that may not have complete trust in one another, and wish to retain complete control over their databases.
3. *Communication autonomy* — The local database systems integrated by the multiple database software layer are not able to coordinate the actions of global transactions executing at several different sites. This constraint implies that the local DBMSs do not share their control information with each other or with the MDBS software layer.

It can be argued [10], that the preservation of local autonomy is both desirable and necessary in an MDBS for the following reasons:

1. Since a local or component database is essentially an independent database system, many applications have been developed prior to integration. These applications should continue to execute after integration.
2. Since local DBMSs controlling access to local databases had total control over their database before integration, it is desirable for them to have the same measure after integration.

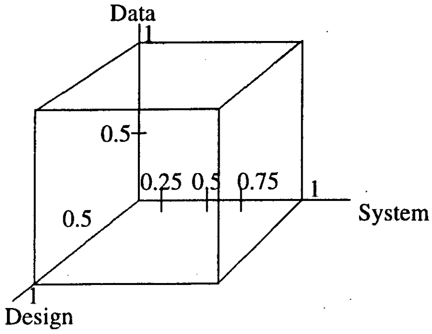

Figure 2. Modification dimension's axes [2, p.156]

3. Local autonomy allows participating database systems to be added to or removed from an MDBS environment with ease.

It would be useful to have some sort of method for determining the autonomy level in specific MDBSs with specific software. One such scheme will be introduced in the following section.

2.4 Quantification of autonomy

Barker [2] has set out a number of guidelines for the quantification of autonomy on what he calls *multidatabase systems* (MDSs). This strategy will be applied to our MDBSs. Barker identifies three fundamental dimensions: *modification, execution* and *information exchange.* The variations in these dimensions are not necessarily absolute, but should be viewed as a continuum where some autonomy can be sacrificed rather than an all-or-nothing situation. Each dimension is awarded a value ranging from 0 (no autonomy violation) to 1 (maximum autonomy violation). These values are used to arrive at a single value indicating the autonomy violation of a particular software protocol.

1. **Modification** — This addresses *what* needs to be modified to permit a database system's participation in the MDBS. This dimension relates to the design autonomy as identified in section 2.3. Each axis has an equal weight, all axes have the same length. This means that each dimension which is used to quantify autonomy is equally important in the calculation of a value for autonomy violation. When no violation occurs, the value zero is assigned to the axis, while a maximal violation will be assigned a value of 1. The information is arranged as follows:
   (a) **The data itself** *(on the y axis)*: Changes to how data is stored in local databases are considered.
   (b) **The DBMS system** *(on the x axis)*: Four values can be assigned, each representing a more invasive modification to the system.
   (c) **The way the database is designed** *(on the z axis)*: This addresses changes to the underlying local schemas used by each DBMS.
2. **Execution** — This addresses the level, either global or local, that controls the execution sequence at each individual database site. This dimension relates to the execution autonomy as identified in section 2.3.

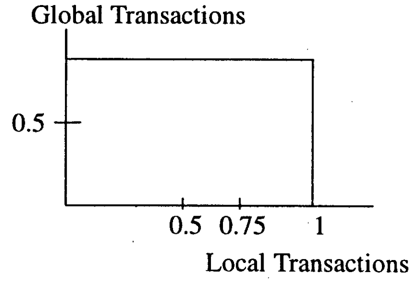

Figure 3. Execution dimension's axes [2, p.158]

Global transactions are typically split up into global subtransactions — one such subtransaction for each database which is accessed by the global transaction.

Transactions carry out various operations on the database, but for the purpose of this discussion the most important are *read, write, commit* and *abort.* The transaction will eventually reach a commit point and will then either commit (in which case, all the operations carried out on the database become permanent) or abort (in which case, none of the operations carried out on the database will be reflected on the database).

   (a) **Local transactions** *(on the x axis)*: This will indicate to what extent the local transactions can execute independently of the MDBS of which it is a component.
   (b) **Global transactions** axis *(the y axis)*: This is an indication of the extent to which the execution of global subtransactions is controlled by the MDBS software layer.
3. **Information exchange** — This dimension addresses the amount of information that must pass between the local database systems and the MDBS software level to permit a database system to join or leave the MDBS, and to ensure correct execution while it is functioning. This dimension relates to the communication autonomy as identified in section 2.3. Dimensions of interest here are:
   (a) **Data information** *(on the y axis)*: How data is exchanged between levels.
   (b) **Schema information** *(on the z axis)*: How schema information is exchanged when a database system joins or leaves the MDBS.
   (c) **Execution information** *(on the x axis)*: How concurrency control, deadlock and reliability information is exchanged at runtime.

There is some overlap between the axes of various dimensions. This is beneficial since there there is a close relationship between the various autonomy dimensions, and a small violation in one area may impact on other dimensions. When values have been determined for the three (or two) axes of autonomy violation for any of the individual dimensions, namely x, y and z, the final overall autonomy violation indicator for that dimension can be worked out using the following formula:

$$\bar{n} = \sqrt{x^2 + y^2 + z^2}.$$

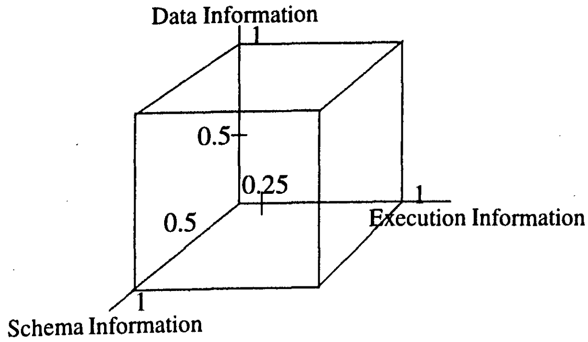

Figure 4. Information Exchange dimension's axes [2, p.159]

The maximum value for $\bar{n}$ is $\sqrt{3} \sim 1.732$. A maximal violation of any single axis is significant because, if it is maximally violated, it represents 57.7% of the maximum and a maximal violation along two axes is 81.6%. This demonstrates the characteristic of this quantification method, which reflects maximal violations far more seriously than multiple minor violations. An adjustment also needs to be made so that the execution dimension makes the same impact on the final result as the other two dimensions. The reason for this is that if the same method is used its maximum contribution would be $\sqrt{2}$, which is less than the maximum value of the other dimensions, namely $\sqrt{3}$. The easiest way to do this would be to multiply the final value awarded to the execution autonomy by $\sqrt{1.5}$. This permits the execution dimension to exhibit the same characteristics as the other dimensions.

After quantifying these three dimensions, a total autonomy violation can be worked out by taking the three dimensions: *m : modification, e : execution*, and *i : information* into account. An autonomy violation value, $\bar{a}$, for the entire system is calculated as follows:

$$\bar{a} = \sqrt{m^2 + e^2 + i^2}.$$

The maximum length of each dimension is $\sqrt{3}$ so that the maximum value of $\bar{a}$ will be $\sqrt{9} = 3$. A single value therefore represents a measure of the autonomy violated by a particular system integration approach [2].

Using these measurements, the terms *fully autonomous, semi-autonomous* and *non-autonomous* can be defined.

1. An MDBS is said to be **fully autonomous** if the individual database systems making up the system are stand-alone database systems that know nothing about the existence of other database systems which make up the MDBS. They also have no notion of any type of communication with the DBMSs of the other component database systems. In this case, the value for the autonomy violation would be 0.
2. An MDBS is said to be **semi-autonomous** if the component database systems can operate independently but have decided to participate in an MDBS to make their local data shareable. They are not fully autonomous because they require certain changes to be made to their DBMSs in order to participate in the MDBS. The final overall autonomy violation would probably be midway between the maximum $\sqrt{9}$ and the minimum 0.
3. An MDBS is said to be **non-autonomous** if a single image of the entire database is available to any user who wants to share the information which may reside in the multiple databases. The individual database systems will typically not operate independently even though they probably have the functionality to do so.

An example of a typical autonomy evaluation of a non-autonomous MDBS may look as follows:

Modification Dimension

System	1	New standards are imposed
Data	0	Data remains untouched
Design	X	Not discussed

Total value of m = 1

Execution Dimension

Local Transaction	1	Submitted via the MDBS
Global Transaction	1	Controlled by the MDBS

Total value of e = 1.732

Information Exchange Dimension

Execution	1	Execution coordinated by the MDBS
Data	X	Not discussed
Schema	1	Entire Schema provided

Total value of i = 1.414

The overall autonomy violation is $\bar{a} = \sqrt{m^2 + e^2 + i^2}$ = 2.5

3 Classification of Multiple Database Systems

There seems to be much confusion in the literature about the classification of MDBSs. Three main classification methods have been identified:

1. classification according to architectural alternatives [3, 11];
2. classification according to degree of autonomy, heterogeneity and distribution [9], and
3. classification according to how tightly the participating multiple databases are coupled in the resulting system [5].

There are shortcomings in each of the above classifications.

Sheth and Larson [11, 3] only classify MDBSs by autonomy and assume that MDSs will be composed of multiple heterogeneous database systems. No provision is made for MDBSs composed of systems with homogeneous DBMSs. Since homogenous systems have fewer problems and need different handling, it is important that they be catered for in a taxonomy of MDBSs. Distribution is not referred to at all in this classification method.

Özsu and Barker [9] introduce a classification method which takes distribution, autonomy and heterogeneity into account, treating each of these as orthogonal issues. Three basic generic systems are presented: *composite systems* which have no autonomy; *federated systems* which are semi-autonomous, and *multidatabase systems* which are fully autonomous. The classification is presented on a graph with three axes (autonomy on the x axis, distribution on the y axis, and heterogeneity on the z axis) and seems to cater for degrees of each of these characteristics of MDBSs. However, distribution and heterogeneity are both seen as taking discrete values, i.e. an MDBS is either distributed or single-site; and the MDBSs are either homogeneous or heterogeneous. The autonomy dimension takes three discrete values in this taxonomy: non-autonomous, semi-autonomous and fully-autonomous.

There are several problems with Özsu and Barker's taxonomy. The issue of whether a global schema is defined for a particular implementation is not explicitly dealt with in the taxonomy. The terminology used is also ambiguous. For example, they refer to a *distributed homogeneous DBMS* which is defined as a distributed, homogeneous, non-autonomous MDBS. The DBMS in this system is not distributed — the component MDBSs are distributed and each has its own DBMS. This type of naming ambiguity persists throughout the taxonomy. The authors also do not make provision for the interoperable type of system which is incorporated into the new taxonomy.

Bright, Hurson and Pakzad [5] present a taxonomy of what they call *global information sharing solutions*. Their taxonomy classifies the systems according to how tightly the participating multiple databases are coupled in the resulting system. They present six discrete classes of systems and examine the method of integration of the systems. They also address the issue of whether a global schema is defined for the information sharing system.

There are also problems with their approach. Their taxonomy does not mention distribution at all and only addresses the autonomy and heterogeneity aspects of the systems, whereas the distribution of the multiple database comprising the system is relevant and should be catered for. The taxonomy also does not provide for semi-autonomous homogeneous systems.

A combination of these methods resulted in the alternative taxonomy of MDBSs described below and illustrated in Figure 5.

In the new taxonomy, distribution, heterogeneity and autonomy are considered to be orthogonal issues. The levels of the classification consider distribution first, then heterogeneity at the next level, followed by autonomy at the lowest level. Within each of these sublevels the different architectural alternatives are used as differentiators.

A decision was made not to incorporate distribution into the classification at a lower level, since distribution is orthogonal to the other issues addressed in this classification. Whether the MDBSs are all located on the same computer or distributed over a network or indeed the world, the problems in handling the system remain identical. The network details are handled by the underlying network operating system and do not require any special effort or consideration from the MDBS software. In order to cater for the distribution aspect, the system is first classified according to heterogeneity and autonomy. The classification is then prefaced with either *distributed* or *single-site* to distinguish these systems from one another.

We can now examine each of the architectural alternatives in turn. The first type of system to be considered is the *coalition of MDBSs*. In this type of system, also called a *tightly-coupled* system, the component database systems would have no autonomy whatsoever, thus being *non-autonomous*, and would probably not function independently even though they have the functionality to do so. The global system has total control over local data and processing. The system will create a global schema by integrating the schemas of all the participating MDBSs. A single image of the database would be made available to any user who wants to use the data in the multiple databases. From the users' perspective, the data is logically centralized in one database.

This type of system is described by Özsu *et al* [9] as a *tightly integrated multidatabase management system*. Because it is so tightly integrated, this type of system can closely synchronize global processing. Also, since the global system has complete control over local systems, processing can be optimized for global requirements. These systems have good global performance, but this is achieved at the cost of significant local modification and loss of control [5].

If our pharmacy example were to be implemented in this way, it would mean that all transactions would have to be entered at the MDBS layer — by the global user. There would be no local users in the system at all.

These systems can be constructed using either homogeneous or heterogeneous systems. The homogeneous option would be called a *homogeneous coalition of MDBSs* whereas the heterogeneous option would be called a *heterogeneous coalition of MDBSs*. The homogeneous option is often referred to as a *distributed database* system in the literature [7].

Next in the autonomy dimension is the *global schema MDBS*. This type of system integrates *semi-autonomous* multiple databases. These types of systems are more loosely coupled than the coalition of MDBSs described above, because global functions access local information through the external user interface of the local DBMS. However, the multiple database system software layer still maintains a global schema, also called the *global conceptual schema*, so the local sites must cooperate closely to maintain the global schema. These systems are typically designed bottom-up and can integrate pre-existing multiple databases without modifying them. This class of MDBSs is introduced by Bright *et al* [5].

Creating the global schema here is more difficult than in the coalition of multiple database type of system. The reason for this is that the DBA of the global system has no control over local schema input to the global schema, and the local schemas are not modified when they join the global schema MDBS.

In this type of MDBS, there are three extra levels on top of the ANSI-SPARC architecture [3]:

1. The *global conceptual schema* — This type of system is simply a logical view of **all** the data available to the MDBS. It is only a subset of the union of all the local

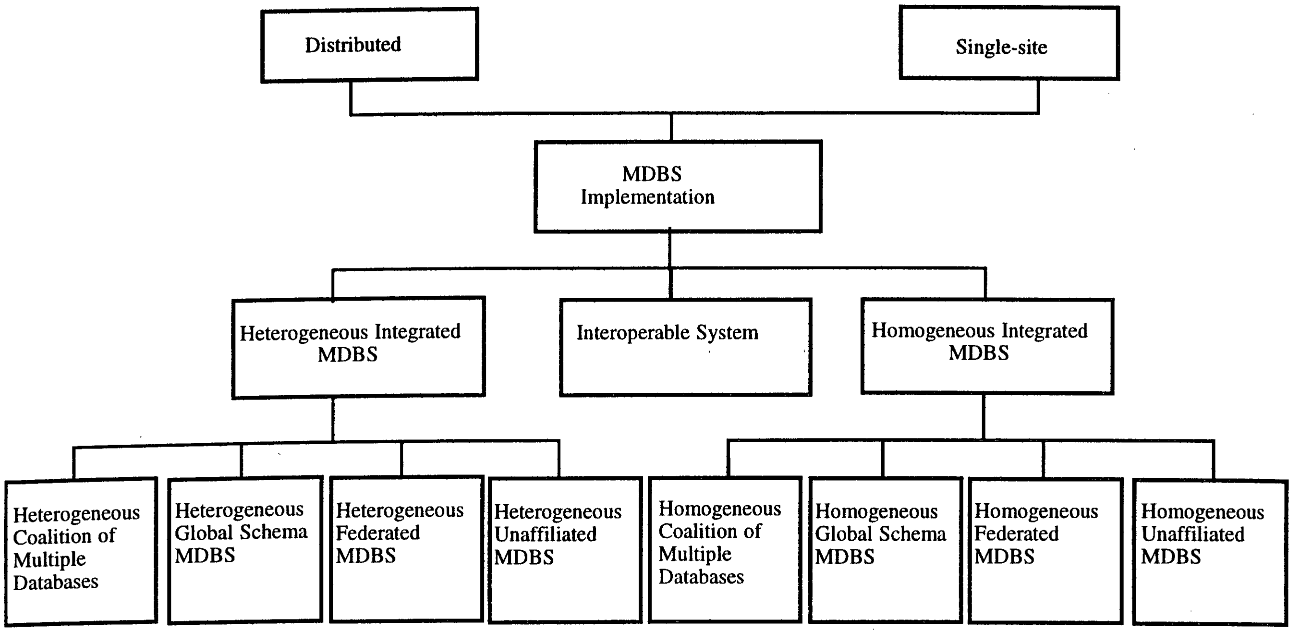

Figure 5. Classification of Multiple Database Systems

conceptual schemas, since the local DBMSs are free to decide what parts of their local databases they wish to contribute to the global schema.

2. The *participation schema* — A component database's participation in the multidatabase system is defined by means of a *participation schema* and represents a view defined over the underlying local conceptual schema.
3. The *global external views* — Support for user views, which is very important because the global conceptual schema will probably be extremely large, is provided by the *auxiliary schema.* The auxiliary schema describes the rules which govern the mappings between local and global levels. For example, some pharmacies may store liquid quantities in cubic units, while others use litres or ounces. Rules for handling null values may be necessary where one site stores additional information which is not stored at another site, for example one site stores the name, home address and home telephone number of its employees, whereas another just stores name and address.

Once again these systems can be either heterogeneous or homogeneous.

Still on the same autonomy level, the next classification is the *federated MDBS.* In this type of system, component database systems are also *semi-autonomous*, but here there is no global schema. The component database systems typically sacrifice some of their autonomy to become part of the MDBS. This would cause some additional difficulties in allowing databases to join or leave the system, but on the other hand, it would make concurrency and recovery maintenance easier to enforce.

These systems are once again more loosely coupled than the global schema MDBS. Each local system maintains a local *import* and *export schema* [9]:

1. The *export schema* is a description of the information the local node is willing to share with the global system.
2. The *import schema* is a description of the information from the other database systems that may be accessed locally. Each import schema is therefore a partial global schema.

Therefore, each local participating database system must cooperate closely only with the nodes it accesses in order to carry out some transaction. This is done by means of the import schema provided by the remote database system that needs to be accessed. User queries are restricted to local data and the data represented in the local import schema [5]. These systems can be constructed using either heterogeneous or homogeneous database systems.

This would be the best option for our pharmacy example MDBS, because the component database systems would retain their functionality, while allowing the global user access to most of the data as well.

A system with full autonomy is called an *unaffiliated MDBS*. Özsu *et al* refer to this system as the *total isolation* option [9]. This means that absolutely no changes are made to the DBMSs of the component database systems in order for them to be part of the MDBS.

This option is almost impossible to achieve if the user requires any update functionality at all. In this type of system, the processing of user operations is especially difficult, since there is no global control over the execution of transactions by individual DBMSs [9].

In these systems no global schema is maintained. The global system supports all global database functions by providing query language tools to integrate information from separate databases in the MDBS. User queries can specify data from the local schema of any of the individual databases participating in the system [5].

The construction of a global conceptual schema is a difficult and complex task and involves resolving both semantic and syntactic differences between sites [6]:

1. *Semantic* differences relate to the meaning and intended use of data. These include inconsistencies in the domain definitions for attributes, discrepancies in naming (synonyms and homonyms), data values and value precision.
2. *Syntactic* differences include differences in data types used (e.g. medicine registration number can be defined as a string or an integer), data manipulation operators (e.g. the set of operations in the CODASYL model which is equivalent to a join operation in the relational model) and units of measurement (e.g. litres or pints).

Sometimes these differences are so extensive that they do not warrant the huge investment involved in developing the global schema, especially when the number of multisite global queries is relatively low [3]. Researchers argue that this type of organization gives this system a significant advantage over non-autonomous type MDBSs [7]. The unaffiliated MDBS system typically has two layers, the *local database system layer* and on top of it the *multiple database system layer*. The local system layer consists of the component database systems. The responsibility for providing access to multiple heterogeneous databases is delegated to the mapping between the external schemas and the local conceptual schemas. This is fundamentally different from the global schema multiple database option, where this responsibility is taken over by the mapping between the global conceptual schema and the local ones.

This difference in responsibility has a practical consequence. If a global schema is defined, then the consistency rules of the global database can be specified according to this single definition. If a global schema does not exist, however, dependencies have to be defined between the various local conceptual schemas [9]. Unaffiliated systems can be built from either heterogeneous or homogeneous systems.

The last type of system is the *interoperable system*. These systems are even more loosely coupled than the unaffiliated systems and are fully autonomous. Global function is limited to simple data exchange and does not support full database functionality. Local systems may, but need not, include other types of information repositories, such as expert systems and knowledge-based systems [5].

Interoperable systems cannot be grouped under either heterogeneous or homogeneous MDBSs, but have a place in the taxonomy since they are definitely an implementation alternative for MDBSs.

Table 1 shows how the new taxonomy compares to, and embodies, the other given taxonomies.

4 Multidatabases

Recent research into MDBSs has concentrated on a group of systems called *multidatabase systems* (MDSs). Researchers, however, differ with respect to the basic definition of exactly what this term refers to. In some of the literature, full autonomy is required when an MDBS is referred to [9], and others require MDSs to be built from *distributed* MDBSs [5]. Still others include the non-autonomous, semi-autonomous and fully-autonomous options in their MDS definition [11].

Since all researchers agree that MDSs are composed of multiple *pre-existing* database systems, it is doubtful whether the coalition of MDBSs option (the non-autonomous type of system), would be a realistic option for an MDS. While the unaffiliated MDBSs would obviously be the ideal choice, Mullen *et al* [8] have proved that full autonomy is almost impossible to attain in an MDS. Therefore, the semi-autonomous options, namely the *global schema* and *federated* MDBSs, also have to be included in the definition of a MDS.

An MDS can thus be defined as a system which is composed of autonomous or semi-autonomous, heterogeneous or homogeneous, distributed or single-site, *pre-existing* databases, together with a software layer, which controls access to all data in the component databases from the point of view of the user of the MDS, while the component database systems still function independently.

A *multidatabase management system* is the software layer built on top of the MDBSs that facilitates access and manipulation of data at local sources (the component databases), by both users of the local databases, as well as users of the MDS.

The following can therefore be identified as MDSs:

1. *Global Schema* MDBSs,
2. *Federated* MDBSs,
3. *Unaffiliated* MDBSs.

5 Summary

The concepts of multiple database systems and multidatabases have been introduced. The characteristics which distinguish multiple database systems from one another have been outlined. The autonomy dimension has been identified as the characteristic which is the most important distinguishing feature of these systems, and a quantification method has been presented for measuring this dimension.

In section 3, a classification taxonomy was introduced that can be used for the precise definition of concepts and terms related to multiple database systems.

A special case of a multiple database system, namely the multidatabase has been defined and its positioning in the classification taxonomy has been indicated.

References

1. K Barker. *Transaction Management on Multidatabase Systems*. PhD thesis, University of Alberta, 1990.
2. K Barker. 'Quantification of autonomy on multidatabase systems'. *Journal of Systems Integration*, 4(2):151–169, (1994).

Table 1. Classification Alternatives

New taxonomy implementation	Classification by Özsu & Barker	Taxonomy of Sheth and Larson	Taxonomy of Bright *et al*	Autonomy
Heterogeneous coalition MDBS	Heterogeneous integrated DBMS	Nonfederated multidatabase system	—	None
Heterogeneous global schema MDBS	—	Tightly coupled single federation multidatabase system	Global schema Multidatabase	Semi-autonomous
Heterogeneous federated MDBS	Heterogeneous federated DBMS	Tightly coupled multiple federation multidatabase system	Federated database	Semi-autonomous
Heterogeneous unaffiliated MDBS	Heterogeneous multidatabase system	Loosely coupled multidatabase system	Multidatabase language system	Fully autonomous
Homogeneous coalition MDBS	Logically integrated homogeneous multiple DBMSs	—	Distributed database	None
Homogeneous global schema MDBS	—	—	—	Semi-autonomous
Homogeneous federated MDBS	Federated DBMS	—	—	Semi-autonomous
Homogeneous unaffiliated MDBS	Multidatabase system	—	Homogenous multidatabase language system	Fully autonomous
Interoperable system	—	Data exchange system	Interoperable system	Fully autonomous

3. D Bell and J Grimson. *Distributed Database Systems.* Addison Wesley, Great Britain, 1992.
4. Y Breitbart, H Garcia-Molina, and A Silberschatz. 'Transaction management in multidatabase systems'. In W Kim, ed., *Modern Database Systems.* Addison Wesley, New York, (1995).
5. M W Bright, A R Hurson, and S Pakzid. 'A taxonomy and current issues in multidatabase systems'. *Computer*, **25**(3):50–60, (1992).
6. D Georgakopoulos. *Transaction Management in Multidatabase Systems.* PhD thesis, University of Houston, 1990.
7. W Litwin. 'From database systems to multidatabase systems: Why and how'. In A R Hurson, M W Bright, and S Pakzad, eds., *Multidatabase Systems: An Advanced Solution for Global Information Sharing.* IEEE Computer Society Press, Los Alamitos, California, (1988).
8. J G Mullen, A K Elmagarmid, W Kim, and J Sharif-Askary. 'On the impossibility of atomic commitment in multidatabase systems'. In P A Ng, C V Ramamoorthy, L C Seifert, and R T Yeh, eds., *Proceedings of The Second International Conference on Systems Integration*, pp. 625–634. IEEE Computer Society Press, Los Alamitos, (1992).
9. M T Ozsu and K Barker. 'Architectural classification and transaction execution models of multidatabase systems'. *Lecture Notes in Computer Science*, **468**(5):285–294, (1990).
10. R R Rastogi. *Concurrency Control in Multidatabase Systems.* PhD thesis, University of Texas, 1993.
11. A Sheth and J Larson. 'Federated database systems'. *ACM Computing Surveys*, **22**(3):183–236, (1990).
12. A Silberschatz, M Stonebraker, and J Ullman. 'Database systems: Achievements and opportunities'. In M Stonebraker, ed., *Readings in Database Systems, Second Edition.* Morgan Kaufmann Publishers, San Francisco, California, (1994).

Received: 5/96, Accepted: 10/96, Final copy: 11/96.

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems, as well as shorter technical research notes. However, non-refereed review and exploratory articles of interest to the journal's readers will be considered for publication under sections marked as Communications or Viewpoints. While English is the preferred language of the journal, papers in Afrikaans will also be accepted. Typed manuscripts for review should be submitted in triplicate to the editor.

Form of Manuscript

Manuscripts for *review* should be prepared according to the following guidelines.

- Use wide margins and $1\frac{1}{2}$ or double spacing.
- The first page should include:
  - title (as brief as possible);
  - author's initials and surname;
  - author's affiliation and address;
  - an abstract of less than 200 words;
  - an appropriate keyword list;
  - a list of relevant Computing Review Categories.
- Tables and figures should be numbered and titled.
- References should be listed at the end of the text in alphabetic order of the (first) author's surname, and should be cited in the text in square brackets [1–3]. References should take the form shown at the end of these notes.

Manuscripts accepted for publication should comply with the above guidelines (except for the spacing requirements), and may be provided in one of the following formats (listed in order of preference):

1. As (a) LaTeX file(s), either on a diskette, or via e-mail/ftp – a LaTeX style file is available from the production editor;
2. As an ASCII file accompanied by a hard-copy showing formatting intentions:
   - Tables and figures should be original line drawings/printouts, (not photocopies) on separate sheets of paper, clearly numbered on the back and ready for cutting and pasting. Figure titles should appear in the text where the figures are to be placed.
   - Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin, if they are not clear in the text.

   Contact the production editor for markup instructions.
3. In exceptional cases camera-ready format may be accepted – a detailed page specification is available from the production editor;

Authors of accepted papers will be required to sign a copyright transfer form.

Charges

Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect typesetting, reproduction and other costs. Currently, the minimum rate is R30-00 per final page for LaTeX or camera-ready contributions that require no further attention. The maximum is R120-00 per page (charges include VAT).

These charges may be waived upon request of the author and at the discretion of the editor.

Proofs

Proofs of accepted papers in category 2 above may be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Note that, in the case of camera-ready submissions, it is the author's responsibility to ensure that such submissions are error-free. Camera-ready submissions will only be accepted if they are in strict accordance with the detailed guidelines.

Letters and Communications

Letters to the editor are welcomed. They should be signed, and should be limited to less than about 500 words.

Announcements and communications of interest to the readership will be considered for publication in a separate section of the journal. Communications may also reflect minor research contributions. However, such communications will not be refereed and will not be deemed as fully-fledged publications for state subsidy purposes.

Book reviews

Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

Advertisement

Placement of advertisements at R1000-00 per full page per issue and R500-00 per half page per issue will be considered. These charges exclude specialized production costs which will be borne by the advertiser. Enquiries should be directed to the editor.

References

1. E Ashcroft and Z Manna. 'The translation of 'goto' programs to 'while' programs'. In *Proceedings of IFIP Congress 71*, pp. 250–255, Amsterdam, (1972). North-Holland.
2. C Bohm and G Jacopini. 'Flow diagrams, turing machines and languages with only two formation rules'. *Communications of the ACM*, **9**:366–371, (1966).
3. S Ginsburg. *Mathematical theory of context free languages*. McGraw Hill, New York, 1966.

South African Computer Journal

Number 18, December 1996
ISSN 1015-7999

Suid-Afrikaanse Rekenaar-tydskrif

Nommer 18, Desember 1996
ISSN 1015-7999

Contents