

Adapting on-line analytical processing for decision modelling: the interaction of information and decision technologies

Nikitas-Spiros Koutsoukis^{*}, Gautam Mitra¹, Cormac Lucas²

Brunel University, Department of Mathematics and Statistics, Uxbridge, Middlesex, UB8 3PH, UK

Accepted 7 April 1999

Abstract

The introduction of new technologies and concepts has redefined the relative positioning of information systems (IS) and decision technologies in a corporate context. Corporate IS have been extended to include not only transaction processing databases but also analytical databases, often known as Data Warehouses. On-line analytical processing (OLAP), as introduced by Codd et al. [E.F. Codd, S.B Codd, C.T. Salley, Providing On-Line Analytical Processing to User-Analysts: An IT Mandate, E.F. Codd and Associates, 1993], is capable of capturing the structure of the real world data in the form of multidimensional tables which are known as ‘*datacubes*’ by management information systems (MIS) and statistical systems specialists. Manipulation and presentation of such information through multidimensional views and graphical displays provide invaluable support for the decision-maker. We illustrate the natural coupling, which exists between data modelling, symbolic modelling and ‘What if’ analysis phases of a decision support system (DSS). In particular, we explore the power of roll-up and drill-down features of OLAP and show how these translate into aggregation and disaggregation of the underlying decision models. Our approach sets out a paradigm for analysing the data, applying DSS tools and progressing through the information value chain to create organisational knowledge. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Decision-maker; OLAP; Multidimensional databases; Datacube; (Dis)aggregation; Information technology; Decision models; Analytic database; Information value chain; OR/MS models; Forecasting; EIS; DSS; Interfaces; Model granularity

1. Introduction

Management information systems (MIS), decision support systems (DSS) and executive information systems (EIS) represent a natural progression in information systems (IS) development. Their progress has also underpinned the convergence of IS and DSS. Their common characteristic is that they analyse and synthesise information from corporate data

[23,31,45]. For historical reasons MIS, DSS and EIS evolved as separate components of the Enterprise Information System considered from a decision support perspective.

On-line analytical processing (OLAP) is a recent advance in the field of IS for decision support. OLAP not only integrates the MIS, DSS, EIS, functionality of the earlier generations of IS, but goes further and introduces spreadsheet-like multidimensional data views [5,6] and graphical presentation capabilities. For a discussion of the recent developments of OLAP and its relationship with the corporate IS, we refer the reader to Refs. [3,11,50]. An illustration of the current relative positioning of IS

^{*} Corresponding author. E-mail:
nikitas-spiros.koutsoukis@brunel.ac.uk

¹ E-mail: gautam.mitra@brunel.ac.uk

² E-mail: cormac.lucas@brunel.ac.uk

and decision tools (DT) and their functional relationship is shown in Fig. 1.

In the past, corporate data was usually stored in production databases which are on-line transaction processing (OLTP) systems, also known as transaction databases. It is only recently that information specialists have realised the need to analyse the data and store it in a different form which can be better utilised for decision-making purposes [31,47]. The work by Codd et al. [5] has led to a distinction between analytic databases (ADBs) and operational or production databases (PDBs). A summary of these differences is presented in Ref. [50] which provides a clear delineation between the functionalities of OLTP and OLAP. We now illustrate this with a simple example. Consider a customer order processing application in which the transaction-oriented database stores information such as: the name of the customer, the customer's identification number, the customer's order reference number, the value and the date of the order, with links to other tables holding other relevant data (e.g., quantity and quality of items ordered, method/date of dispatch, etc.). A manager however, needs to analyse the data and requires the resulting information in a more succinct form, such as: total number of orders per day, total number of deliveries per day, or how many orders were dispatched without a delay, etc. It is much more important for the manager to have this information in a time-order manner especially to understand trends in customer behaviour. In traditional PDBs, such time-order in-

formation is not available. As a result, trends in customer behaviour cannot be identified which in turn denies the business decision-maker a valuable insight. By extending and analysing information from the PDB and storing it in the ADB in time-order, the information is readily available to executives for decision-making. Applying other criteria for analysing the data as found in the PDB leads to more extensive collections of information which have become known as data warehouses and data marts [3,7,16,17,22,41,44]. The complete set of ADBs created by these investigations is referred to as the corporate data warehouse. Any subset of the ADBs which relates to a particular department or a particular application is commonly known as a data mart.

The IS infrastructure as presented in Fig. 1 includes a connectivity layer (commonly known as Middleware) which is used to reorganise data and achieve cross-platform integration [26,42]. The middle and top layers in Fig. 1, consist of analytic tools, which comprise management science/operational research (MS/OR) models and data mining (DM)/knowledge discovery in databases (KDD) tools [8,15,36]. Finally, OLAP and EIS interface play a vital role of accessing and reporting knowledge or 'business intelligence' created by the system.

This process of analysing data, synthesising it into information and further refining it into knowledge is a well-known DSS process (for instance, see Ref. [14]). An alternative perspective of this process is set out in Fig. 2 which shows the information value

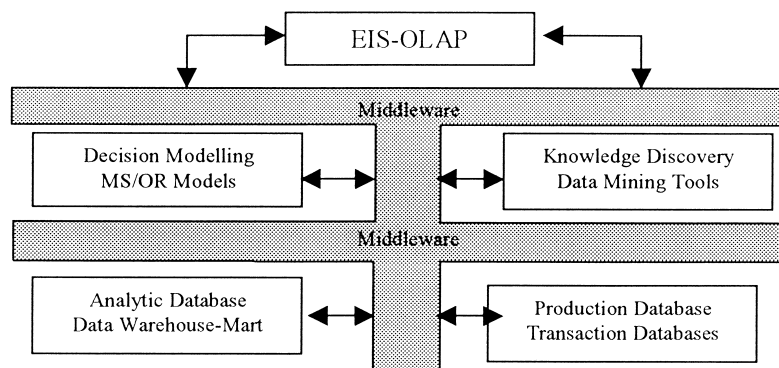


Fig. 1. Relative positioning of information and decision tools.

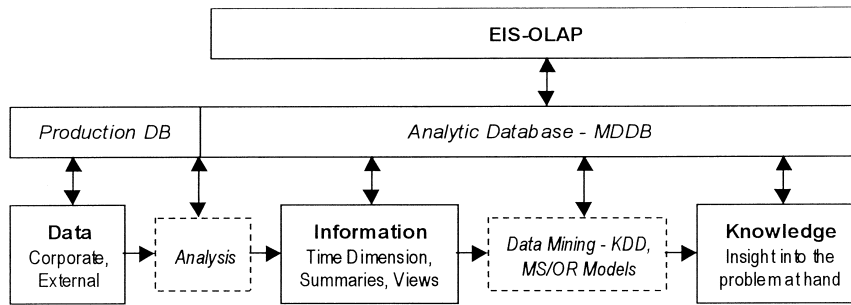


Fig. 2. Information value chain.

chain (IVC). A detailed description of these terms is provided as a glossary and appears in Appendix B.

The IVC model, illustrated in Fig. 2, works as follows: different transactional data, which is present in the PDB, is transferred to the analytic database, where it is stored in a form customised for the user and suitable for analysis (e.g., time-ordered, aggregated, etc.). This form also provides a ‘single version of the truth’. The data is then analysed and synthesised, resulting in information. Then the KDD tools and MS/OR models, are instantiated by the available information to produce knowledge (e.g., using optimisation, rule induction). The OLAP–multidimensional database (MDDB) has continued to gain acceptance in the DSS community because it places the analyst in direct contact with the data, models and the related analyses [46].

Thus, the analytic database achieves the following: (a) it stores data ranged over time; (b) it links otherwise disparate data items appearing as transactional data; (c) it provides a ‘single version of the truth’; (d) it provides accessibility to the data; and (e) it allows for customised user views of the data.

The components in our modelling paradigm use the same underlying structure, and are linked to each other as shown in Fig. 3. From the user–analyst’s

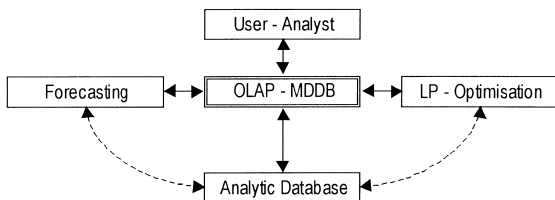


Fig. 3. Interaction between information and decision technologies.

point of view, OLAP and the MDDB serve as an interface. They connect the user with the analytic database, as well as with the analytic tools (MS/OR, KDD/DM). This interface is then used for a variety of purposes, including model formulation and fine-tuning, post-solution analysis and ADB customisation for a specific user. DOME [30] provides a prototype example of an MDDB/OLAP-based linear program (LP) model formulator and post-solution analysis reporter.

The OLAP–MDDB interface is used to assist the decision-maker in creating appropriate KDD and MS/OR models by browsing the appropriate data groups, and defining the model-based relations between these data groups. The models and their data are then passed to the appropriate computational tool for solution execution, through Middleware connectivity, and the solution is reported back to the decision-maker. This approach is useful to the user–analyst as there is no need to acquire expert knowledge for a particular MS/OR model or DM/KDD tool, and they can thus concentrate on interpreting the results, or carry out further analyses. It is assumed that the decision-maker is familiar with the functionality of the different models or tools (e.g., optimisation, rule induction tools) that may be implemented. Geoffrion [12] has developed the paradigm of structured modelling which provides an alternative but novel method of data information and knowledge representation. Raghunathan [39] has presented this as a modelling methodology for designing DSS.

The rest of this paper is organised in the following way: in Section 2, we introduce OLAP for the reader who is not familiar with the concept. We explain the structure of MDDBs and describe the

OLAP features in respect of viewing the data as well as manipulating the data items. Section 3 sets out the interaction of data models, decision models and resulting investigation. In this section, a business problem is first introduced. This is followed by modelling of the data, and illustrations of roll-up, drill-down and slice features of OLAP. These are subsequently exploited in the definition of decision models. In the formulation of the decision models, the natural coupling of the multidimensional structure with the symbolic models are illustrated. Subsequently, we expand on this ‘natural coupling’ by showing how roll-up, drill-down features of OLAP are used to create aggregated and disaggregated models. This section is concluded with the discussion of analysis and investigation of models and the use of multidimensional tables in capturing the results for multiple scenarios. In Section 4, we summarise our conclusions regarding the interaction of decision models and IS.

2. OLAP features: multidimensional data, viewing and manipulation

2.1. An introduction to OLAP

Introduced by Codd et al. [5], OLAP is an architecture used to provide users with the ability to

perform dynamic data analysis. By using OLAP tools, users gain access to the corporate analytical database. This type of access provides decision-makers with the potential to improve their understanding of corporate changes, their ability to identify or generate possible solutions for a variety of decision problems, and their ability to develop tactical and strategic formulae regarding corporate variables. This ability, to analyse and synthesise information from OLAP, is ‘made up of numerous, speculative ‘what if’ and ‘why’ data model scenarios’ [5]. In these model scenarios, key variables are changed, and the behaviour of organisational factors is then observed. The data models are then animated, and as the user synthesises information, he gains knowledge about the behaviour of corporate parameters, as presented by the model(s). According to Codd et al. [5], the models used in OLAP systems are one or more of the enterprise data models: categorical, exegetical, contemplative, formulaic. Their characteristics are summarised in Table 1.

A more familiar classification of models is the one that separates models into descriptive (instead of categorical), explanatory (instead of exegetical), prescriptive and normative [23]. The descriptive–explanatory–prescriptive–normative classification examines models in relation to the decision-maker’s actions, rather than their relation to corporate planning. However, conceptual similarities can be no-

Table 1
Enterprise data models

Categorical	Exegetical	Contemplative	Formulaic
<ul style="list-style-type: none"> • Static data analysis, use of historic data • Describes history of enterprise/events; shows state of enterprise 	<ul style="list-style-type: none"> • Static data analysis, use of historic data • Explains the state shown by categorical 	<ul style="list-style-type: none"> • Dynamic data analysis, scenario data • Projection of state according to introduction/ variance of parameters across the consolidation paths of the data models 	<ul style="list-style-type: none"> • Most dynamic of the models • Seeks a particular outcome through the model
<ul style="list-style-type: none"> • Little/no user interaction 	<ul style="list-style-type: none"> • User interaction required 	<ul style="list-style-type: none"> • Sufficient user interaction 	<ul style="list-style-type: none"> • Highest degree of user interaction
<ul style="list-style-type: none"> • Consolidation paths inherent in DB design 	<ul style="list-style-type: none"> • Consolidation paths not readily available, must be created on the fly 	<ul style="list-style-type: none"> • New consolidation paths must be created 	<ul style="list-style-type: none"> • Highest degree of variable data consolidation paths
<ul style="list-style-type: none"> • Data accessible via query tools, report writers, spreadsheets 	<ul style="list-style-type: none"> • Existing tools not adequate 	<ul style="list-style-type: none"> • Spreadsheet tools exist for single data dimensions. No adequate database support, or support for multidimensionality 	<ul style="list-style-type: none"> • No products exist

ticed between the two classifications, regarding the purpose(s) of the models.

OLAP was introduced together with 12 rules, which were thought to help eliminate misconceptions about OLAP's key characteristics. These 12 rules were set forward by Codd and Codd [6]:

1. *Multidimensional conceptual view* (discussed above). User's perspective of the enterprise is multidimensional; therefore, user's data view should be multidimensional. Views should be customisable by roll-up–drill-down–slice–dice functionality.
2. *Transparency*. Open systems architecture. The analytical tool can be embedded anywhere the user requires.
3. *Accessibility*. Users access data from heterogeneous data sources. OLAP must be able to adapt accordingly.
4. *Consistent reporting performance*. If performance decreases when model/data size increases, the user is affected by trying to overcome the performance problem, rather than performing his analysis as intended.
5. *Client/server architecture*. Data is stored in mainframes, so client/server is essential. Intelligent servers will allow multiple clients with minimum effort/modifications.
6. *Generic dimensionality*. Dimension-functionality should not be biased towards any dimension.
7. *Dynamic sparse matrix handling*. OLAP's physical data schema should adapt to data sparseness and distribution.

8. *Multi-user support*. OLAP is meant to be a strategic tool, so it must support concurrent user models.
9. *Unrestricted cross-dimensional operations*. Along consolidation paths, the tool should infer calculations. The tool must not restrict calculations between dimensions.
10. *Intuitive data manipulation*. Data manipulation should be accomplished by direct action upon cells of the analytical model.
11. *Flexible reporting*. Data or synthesised information must be presented according to any orientation the user requires.
12. *Support for unlimited number of dimensions and aggregation levels*. In practice, however, most tools support a limited number of dimensions, which, nevertheless, is adequate for most purposes.

2.1.1. Other perspectives on OLAP

OLAP, as presented above is a powerful concept. However, its introduction was also accompanied with the evaluation of commercial products to assess whether they were appropriate OLAP tools. As a result, software and consulting added their own 'rules', or redefined OLAP's aim to suit a particular product. For example, the Gartner Group added nine further rules and IRI software added three more rules [4]. One of the most important and most quoted additional rules is 'incremental database refresh'. This enables the database to be updated only in the area where changes have taken place. A further

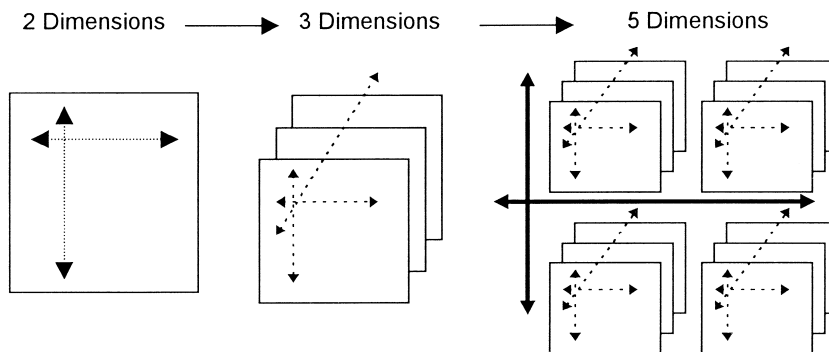


Fig. 4. Multidimensional viewing.

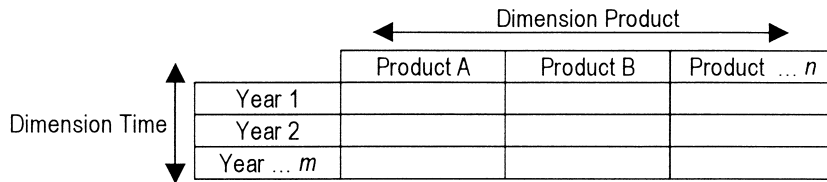


Fig. 5. Two-dimensional view.

extension to this efficient processing of data is the intelligent handling of duplicate data as discussed in Ref. [3]. We do not discuss the details for these additional rules or refinements, but we note that OLAP, at an application level, is very much product-specific, and varies from implementation to implementation.

From studying the characteristics of OLAP, one concludes that multidimensional viewing would perform best if the data were stored in MDDBs. MDDBs have performance advantages when the data is dense. In addition, Codd, when introducing OLAP emphasised the fact that relational systems at that time were inadequate and lacked depth compared to OLAP in terms of analytical functionality. Nevertheless, since the introduction of OLAP, many have argued in favour of relational-OLAP (ROLAP), i.e., OLAP tools based on relational database technology and this view is supported by vendors of relational database systems (e.g., Ref. [27]).

In practice, ROLAP systems have a relational-based ADB and use a special software layer for providing multidimensional data viewing (MDV), incorporating OLAP functionality. OLAP systems that use MDDBs do not use such layers, as the analytic-MDDB provides the multidimensional views by default. Performance differences, advantages or

disadvantages are considerable for both OLAP and ROLAP but are beyond the scope of this paper.

Whether the actual data storage is relational or multidimensional is irrelevant for the decision-maker. What the decision-maker needs, and what OLAP must provide, are the following functions [6]:

1. Access to the data in the database management system (DBMS).
2. Data and data consolidation paths (or dimensions) that can be defined according to the user requirements.
3. Accommodation for the variety of ways or different contexts in which the user may wish to view, manipulate and animate the data analyses.
4. Accessibility to these functions via the end-user's interface.

MDDBs are better suited for OLAP-type applications because of their structure and embedded functionality.

2.2. MDV

MDV or data viewing across multiple dimensions, is a key OLAP feature. In general, a 'dimension' is a data category (e.g., product, district, time). This category may have one or more different characteristics which are the 'dimension values' (e.g., products: A,

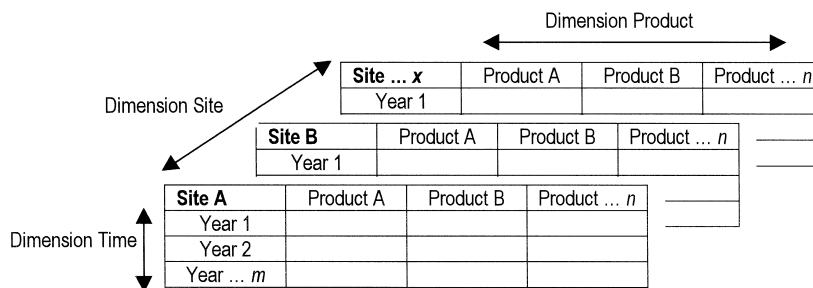


Fig. 6. Three-dimensional view.

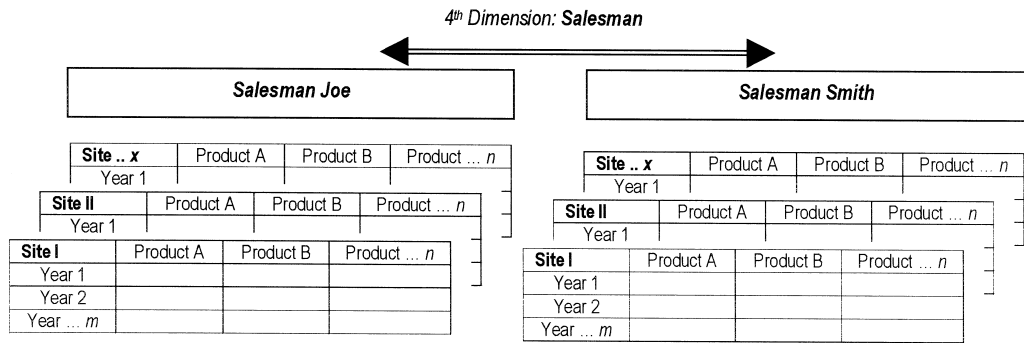


Fig. 7. A four-dimensional view.

B, and C; district: north, east, west, south; time: 1990–1996). A data value in a multidimensional table then reflects a particular business or user perspective (e.g., sales) for a corresponding combination of dimension values, or sets of dimension values (e.g., sales for product A, district south, for all years).

In relational database terminology, dimensions correspond to the ‘attributes’ of a relational table, while the dimension values correspond to the attribute’s ‘domain’ (or the set of possible attribute values). The multidimensional data table values correspond to the ‘tuples’ or ‘records’ of a relational table [22,28]. Multidimensional data tables reflect a specific business view or perspective on the data (e.g., sales by region, and product, over time). The benefit is that the data fits the user’s perspective and not vice versa [1,5,6,20,21,37].

One example to visualise this multidimensional viewing is a spreadsheet. A single spreadsheet is two-dimensional (i.e., one dimension being the columns, the other dimension being the rows). A ‘stack’ of spreadsheets is three-dimensional. This structure is more commonly known as a ‘cube’ or a ‘hypercube’. A stack of cubes viewed along a new

dimension is considered four-dimensional. Along a further dimension is five-dimensional, and so on for all dimensions employed in a multidimensional table (Fig. 4).

For example, suppose a decision analyst wishes to analyze data concerning some ‘products’, over some period of ‘time’, for some ‘salesmen’ for some particular sales ‘sites’. In dimension terms, the data dimensions are product, time, salesman and site. A two-dimensional view of time and product is seen in Fig. 5.

Considering the third dimension (e.g., site) produces a ‘stack’ of time–period tables (one for each site), as in Fig. 6.

The addition of another dimension (in this case, salesman), along one axis, creates a four-dimensional view (Fig. 7). It is easy to create a five-dimensional view by adding another dimension along another axis. However, it is difficult to visualise more than four dimensions at a time.

It is important to understand that these data-views are user-configurable. Dimensions are not statically set in columns or rows, or along one axis or another. To use the OLAP terminology, the user has the ability to *roll-up*, *drill-down*, *slice*, *dice* or *nest* his

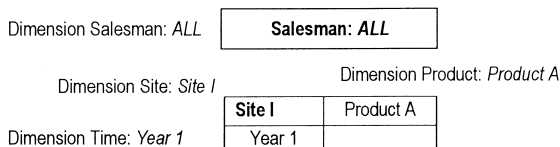


Fig. 8. Slice.

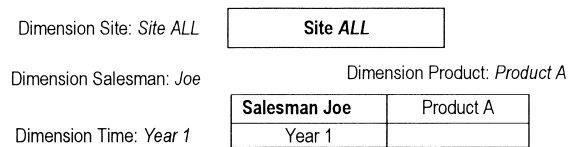


Fig. 9. Dice.

Dimension Product: *Product A*
Dimension Time: *Years 1 - m*

Dimension Salesman: *ALL*

	Salesman <i>ALL</i>	Product A		
		<i>Year 1</i>	<i>Year 2</i>	<i>Year ... m</i>
Dimension Site: <i>Sites I, II</i>	Site I			
	Site II			

Fig. 10. Pivoting and nesting.

data. We classify these functions as multidimensional data ‘viewing’ and as multidimensional data ‘manipulation’ functions. The viewing functions then are slice, dice and nest. The manipulation functions are drill-down and roll-up.

Slice is any two-dimensional ‘slice’ of the data. In Fig. 8, the data slice reveals information for site ‘I’, product ‘A’, and year ‘1’, for ‘all’ salesmen. Slice is the ‘dropping’ of the dimension being ‘sliced’.

Dice is the ‘rotation’ of the hypercube to reveal another, different slice of data (i.e., the revealing of data for a different set of dimension values). In Fig. 9, the new ‘dice’ has dimension site as its most aggregate dimension, products are now in columns, time is in rows, and salesman is ‘Joe’. *Pivot* and *rotate* are identical to dice, and were adopted from spreadsheet terminology. Dice, rotate and pivot are now used synonymously [32].

Nesting is the ability to ‘nest’ dimensions (i.e., display values from one dimension within another dimension). In Fig. 10, the product dimension is pivoted to the columns and dimension time is now nested by the dimension product.

Multidimensional viewing is not a new concept, and much of its functionality is already present in spreadsheets and EIS. However, multidimensional viewing has received much attention because of its close relationship with OLAP and its strong direct relationship with databases.

There are two multidimensional data manipulation functions: drill-down and roll-up. *Drill-down* is the exploration of data to subsequent levels of more detail along a dimension. In Fig. 11, Year ‘1’ has been drilled down to quarters 1, 2, 3 and 4. *Roll-up* is the aggregation of data to subsequent levels of summary, along a dimension. This implies that dimensions are typically hierarchical in nature based on parent/child relationships between dimension values. These relationships are normally data summations, but, in principle, other meaningful operations such as average can also be used. In Fig. 12, the quarters have been rolled up to year. Roll-up, drill-up, consolidate and aggregate are used synonymously in OLAP terminology [32].

Multidimensional viewing as discussed above has the power to represent multiple conceptual dimensions in a two-dimensional arrangement, as high-

Dimension Product: *Products A - n*

Dimension Site: *Site I*

Dimension Time:
Year 1 → Quarters

Site I		Product A	Product B	Product ... n
Year 1	Quarter 1			
	Quarter 2			
	Quarter 3			
	Quarter 4			

Fig. 11. Drill-down.

Dimension Site: <i>Site l</i>	Dimension Product: <i>Products A - n</i>			
	Site l	Product A	Product B	Product ... n
Dimension Time: Quarters → Year 1	Year 1			

Fig. 12. Roll-up.

lighted in Refs. [18,19]. Detailed discussion of OLAP, and multidimensional viewing terms can be found in Refs. [2–6,10,32,38,50], among others.

3. Interaction of data models and DSS models

Data modelling, decision modelling, model investigation, are logical constructs with a vital role in both the Interaction of IS and DT, and in rational decision-making. It is easily seen that data modelling and decision modelling closely interact with each other. The following describes their logical constructs and how they derive knowledge.

1. *Data modelling*. This is how to extract recorded facts, internal or external, which provide the decision-maker with the facts about the decision problem that he is facing.
2. *Decision modelling*. This is the creation of a model that represents a simplified representation of a real system. It is used to evaluate possible actions and the probable outcomes of these actions.

3. *Model analysis and investigation*. As the future is uncertain, ‘what if’-type investigations on data and model parameters assist the decision-maker to prepare for the ‘extreme’ future.

Typically, data modelling involves defining relationships between data items leading to a relational data model, or identifying categories which are then used to define multidimensional tables. The classical decision models are usually linear, non-linear, or discrete optimisation models. However, we consider cluster analysis, forecasting, DM and other analytic models equally appropriate as decision models [36]. A descriptive analysis of the results obtained from the decision model is applied to gain insight, or knowledge in respect of a given decision problem. In particular, a series of ‘what if’ questions in the analysis phase are used to evaluate the decision-maker’s understanding of the problem by changing parameters. We view the various software layers and the user interaction as shown in Fig. 13 (see also Ref. [30]).

The diagram also highlights the various constituents involved in the overall decision process and their participation at each system level. At the outer level, the decision-maker uses the system to guide him in his decisions. At the inner level, the database and model specialists are responsible for maintaining and developing their appropriate components. This is analogous to the concept of different views put forward by Greenberg and Murphy [13].

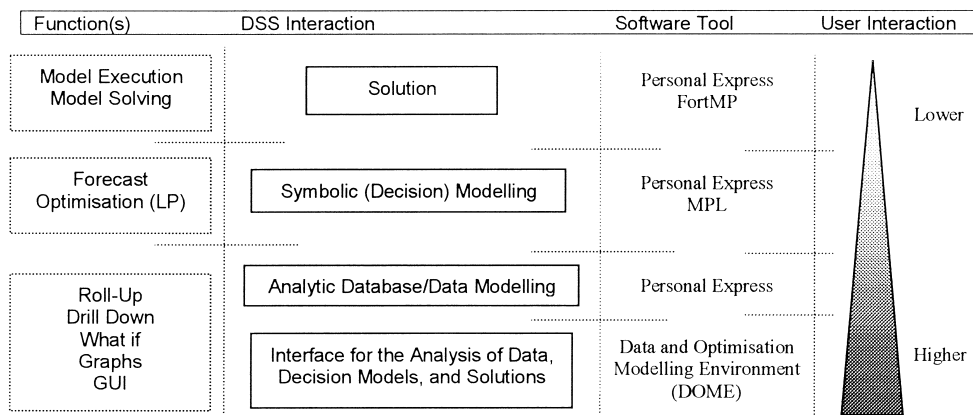


Fig. 13. Implemented software architecture.

Table 2
The modelling dimensions of the decision problem

Dimension	Index	Description
Time	$i = 1,2$	Denotes the time periods associated with production
Modes	$j = 1,2$	Denotes the production modes
Products	$k = 1, \dots, K$	Denotes the products
Machines	$g = 1, \dots, G$	Denotes the machines
Sales area	$o = 1, \dots, O$	Denotes the sales areas
Site	$e = 1,2$	Denotes the sites

3.1. An illustrative example

In this section, we illustrate how data modelling and decision modelling interact taking into account a closely related pair of forecasting and LP-optimisation models. Our research group was involved earlier in creating a modelling system, UIMP [9], in which the multidimensional tables and LP model structures were strongly coupled. We have taken the current example from that earlier paper.

A Company uses three different Machines (A, B, C), to produce three products (Nuts, Bolts and Washers) in two Production Modes (Normal, Overtime), in two sites (InTown, OutOfTown). Annual production is split into two periods (Summer, Winter). The final products are distributed to different sales areas (North, East, South, West). In order to plan production, the management uses past sales data to forecast demand. The resulting forecast is used to plan the production schedule for the forthcoming year.

Table 3
Production times (two-dimensional summary representation of a five-dimensional table)

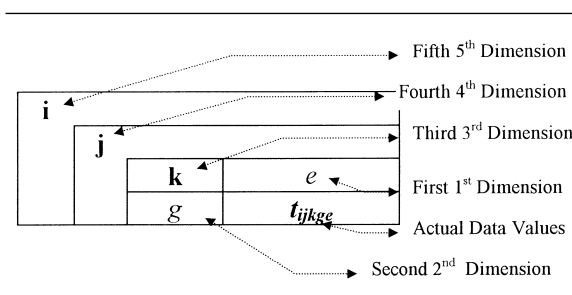


Table 4
Machine availability

i	j	e
	g	a_{ijge}

To investigate the decision problem described, we first create a forecasting model that generates data input for our decision model. The decision model in this example is a linear programming-based optimisation model for production planning.

The ‘target knowledge’ from the combination of these two models, is to:

- (I) Use the forecast as an indication of demand for the generation of an optimum production plan.
- (II) Identify if the production plan will satisfy our forecasted demand, and to indicate where, how, and by how much, demand is not met.

3.2. Data modelling

In our example, the information sets are utilised from the company’s PDB. Items considered in our ADB are:

- *Company structure information:* production time-horizon, products, production modes, machines, sites and sales areas.
- *Resources information:* production times, machine availability, storage availability, product demand.
- *Financial information:* production costs, storage costs, shipment costs and actual sales over the past few years.
- *Decision-making information:* level of production, level of storage, level of shipments and demand-shortage level.

Table 5
Storage capacity

	e
k	h_{ke}

Table 6
Demand

i	<i>o</i>
<i>k</i>	d_{iko}

This information is collected from various databases throughout the organisation. For example, storage availability is collected from databases at the warehouse; shipment costs are taken from the distribution/dispatch department database, and so on.

The available company information fits naturally in a multidimensional analytic database. As we transfer the above information sets to our analytic database, the company structure information forms the modelling dimensions, or indexes in mathematical terms. These are summarised in Table 2.

The ‘sales area’ dimension is a hierarchical dimension. Its most aggregate level is ‘all areas’, which represents the total for all sales areas; in this case $O = 1$ (i.e., roll-up). When investigating individual sales area (i.e., drill-down), $O =$ number of sales areas. The use of this dimension (and hierarchical dimensions in general) will be demonstrated in detail later.

Once the dimensions are defined, the company resources, financial and decision-making information sets are easily represented in multidimensional tables in the analytic database. The dimensions provide a ‘natural way’ to capture the existing real-world information structure within a particular company perspective. The multidimensional tables, which represent the resources, financial and decision-making information sets, are described and presented in tabular format. For the detailed datasets, the reader is referred to Appendices A–C. The MDDB used is ORACLE Personal Express (Version 5.01, Refs. [33,34]).

Table 7
Selling price

	<i>k</i>
<i>i</i>	p_{ik}

Table 8
Storage cost

	<i>e</i>
<i>k</i>	s_{ke}

3.2.1. Resources information

Production times (t_{ijkge}): represents the number of hours (t), needed by machine (g) to produce a single unit of product (k), in a given production mode (j), at site (e), in time period (i) (Table 3).

Machine availability (a_{ijge}): represents the number of hours machine (g) is available (a), in production mode (j), at site (e), in time period (i) (Table 4).

Storage capacity (h_{ke}): represents the maximum number of units that can be stored (h), for product (k), at site (e) (Table 5).

Demand (d_{iko}): represents the number of units of product (k) demanded (d), at sales area (o), in time period (i) (Table 6).

In the decision models, demand is represented in two data tables with the same structure as the one presented here. The forecast model uses the actual sales observations over i, k, o and the decision model uses the outcome of the forecast (i.e., the forecasted demand), also over i, k, o .

3.2.2. Financial information

Selling price (p_{ik}): represents the unit price (p) of product (k), in time period (i) (Table 7).

Storage cost (s_{ke}): represents the unit storage cost (s), for product (k), at site (e) (Table 8).

Production costs (c_{ijkge}): represents the unit production cost (c), for product (k), production mode (j), machine (g), at time period (i), at site (e) (Table 9).

Table 9
Production cost

i	j	k	<i>e</i>
		<i>g</i>	c_{ijkge}

Table 10
Shipment cost

i		
	k	e
	o	m_{ikoe}

Shipment costs (m_{ikoe}): represents the unit shipping cost (m), for product (k), from site (e) to sales area (o), in time period (i) (Table 10).

3.2.3. Decision information

These tables store information about our decisions. Initially, these values are unknown but after optimisation they are refreshed with their solution values.

Production (x_{ijkge}): represents the amount of product (k) produced (x), in production mode (j), by machine (g), at site (e), in time period (i) (Table 11).

Storage (y_{ike}): represents the amount of product (k) stored (y), at site (e), in time period (i) (Table 12).

Shipment (z_{ikeo}): represents the amount of product (k) shipped (z), from site (e), to sales area (o), in time period (i) (Table 13).

Demand shortage (w_{iko}): represents the shortage (w) of sales area's (o) demand not met (w), for product (k), in time period (i) (Table 14).

3.2.4. The introduction of time

The ease with which the multidimensional data model captures the structure of the real-world data is easily seen. An additional requirement of the ADB and a regular feature of OLAP-based databases, is the capturing of the time dimension. In most MDDBs, time is a 'built-in', fully customisable (i.e., definition of 'customised time periods' is possible),

Table 11
Production

i	j	k	e
		g	x_{ijkge}

Table 12
Storage

i	e
k	y_{ike}

hierarchical dimension (e.g., year, quarter, month, week, day).

In our example, we use the dimension 'year' to represent the years 1993 to 1999. For each year, we have information split into two seasons (summer/winter). Thus, a hierarchical type dimension year:time is adequate for representing our information. We, however, have combined the dimensions 'year' and 'time' to create a 'conjoint dimension' which is particularly efficient for managing sparse data. This is advantageous when a given year (in our example) does not have data for a particular season thus avoiding inefficient data storage in such a situation. Obviously, our data is not sparse, and using a conjoint dimension provides no efficiency advantage, but this does demonstrate the ability of MDDBs to provide customized dimensions. Thus, the elements of observed demand can be found as an entry in the form $d_{(year,i)ko}$ where (year, i) is an access index for the conjoint dimension. For simplicity and clearer explanation of the mathematical models, we refer back to the notation d_{iko} where i represents the time dimension.

Observed demand (d_{iko}): represents the amount of product k sold in i in sales area o . (The multidimensional table for forecasted demand is of identical structure but holds values only when a forecast is produced. The database names for observed demand and forecasted demand are OLD.DEMAND and FCST.DEMAND, respectively.) (Table 15)

The incorporation of time is not only important for our decision problem example, but also an essential element of most organizational decision problems. It is therefore beneficial to have the ability to

Table 13
Shipment

i	k	o
	e	z_{ikeo}

Table 14
Demand shortage

i	o
k	w_{iko}

capture the time dimension with relative ease, as is the case for the multidimensional data model and OLAP-based databases.

3.2.5. Example of OLAP features

Having captured the data in an analytic database, the data items can be investigated using OLAP functionality which provides valuable insight into company information. For example, the table observed demand, can provide information concerning the product sales figures for each year in either summary or detailed form (see Fig. 14). The actual data tables are shown in Fig. 15.

In this case, the roll-up function is the summation over the index ‘o’. Index ‘o’ is the new aggregated index with the value ‘all areas’ instead of ‘north, ..., west’. Other functions such as the average of all the dimension values over the dimension index can also be used. For instance, an ‘averaging’ roll-up has been used for our cost data. Using the slice feature, the demand for a product *k*, at sales area (south) *o* = *O*, in season *i*, is obtained by slicing the demand table (Fig. 16).

Slicing involves ‘fixing’ a given dimension and looking at the reduced subset of data table entries where the dimensionality of a table is reduced by one. If applied progressively, we reach individual table entries where the dimensions *i*, *k*, *o* are fixed (i.e., *d_{iko}*).

3.3. Decision modelling

The real world structure of the data is easily transferred to the model structure. As mentioned in

Table 15
Observed demand

i	k
o	d_{iko}

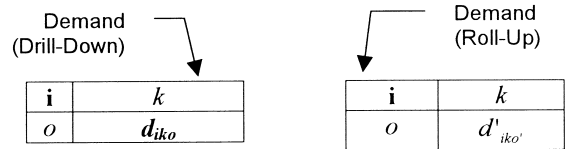


Fig. 14. The demand table in drill-down and roll-up modes.

the Section 3.2, in order to plan for production, demand is forecasted using historic sales data and the resulting forecast is used in creating a future production schedule. Thus, two models are implemented: a forecasting model for estimating future demand, and an LP-optimisation model for solving the production planning problem. Both these models employ the historical demand data as well as the structure of the data as presented in multidimensional tables.

3.3.1. The forecasting model

Our choice of a forecast model is based on the assumption that the sales data exhibits both a seasonal and a linear trend. Winter’s method of forecasting [48,49] was chosen as an appropriate model. The general model is shown below.

The forecast *y* developed at time *t* for a period *T* time units into the future is given by:

$$y_{t+T} = (a_t + b_t T) F^* \tag{1}$$

where *a_t* is the estimate of the current intercept between the trend line and the slope, and is given by:

$$a_t = \alpha \left(\frac{x_t}{F_{t-N'}} \right) + (1 - \alpha)(a_{t-1} + b_{t-1}), \tag{2}$$

b_t is the estimate of the slope of the trend line:

$$b_t = \beta(a_t - a_{t-1}) + (1 - \beta)b_{t-1} \tag{3}$$

F_t is the estimate of the multiplicative seasonal factor:

$$F_t = \sigma \left(\frac{x_t}{a_t} \right) + (1 - \sigma)F_{t-N'} \tag{4}$$

x_t is the actual observation at time *t*, *N'* is the number of observations comprising the periodicity of the data, *F_{t-N'}* is the estimate of the seasonal factor *N'* periods in the past, *F** denotes our best estimate of the seasonal factor in period *t* + *T* and α, β, σ are exponential smoothing constants, where $0 < \alpha, \beta, \sigma < 1$.

Note that Eqs. (2)–(4) are utilised prior to the actual forecast equation (Eq. (1)). The periodicity of

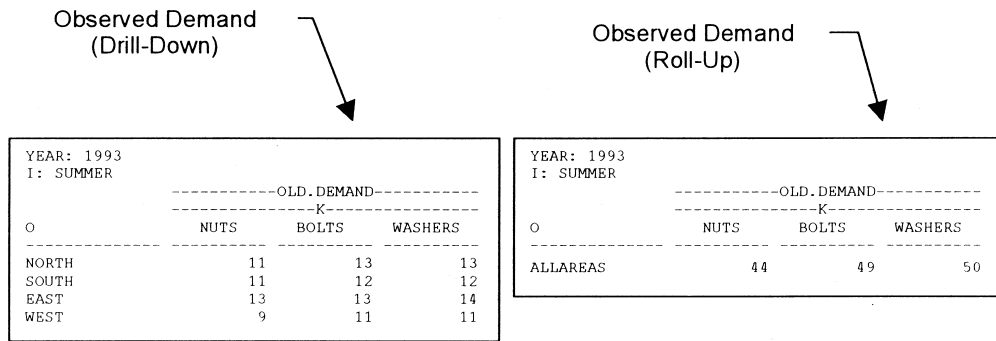


Fig. 15. The (observed) demand data table in roll-up and drill-down modes.

the data N' in our model is two time periods, and the forecast is for $T = 2$ time periods into the future.

In this forecasting model, we have used the notation y_t for demand in order to present the relationship in a clear and simple form. In the analytic database, however, the demand quantities are forecast and stored in categorized form using the symbol d_{iko} . We therefore use y_t as an alias for d_{iko} and, similarly, x_t is an alias for observeddemand $_{iko}$. Thus, $y_t \equiv d_{iko}$ and $x_t \equiv \text{observeddemand}_{iko}$, that is actual sales.

Most established MDDBs include built-in forecasting tools, as indeed does Personal Express. For details of the implementation of this model and its output, the reader is referred to Appendices A–C. Once the forecast results are obtained, they are stored in the database as a demand table. The LP-optimisation model then accesses data items in this table.

3.3.2. The decision model: LP-optimisation

Having obtained an estimate for the forthcoming year's demand, an LP production planning model is formulated, which when solved, provides values for our decision variables, that is, a production schedule for the forthcoming year.

This model is implemented using the beta test software DOME [30]. In brief, DOME uses a MDDB (Personal Express) in order to formulate a structured LP in the MPL language [25]. Further details of the DOME formulation can be found in the work of Mousavi et al. [30], while the resulting MPL code is set out in Appendices A–C.

3.3.3. Model definitions Indices (dimensions)

- $i = 1,2$ Denotes different time periods
- $j = 1,2$ Denotes different production modes
- $k = 1, \dots, K$ Denotes different products
- $g = 1, \dots, G$ Denotes different machines
- $o = 1, \dots, O$ Denotes different sales areas
- $e = 1,2$ Denotes different sites

Data

- t_{ijkge} The amount of time to produce one unit of k at site e , on machine l , in mode j , in time period i
- a_{ijge} The amount of time available on machine g , at site e , in time period i , and mode of production j

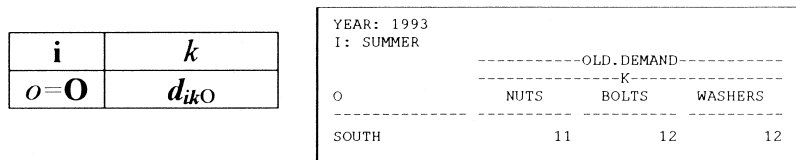


Fig. 16. A slice of the demand table.

p_{ik}	Unit selling price of product k , in time period i
d_{iko}	(Forecasted) demand for product k , at sales area o , in time period i
s_{ke}	Unit storage cost of product k , at site e
h_{ke}	Storage capacity of product k , at site e
c_{ijkge}	Unit production cost in time period i , for product k , in production mode j , for machine g , at production site e
m_{ikoe}	Unit shipment cost in time period i , for product k , from site e , to sales area o

Decision variables

x_{ijkge}	Amount of product k produced in time period i , in production mode j , by machine g , at site e
y_{ike}	Amount of product k stored in time period i , at site e
z_{ikeo}	Amount of product k sent from production site e , to sales area o , in time period i
w_{iko}	Amount of sales area's o demand, for product k , not met in time period i .

3.3.4. Model

3.3.4.1. Objective function.

Maximise profit

$$\begin{aligned} & \sum_i \sum_k \sum_o \sum_e p_{ik} z_{ikeo} \\ & - \sum_i \sum_j \sum_k \sum_g \sum_e c_{ijkge} x_{ijkge} \\ & \sum_i \sum_k \sum_o \sum_e m_{ikoe} z_{ikeo} \\ & - \sum_i \sum_k \sum_e s_{ke} y_{ike} - 15 \sum_{iko} w_{iko} \end{aligned}$$

where 15 is the unit cost for not meeting demand.

Subject to:

Machine availability:

$$\sum_k t_{ijkge} x_{ijkge} \leq a_{ijge} \quad \forall i, j, g, e.$$

Inventory balance:

$$\sum_j \sum_g x_{ijkge} - y_{ike} - \sum_o z_{ikeo} = 0 \quad \forall k, e; i = 1$$

$$\sum_j \sum_g x_{2jkge} + y_{1ke} - y_{2ke} - \sum_o z_{2keo} = 0$$

$$\forall k, e; i = 2;$$

Demand:

$$\sum_e z_{ikeo} + w_{iko} \geq d_{iko} \quad \forall i, k, o.$$

Bounds:

Storage:

$$y_{ike} \leq h_{ke} \quad \forall i, k, e.$$

Non-negativity

$$x_{ijkge}, y_{ike}, z_{ikeo} \geq 0 \quad \forall i, j, k, g, e, o.$$

3.4. Model analysis and investigation: exploiting OLAP features

We discuss three investigations that are based on applying the OLAP features to the model data. First, we instantiate different model inputs by using a top-level dimension, 'case'. Second, we achieve a varying level of detail in the application of the models by using a hierarchical dimension. Third, by using the slice functionality, we have the ability to investigate sub-problems by instantiating the appropriate subsets of data. Because of the natural coupling between the model structure and the data structure, each of these three investigations is a result of direct manipulation of the database; the models are not altered in any way for any of these investigations.

3.4.1. Scenario investigations

It is well known in the decision support community that a decision-maker or problem owner investigates future uncertainties by performing one, or more, of 'what if?' queries. A series of such models are variously known as 'cases' or scenarios. It has been suggested, by a number of investigators [46], that

<p>CASE: OPTIMISTIC I: SUMMER</p> <p style="text-align: center;">--DEMAND-- --O-- -----</p> <p>K ALLAREAS</p> <hr/> <p>NUTS 46.78 BOLTS 60.36 WASHERS 59.38</p> <p>CASE: OPTIMISTIC I: WINTER</p> <p style="text-align: center;">--DEMAND-- --O-- -----</p> <p>K ALLAREAS</p> <hr/> <p>NUTS 48.59 BOLTS 53.83 WASHERS 55.28</p>	<p>CASE: EXPECTED I: SUMMER</p> <p style="text-align: center;">--DEMAND-- --O-- -----</p> <p>K ALLAREAS</p> <hr/> <p>NUTS 46.71 BOLTS 65.65 WASHERS 62.38</p> <p>CASE: EXPECTED I: WINTER</p> <p style="text-align: center;">--DEMAND-- --O-- -----</p> <p>K ALLAREAS</p> <hr/> <p>NUTS 48.91 BOLTS 61.89 WASHERS 61.56</p>	<p>CASE: PESSIMISTIC I: SUMMER</p> <p style="text-align: center;">--DEMAND-- --O-- -----</p> <p>K ALLAREAS</p> <hr/> <p>NUTS 49.44 BOLTS 82.19 WASHERS 74.31</p> <p>CASE: PESSIMISTIC I: WINTER</p> <p style="text-align: center;">--DEMAND-- --O-- -----</p> <p>K ALLAREAS</p> <hr/> <p>NUTS 53.09 BOLTS 82.08 WASHERS 76.68</p>
---	---	---

Fig. 17. Forecast demand for each of the three scenarios (roll-up).

‘case’ should be considered an additional dimension in the model data tables and the solution tables. Although this expands the volume of information, OLAP-enabled tools make it possible to display this information in graphical form, and therefore help the problem owner in analysing the results. Note, however, that the models must be solved for the corresponding number of cases before the built-in ‘what if?’ functionality of an MDDB can be used.

The multidimensional data model naturally provides a ‘what if’ dimension, often referred to as ‘case’ or ‘scenario’. This enables version control as illustrated by Ramirez et al. [40]. This ‘case’ dimension is best used as the leading dimension. As the top-level dimension, it then ‘drives’ the lower-level dimensions and the data models can handle ‘what if’ investigations naturally, without the need for special software functionality.

Different scenarios are represented in the data structures and the resulting model instances, by the ‘case’ dimension. Each value of the ‘case’ dimension represents a different data scenario. For illustration purposes, we have considered three different cases for the purpose of forecasting demand: optimistic, average and pessimistic (Fig. 17). For each case, we change the forecast parameters as appropriate. The corresponding forecast outputs for each case are presented in Appendices A–C. Similarly, changes in resources can be considered as different cases in the production-planning model. Although this is not illustrated here, the same principle applies.

3.4.2. Granularity, aggregation–disaggregation

Roll-up and drill-down reflect the ‘granularity’ of the data the decision-maker wishes to view or inves-

tigate. By coupling the data to the model, we can create instances of aggregated and disaggregated models (Fig. 18). The results (output) of the aggregated and disaggregated models can be supplied side by side with the aggregated and disaggregated data (input). This functionality brings a number of benefits to the problem owner.

When constructing large-scale models analysts tend to work with less detail (*‘coarse grain model’*) in order to investigate the model behaviour and gain confidence in the model. This is also the case when investigating the feasibility of new policies. After model confidence is gained, the analysts then use progressively more and more detail (*‘fine grain model’*) in order to gain insight into the problem and assist the decision-makers to make more informed

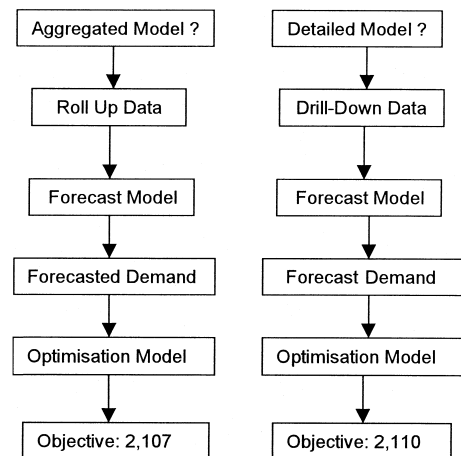


Fig. 18. The level of data detail drives the level of the model detail.

decisions. Indeed our results indicate that changing the model ‘grain’ by varying level of detail in the data does produce results that are very useful in the context we just described, as shown in Fig. 20 (results for the other two cases are in Appendices A–C). This indicates that the aggregated models can be used as good approximations to the detailed models.

In the detailed mode (drill-down over sales areas), all three different demand cases are computed by taking into consideration each individual sales area. In summary form (roll-up over sales areas), the forecast cases are the aggregation of demand for all the sales areas. The aggregated forecast and the detailed forecast are independent of each other, hence, on creating summary reports of the detailed model, these are not necessary the same as the results found in the aggregated model (Fig. 19).

In our example, consider the various managers within a company. The higher level management require the summary forecast, as an indication of how the company is going to perform in the immediate future, or to examine the feasibility of a new policy. The distribution managers, however, are more

likely to be interested in the detailed forecast, in order to estimate the workload of their department. In both cases the same computer model formulation, and data are used, while OLAP ensures the appropriate level of detail is accessed. Rogers et al. [43] consider another obvious reason for model aggregation, which is to reduce the ‘computational burden’ of solving the model.

3.4.3. Slice and sub-problems

The use of the ‘slice’ feature is useful when the problem owner requires sub-problem investigation. Consider a multi-national company for the purposes of our example. We could then use an additional dimension to represent the different countries where our company has production and distribution facilities. Slicing the country dimension for a particular nation results in creating a sub-problem for that particular division of the company.

Often, models are broken down into sub-problems (‘selective grain models’) in order to investigate the behaviour of specific levels of detail or summary within the problem (e.g., for a single product in all outlets). Similar work has been popular in the past

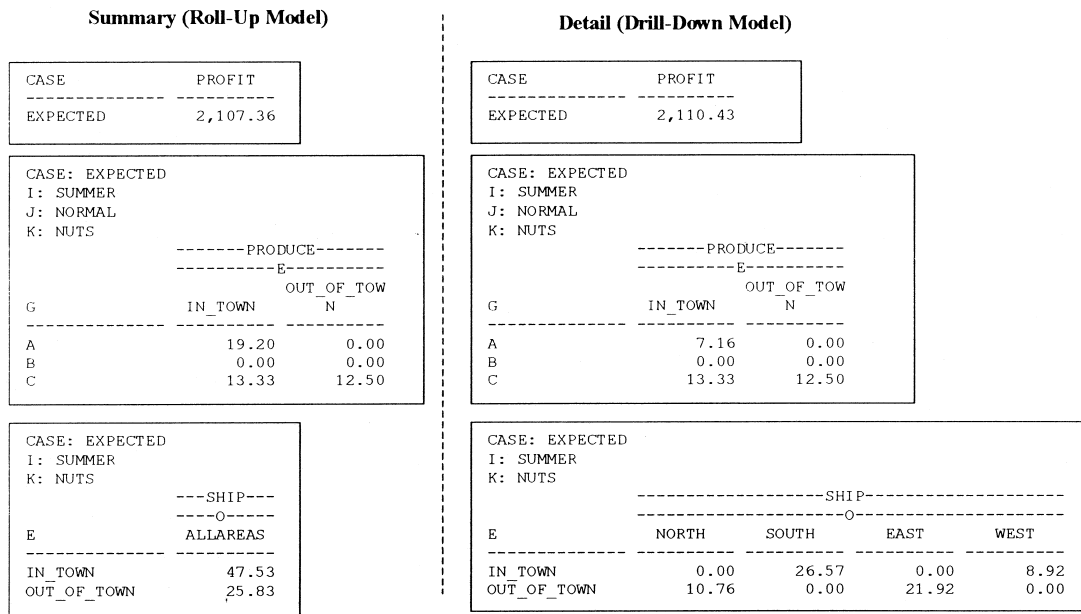


Fig. 19. Expected case: roll-up and drill-down forecasts.

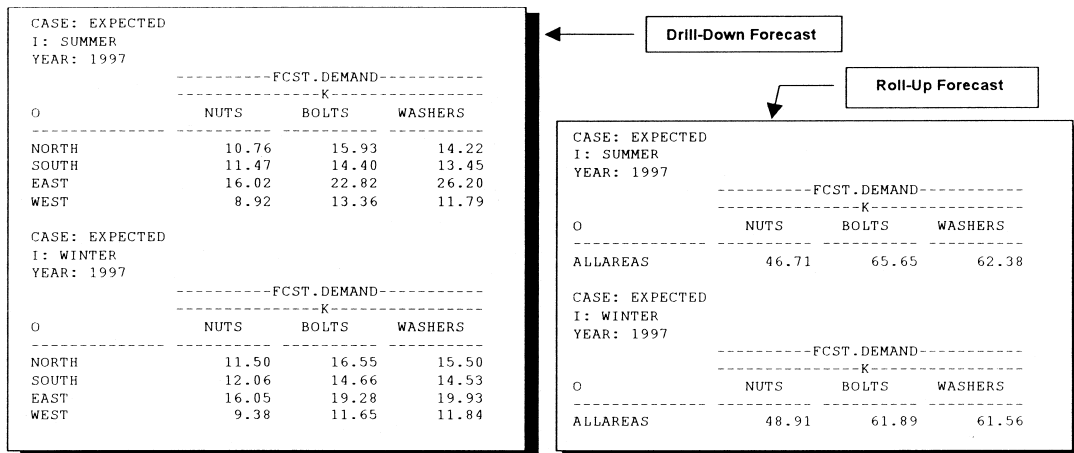


Fig. 20. Results from the aggregated and detail model.

especially with oil companies. PLATOFORM is an example of an earlier generation, sophisticated system with analogous functionality [35]. We have not explicitly shown how the slice feature can be used as a means of creating sub-problems, but the reader can easily visualise the procedure.

4. Summary and conclusions

In this paper we have first analysed different aspects and the emerging concept of IS which are used to capture and represent 'organisational knowledge'. We have analysed data warehousing and its relation to OLTP Databases. We have presented OLAP and multidimensional viewing and we have also considered the relative positioning of a number of interrelated tools such as Middleware and DM.

Our real interest is, however, to position decision modelling and illustrate its interaction with IS. We show how OLAP-enabled databases can be adapted to support symbolic modelling and used in the context of the IVC, and how modelling contributes towards acquiring corporate knowledge.

Thus, we have highlighted the strong coupling that exists between structure within organisational data and the structure of the symbolic models which are used in decision-making. In particular, we have shown that the drill-down and roll-up concepts introduced by commercial information specialists can be

naturally extended through aggregation and disaggregation into the domain of decision models.

The work of Möhring et al. [29] highlights the contribution from MS and IS are equally important in the creation of DSS. We have also put special emphasis in bringing together these two different worlds of specialisation. Namely the information systems designer, and the model-based management science specialist. In the context of organisational decision-making, these groups have important contributions to make. We believe in our work, we have shown how the specialisation of these groups can be brought together to create valuable decision support tools.

Appendix A. Role of different constituents

Database specialist: the person responsible for implementing and structuring the analytic database. Liaises with the model specialist to make sure the structure in the data is consistent with the structure of the decision model.

Model specialist: the person(s) responsible for the creating the models used in the decision process (e.g., forecasting, LP). Liaises with the database specialist to ensure that the structure in the decision model is consistent with that of the database model.

Decision-maker: the person who owns the decision problem and communicates his requirements to

the model specialist and the database specialist. He also participates in defining the investigative steps and uses the results of the system to guide him in his decision-making.

Appendix B. Glossary of terms

ADB or analytical database. A database that has special features that make it suitable for use for decision-making purposes. A distinct characteristic of the ADB is its incorporation of time within its data structure(s). The ADB gets its data from the transaction database.

Analytic tools. By analytic tools, we refer to the models, or techniques available to the decision-maker. These models or techniques are useful for the decision process, and include optimisation models or DM techniques.

EIS or executive information systems. IS built to suit the particular requirements of top corporate management. Main subjects of IS research and applications in the 1980s, and were characterised by their intuitive graphical-based interface, and their fast response times for ‘ad-hoc’ type queries, and ‘what if’ analyses. Roll-up and drill-down functionality was also considered an essential EIS feature.

OLAP or on-line analytical processing. A term coined by Codd, to represent the latest generation of EIS or decision support-based information systems. Closely coupled with a multidimensional data-viewing interface, OLAP is characterised by powerful data browsing functionality, and the ability to create and animate analytic data models.

ROLAP or relational on-line analytical processing. A reaction to the assumption that OLAP is only possible via the use of MDDBs. Essentially OLAP with a relational database.

MDDB or multidimensional database. MDDBs (commonly mentioned in conjunction with OLAP) store data along multiple dimensions. Dimensions are consolidation or aggregation paths that ‘fit’ the user-analyst’s perspective of the data. MDDBs do not need to be normalised for storage efficiency as is the case with relational databases. When MDDBs are coupled with an OLAP interface, they prove useful in performing multidimensional analyses [1,20,37]. This coupling of OLAP and MDDBs is becoming the de facto standard for analytical databases.

Data mining. See knowledge discovery databases.

Data warehouse. Commercial term for denoting the ADB. Indicates the central storage or ‘warehousing’ of data, so that it can be used for a variety of purposes including decision-making and KDD investigations. Data stored in a data warehouse, and therefore the ADB is assumed to be ‘cleansed’ of various inconsistency errors, such as misspellings, missing values, etc.

Data mart. Originally used as a synonym for the data warehouse. Data marts actually represent a collection of data that is a subset of the data stored in a data warehouse. Data marts tend to be application or department oriented.

MDV or multidimensional data viewing. The term we use to denote the ability to view, use and manipulate data sources in a multidimensional way. MDV can be used independently of the data store, and whether it is a MDDB or a specially configured relational database.

Middleware. Rock-Evans [42] characterises Middleware as an “off the shelf connectivity software which: supports distributed data processing and is used by developers to build distributed applications.” Overall, Middleware applications can be classified into one or more of these groups: translational (e.g., between applications), managerial (e.g., for certain tasks such as security, delivery, timing), a means of communication (e.g., between different network protocols, etc.) and a transport provision (e.g., routing a process through a network, hiding physical details from user). For example, the connectivity between a database, spreadsheet, e-mail, desktop publishing, and the operating system software is handled by Middleware software, whether the applications are in the same environment, or distributed across a network. Such Middleware examples include open data base connectivity—ODBC, or object-linking and embedding—OLE.

Management science (MS) / operational research (OR) models. MS/OR models include optimisation, decision analysis and simulation, among others (Mitra, 1988; Ref. [23]). These models have a strong mathematical component, are often used in a corporate context, and provide valuable insight to the decision-maker for various decision problems such as production planning, resource allocation, timetabling, crew scheduling and manpower plan-

ning. Such models are usually large, because of their scale of implementation. Most of the research issues in these areas focus on model formulation and especially algorithmic solver capability.

KDD/DM tools. KDD or DM tools is used to identify hidden or previously unknown relationships between corporate data. This is a generic name given to otherwise a diverse set of ‘software’ techniques such as chi-square tests (statistics), neural nets, genetic algorithms, data analysis through visualisation, rule induction and intelligent text search.

Appendix C. Illustrative tables of data instance, and aggregated and disaggregated results

C.1. Data instance

C.1.1. Database structure

See Fig. 21.

C.1.2. Data tables

The tables that are not affected by the different case dimension values are shown only once (Fig. 22).

The demand table is first populated by the forecast and then used in the LP (Figs. 23–29).

For the ‘shipment cost’ table below, the ‘AL-LAREAS’ values show the *average* transportation costs from production site *e* to all sales areas *o* (Fig. 30).

C.2. Results data tables

In this section, we present a selection of LP-optimisation results for each case. These results demonstrate the close performance between the summary and detailed representations of our production-planning model. Variables with 0 (zero) values after optimisation are not presented. For a detailed list of data tables, database automation routines and model implementation (Forecast and LP), see Ref. [24].

C.2.1. Expected case

See Fig. 31.

C.2.2. Optimistic case

See Fig. 32.

C.2.3. Pessimistic case

See Fig. 33.

C.3. Models and OLAP automation routines

C.3.1. Forecast model: express code

Personal express commands and keywords are shown in CAPITALS.

10 DIMENSIONs	20 VARIABLEs	5 PROGRAMs	1 RELATION
CASE	COSTOR	AREAS . PROGRAM	O . TOTAL
CONSTVALUES	DEMAND	CONJDIMS . PROGRAM	
E	DEMSHIP	DEMAND . PROGRAM	
G	FCST . DEMAND	DETAIL . PROGRAM	
I	OLD . DEMAND	SUMMARY . PROGRAM	
I . YEAR	PRICE		
J	PRODCOST		
K	PRODUCE		
O	PROFIT		
YEAR	SHIP		
	SHORTAGE		
	STOCKB1		
	STOCKB2		
	STORCAP		
	STORE		
	TIMEAVL		
	TIMECON		
	TIMEUS		
	TRANCST		

Fig. 21. The contents of the database.

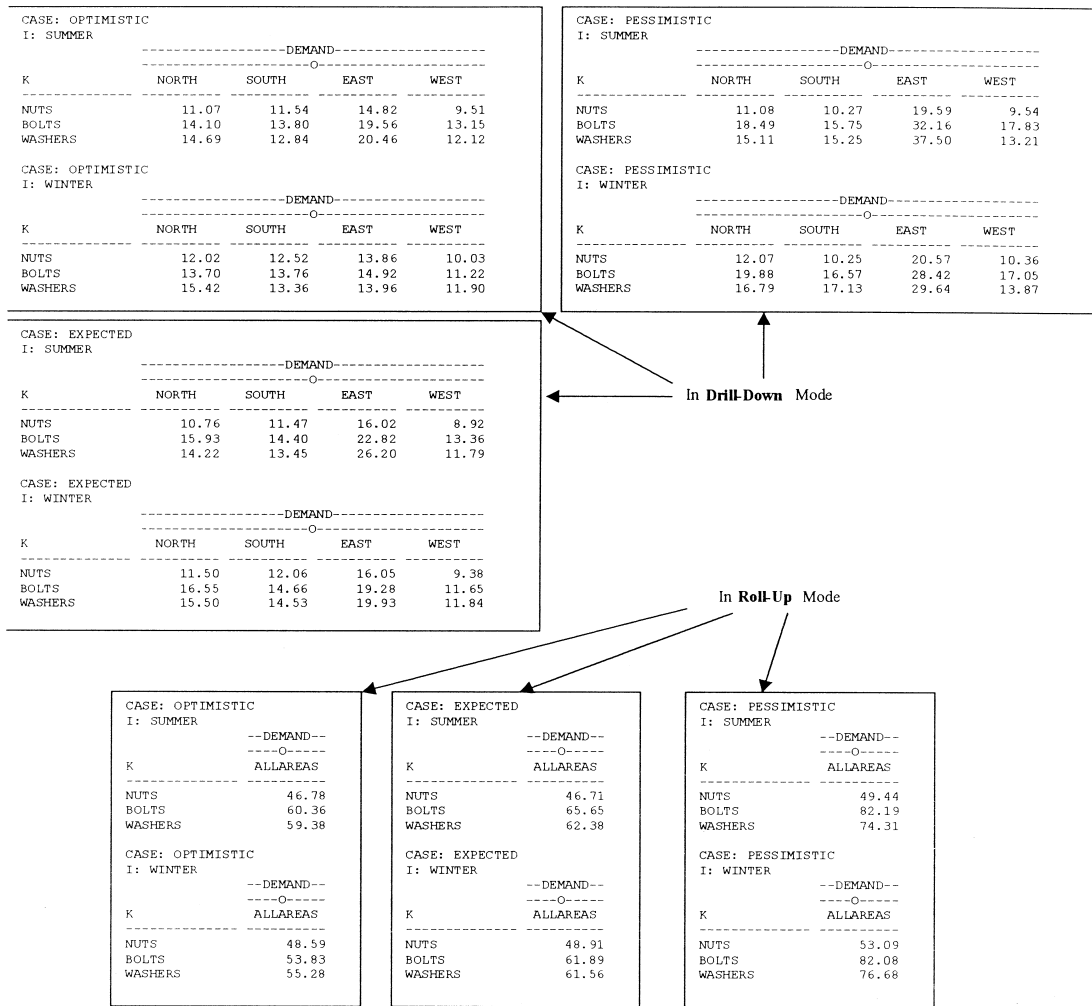


Fig. 22. The (forecasted) demand table (DEMAND).

C.3.1.1. Listing 1. Program for forecasting demand.

```
DEFINE DEMAND.PROGRAM PROGRAM
PROGRAM
```

"Program for Forecasting demand for two-time periods

CASE: CASE1		
-----COSTOR-----		
-----E-----		
K	IN_TOWN	OUT_OF_TOW N
NUTS	1.50	1.30
BOLTS	1.50	1.30
WASHERS	1.50	1.30

Fig. 23. The storage costs table (COSTOR).

"alpha-beta-gamma are selected according to the status

"of the case dimension

"Temporary variable declaration

VARIABLE product TEXT

VARIABLE area TEXT

OKFORLIMIT = YES

FOR k o

DO

"Perform the forecast

product = k

area = o

LIMIT i.year TO FIRST 8 "Setting of the forecast base data

<p>YEAR: 1993 I: SUMMER</p> <pre> -----OLD.DEMAND----- K O NUTS BOLTS WASHERS ----- NORTH 11 13 13 SOUTH 11 12 12 EAST 13 13 14 WEST 9 11 11 ALLAREAS 44 49 50 </pre> <p>YEAR: 1993 I: WINTER</p> <pre> -----OLD.DEMAND----- K O NUTS BOLTS WASHERS ----- NORTH 13 11 11 SOUTH 13 12 12 EAST 11 9 9 WEST 11 11 11 ALLAREAS 48 43 43 </pre>	<p>YEAR: 1994 I: SUMMER</p> <pre> -----OLD.DEMAND----- K O NUTS BOLTS WASHERS ----- NORTH 12 11 13 SOUTH 11 12 12 EAST 13 15 15 WEST 12 12 13 ALLAREAS 48 50 53 </pre> <p>YEAR: 1994 I: WINTER</p> <pre> -----OLD.DEMAND----- K O NUTS BOLTS WASHERS ----- NORTH 13 10 13 SOUTH 12 12 12 EAST 11 10 8 WEST 10 10 11 ALLAREAS 46 42 44 </pre>
<p>YEAR: 1995 I: SUMMER</p> <pre> -----OLD.DEMAND----- K O NUTS BOLTS WASHERS ----- NORTH 12 12 13 SOUTH 12 13 12 EAST 13 15 15 WEST 10 17 12 ALLAREAS 47 57 52 </pre> <p>YEAR: 1995 I: WINTER</p> <pre> -----OLD.DEMAND----- K O NUTS BOLTS WASHERS ----- NORTH 12 12 16 SOUTH 13 12 12 EAST 12 10 9 WEST 11 10 12 ALLAREAS 48 44 49 </pre>	<p>YEAR: 1996 I: SUMMER</p> <pre> -----OLD.DEMAND----- K O NUTS BOLTS WASHERS ----- NORTH 11 13 13 SOUTH 12 13 12 EAST 13 15 16 WEST 9 11 11 ALLAREAS 45 52 52 </pre> <p>YEAR: 1996 I: WINTER</p> <pre> -----OLD.DEMAND----- K O NUTS BOLTS WASHERS ----- NORTH 12 15 15 SOUTH 12 14 14 EAST 15 17 17 WEST 10 12 12 ALLAREAS 49 58 58 </pre>

Fig. 24. Old demand (past sales figures).

LIMIT *k* TO product
LIMIT *o* TO area
"Determine what is the case dimension value
"then do the appropriate forecast
SWITCH VALUES(case STATUS)
DO

CASE 'OPTIMISTIC':
FORECAST LENGTH 2 METHOD
WINTERS PERIODICITY 2 -
ALPHA 0.1 BETA 0.1 GAMMA 0.1 -
TIME *i*.year FCNAME fcst.demand
old.demand

CASE: CASE1	-----PRICE-----		
	-----K-----		
I	NUTS	BOLTS	WASHERS
SUMMER	10.00	10.00	9.00
WINTER	11.00	11.00	10.00

Fig. 25. Selling prices (PRICE).

CASE: CASE1	-----STORCAP-----	
	-----E-----	
K	IN_TOWN	OUT_OF_TOWN
NUTS	30.00	30.00
BOLTS	30.00	30.00
WASHERS	30.00	0.00

Fig. 26. Storage capacity table (STORCAP).

<p>CASE: CASE1 I: SUMMER J: NORMAL K: NUTS</p> <pre> -----PRODCOST----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 2.00 2.00 B 4.00 4.00 C 2.00 1.00 </pre>	<p>CASE: CASE1 I: SUMMER J: NORMAL K: BOLTS</p> <pre> -----PRODCOST----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 3.50 3.00 B 3.50 3.00 C 0.00 0.00 </pre>	<p>CASE: CASE1 I: SUMMER J: NORMAL K: WASHERS</p> <pre> -----PRODCOST----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 3.50 4.00 B 3.50 3.00 C 0.00 0.00 </pre>
<p>CASE: CASE1 I: SUMMER J: OVERTIME K: NUTS</p> <pre> -----PRODCOST----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 4.00 4.00 B 6.00 6.00 C 3.00 3.00 </pre>	<p>CASE: CASE1 I: SUMMER J: OVERTIME K: BOLTS</p> <pre> -----PRODCOST----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 5.00 5.00 B 5.00 5.00 C 0.00 0.00 </pre>	<p>CASE: CASE1 I: SUMMER J: OVERTIME K: WASHERS</p> <pre> -----PRODCOST----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 6.00 6.00 B 4.00 4.00 C 0.00 0.00 </pre>
<p>CASE: CASE1 I: WINTER J: NORMAL K: NUTS</p> <pre> -----PRODCOST----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 3.00 3.00 B 5.00 5.00 C 3.00 2.00 </pre>	<p>CASE: CASE1 I: WINTER J: NORMAL K: BOLTS</p> <pre> -----PRODCOST----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 5.00 5.00 B 5.00 5.00 C 0.00 0.00 </pre>	<p>CASE: CASE1 I: WINTER J: NORMAL K: WASHERS</p> <pre> -----PRODCOST----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 5.00 6.00 B 3.00 4.00 C 0.00 0.00 </pre>
<p>CASE: CASE1 I: WINTER J: OVERTIME K: NUTS</p> <pre> -----PRODCOST----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 5.00 5.00 B 7.00 7.00 C 4.00 4.00 </pre>	<p>CASE: CASE1 I: WINTER J: OVERTIME K: BOLTS</p> <pre> -----PRODCOST----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 6.00 5.00 B 6.00 5.00 C 0.00 0.00 </pre>	<p>CASE: CASE1 I: WINTER J: OVERTIME K: WASHERS</p> <pre> -----PRODCOST----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 7.00 6.00 B 5.00 5.00 C 0.00 0.00 </pre>

Fig. 27. Production costs table (PRODCOST).

<p>CASE: CASE1 I: SUMMER J: NORMAL</p> <pre> -----TIMEAVL----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 100.00 110.00 B 100.00 110.00 C 40.00 50.00 </pre>	<p>CASE: CASE1 I: WINTER J: NORMAL</p> <pre> -----TIMEAVL----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 110.00 100.00 B 110.00 100.00 C 50.00 40.00 </pre>
<p>CASE: CASE1 I: SUMMER J: OVERTIME</p> <pre> -----TIMEAVL----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 80.00 90.00 B 90.00 100.00 C 30.00 40.00 </pre>	<p>CASE: CASE1 I: WINTER J: OVERTIME</p> <pre> -----TIMEAVL----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 90.00 80.00 B 100.00 90.00 C 40.00 30.00 </pre>

Fig. 28. Machine availability table (TIMEAVL).

CASE: CASE1 I: SUMMER J: NORMAL K: NUTS -----TIMEUS----- -----E----- G IN_TOWN OUT_OF_TOWN ----- A 4.00 5.00 B 7.00 8.00 C 3.00 4.00	CASE: CASE1 I: SUMMER J: NORMAL K: BOLTS -----TIMEUS----- -----E----- G IN_TOWN OUT_OF_TOWN ----- A 5.00 6.00 B 6.00 7.00 C 0.00 0.00	CASE: CASE1 I: SUMMER J: NORMAL K: WASHERS -----TIMEUS----- -----E----- G IN_TOWN OUT_OF_TOWN ----- A 5.00 6.00 B 6.00 7.00 C 0.00 0.00
CASE: CASE1 I: SUMMER J: OVERTIME K: NUTS -----TIMEUS----- -----E----- G IN_TOWN OUT_OF_TOWN ----- A 3.00 4.00 B 6.00 7.00 C 2.00 3.00	CASE: CASE1 I: SUMMER J: OVERTIME K: BOLTS -----TIMEUS----- -----E----- G IN_TOWN OUT_OF_TOWN ----- A 4.00 5.00 B 5.00 6.00 C 0.00 0.00	CASE: CASE1 I: SUMMER J: OVERTIME K: WASHERS -----TIMEUS----- -----E----- G IN_TOWN OUT_OF_TOWN ----- A 5.00 5.00 B 5.00 6.00 C 0.00 0.00
CASE: CASE1 I: WINTER J: NORMAL K: NUTS -----TIMEUS----- -----E----- G IN_TOWN OUT_OF_TOWN ----- A 5.00 5.00 B 8.00 6.00 C 4.00 0.00	CASE: CASE1 I: WINTER J: NORMAL K: BOLTS -----TIMEUS----- -----E----- G IN_TOWN OUT_OF_TOWN ----- A 6.00 5.00 B 7.00 6.00 C 0.00 0.00	CASE: CASE1 I: WINTER J: NORMAL K: WASHERS -----TIMEUS----- -----E----- G IN_TOWN OUT_OF_TOWN ----- A 7.00 6.00 B 7.00 6.00 C 0.00 0.00
CASE: CASE1 I: WINTER J: OVERTIME K: NUTS -----TIMEUS----- -----E----- G IN_TOWN OUT_OF_TOWN ----- A 4.00 3.00 B 7.00 6.00 C 3.00 2.00	CASE: CASE1 I: WINTER J: OVERTIME K: BOLTS -----TIMEUS----- -----E----- G IN_TOWN OUT_OF_TOWN ----- A 5.00 4.00 B 6.00 5.00 C 0.00 0.00	CASE: CASE1 I: WINTER J: OVERTIME K: WASHERS -----TIMEUS----- -----E----- G IN_TOWN OUT_OF_TOWN ----- A 5.00 5.00 B 6.00 5.00 C 0.00 0.00

Fig. 29. Production times table (TIMEUS).

```

BREAK
CASE 'EXPECTED':
FORECAST LENGTH 2 METHOD
WINTERS PERIODICITY 2 -
ALPHA 0.5 BETA 0.5 GAMMA 0.5 -

```

```

TIME i.year FCNAME fcst.demand
old.demand
BREAK
CASE 'PESSIMISTIC':
FORECAST LENGTH 2 METHOD
WINTERS PERIODICITY 2 -
ALPHA 0.9 BETA 0.9 GAMMA 0.9 -

```

```

TIME i.year FCNAME fcst.demand
old.demand
BREAK
DOEND
LIMIT k TO ALL
LIMIT o TO ALL

```

```

LIMIT i.year TO ALL
DOEND
"Use only the forecasted values
LIMIT i.year TO year '1997'
"Copy the values to the Demand table for the
LP.
demand(i summer) = fcst.demand(i.year <
summer '1997' >)
demand(i winter) = fcst.demand(i.year <
winter '1997' >)
LIMIT i.year TO ALL

```

C.3.2. Other personal express automation programs

C.3.2.1. Listing 2. Program for preparing the database in roll-up mode.

```

"Program for setting the appropriate dimensions
" and calling the forecast-average programs
" in ROLL-UP mode

```

<p>CASE: CASE1 I: SUMMER K: NUTS</p> <pre> -----TRANCST----- -----E----- O IN_TOWN OUT_OF_TOW N N ----- NORTH 1.00 0.80 SOUTH 0.80 1.00 EAST 1.10 0.70 WEST 0.90 0.90 ALLAREAS 0.95 0.85 </pre> <p>CASE: CASE1 I: SUMMER K: BOLTS</p> <pre> -----TRANCST----- -----E----- O IN_TOWN OUT_OF_TOW N N ----- NORTH 1.00 0.90 SOUTH 0.80 1.10 EAST 1.10 0.80 WEST 0.90 0.90 ALLAREAS 0.95 0.93 </pre> <p>CASE: CASE1 I: SUMMER K: WASHERS</p> <pre> -----TRANCST----- -----E----- O IN_TOWN OUT_OF_TOW N N ----- NORTH 1.00 0.80 SOUTH 0.80 1.00 EAST 1.10 0.70 WEST 0.90 0.90 ALLAREAS 0.95 0.85 </pre>	<p>CASE: CASE1 I: WINTER K: NUTS</p> <pre> -----TRANCST----- -----E----- O IN_TOWN OUT_OF_TOW N N ----- NORTH 1.10 0.90 SOUTH 0.90 1.10 EAST 1.20 0.80 WEST 1.00 1.00 ALLAREAS 1.05 0.95 </pre> <p>CASE: CASE1 I: WINTER K: BOLTS</p> <pre> -----TRANCST----- -----E----- O IN_TOWN OUT_OF_TOW N N ----- NORTH 1.10 0.90 SOUTH 0.90 1.10 EAST 1.20 0.80 WEST 1.00 1.00 ALLAREAS 1.05 0.95 </pre> <p>CASE: CASE1 I: WINTER K: WASHERS</p> <pre> -----TRANCST----- -----E----- O IN_TOWN OUT_OF_TOW N N ----- NORTH 1.10 0.90 SOUTH 0.90 1.10 EAST 1.20 0.80 WEST 1.00 1.00 ALLAREAS 1.05 0.95 </pre>
--	--

Fig. 30. Shipment costs table (TRANCST).

Summary (Roll-Up Model)	Detail (Drill-Down Model)								
<table border="1"> <tr> <th>CASE</th> <th>PROFIT</th> </tr> <tr> <td>EXPECTED</td> <td>2,107.36</td> </tr> </table>	CASE	PROFIT	EXPECTED	2,107.36	<table border="1"> <tr> <th>CASE</th> <th>PROFIT</th> </tr> <tr> <td>EXPECTED</td> <td>2,110.43</td> </tr> </table>	CASE	PROFIT	EXPECTED	2,110.43
CASE	PROFIT								
EXPECTED	2,107.36								
CASE	PROFIT								
EXPECTED	2,110.43								
<p>CASE: EXPECTED I: SUMMER J: NORMAL K: NUTS</p> <pre> -----PRODUCE----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 19.20 0.00 B 0.00 0.00 C 13.33 12.50 </pre>	<p>CASE: EXPECTED I: SUMMER J: NORMAL K: NUTS</p> <pre> -----PRODUCE----- -----E----- G IN_TOWN OUT_OF_TOW N ----- A 7.16 0.00 B 0.00 0.00 C 13.33 12.50 </pre>								
<p>CASE: EXPECTED I: SUMMER K: NUTS</p> <pre> ---SHIP--- ---O--- E ALLAREAS ----- IN_TOWN 47.53 OUT_OF_TOWN 25.83 </pre>	<p>CASE: EXPECTED I: SUMMER K: NUTS</p> <pre> -----SHIP----- -----O----- E NORTH SOUTH EAST WEST ----- IN_TOWN 0.00 26.57 0.00 8.92 OUT_OF_TOWN 10.76 0.00 21.92 0.00 </pre>								

Fig. 31. Selected optimisation results from roll-up and drill-down modes for the expected case.

Summary (Roll-Up Model)			Detail (Drill-Down Model)																																																																												
<table border="1"> <thead> <tr> <th>CASE</th> <th colspan="2">PROFIT</th> </tr> </thead> <tbody> <tr> <td>OPTIMISTIC</td> <td colspan="2">2,079.68</td> </tr> </tbody> </table>			CASE	PROFIT		OPTIMISTIC	2,079.68		<table border="1"> <thead> <tr> <th>CASE</th> <th colspan="3">PROFIT</th> </tr> </thead> <tbody> <tr> <td>OPTIMISTIC</td> <td colspan="3">2,210.80</td> </tr> </tbody> </table>				CASE	PROFIT			OPTIMISTIC	2,210.80																																																													
CASE	PROFIT																																																																														
OPTIMISTIC	2,079.68																																																																														
CASE	PROFIT																																																																														
OPTIMISTIC	2,210.80																																																																														
<table border="1"> <thead> <tr> <th colspan="3">CASE: OPTIMISTIC</th> </tr> <tr> <th>I:</th> <th colspan="2">SUMMER</th> </tr> <tr> <th>J:</th> <th colspan="2">NORMAL</th> </tr> <tr> <th>K:</th> <th colspan="2">NUTS</th> </tr> <tr> <th colspan="3">-----PRODUCE-----</th> </tr> <tr> <th colspan="3">-----E-----</th> </tr> <tr> <th>G</th> <th>IN_TOWN</th> <th>OUT_OF_TOW N</th> </tr> <tr> <th colspan="3">-----</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>25.00</td> <td>0.00</td> </tr> <tr> <td>B</td> <td>0.00</td> <td>0.00</td> </tr> <tr> <td>C</td> <td>0.00</td> <td>12.50</td> </tr> </tbody> </table>			CASE: OPTIMISTIC			I:	SUMMER		J:	NORMAL		K:	NUTS		-----PRODUCE-----			-----E-----			G	IN_TOWN	OUT_OF_TOW N	-----			A	25.00	0.00	B	0.00	0.00	C	0.00	12.50	<table border="1"> <thead> <tr> <th colspan="4">CASE: OPTIMISTIC</th> </tr> <tr> <th>I:</th> <th colspan="3">SUMMER</th> </tr> <tr> <th>J:</th> <th colspan="3">NORMAL</th> </tr> <tr> <th>K:</th> <th colspan="3">NUTS</th> </tr> <tr> <th colspan="4">-----PRODUCE-----</th> </tr> <tr> <th colspan="4">-----E-----</th> </tr> <tr> <th>G</th> <th>IN_TOWN</th> <th>OUT_OF_TOW N</th> </tr> <tr> <th colspan="4">-----</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>25.00</td> <td>0.00</td> </tr> <tr> <td>B</td> <td>0.00</td> <td>0.00</td> </tr> <tr> <td>C</td> <td>13.33</td> <td>12.50</td> </tr> </tbody> </table>				CASE: OPTIMISTIC				I:	SUMMER			J:	NORMAL			K:	NUTS			-----PRODUCE-----				-----E-----				G	IN_TOWN	OUT_OF_TOW N	-----				A	25.00	0.00	B	0.00	0.00	C	13.33	12.50
CASE: OPTIMISTIC																																																																															
I:	SUMMER																																																																														
J:	NORMAL																																																																														
K:	NUTS																																																																														
-----PRODUCE-----																																																																															
-----E-----																																																																															
G	IN_TOWN	OUT_OF_TOW N																																																																													

A	25.00	0.00																																																																													
B	0.00	0.00																																																																													
C	0.00	12.50																																																																													
CASE: OPTIMISTIC																																																																															
I:	SUMMER																																																																														
J:	NORMAL																																																																														
K:	NUTS																																																																														
-----PRODUCE-----																																																																															
-----E-----																																																																															
G	IN_TOWN	OUT_OF_TOW N																																																																													

A	25.00	0.00																																																																													
B	0.00	0.00																																																																													
C	13.33	12.50																																																																													
<table border="1"> <thead> <tr> <th colspan="2">CASE: OPTIMISTIC</th> </tr> <tr> <th>I:</th> <th>SUMMER</th> </tr> <tr> <th>J:</th> <th>NORMAL</th> </tr> <tr> <th>K:</th> <th>NUTS</th> </tr> <tr> <th colspan="2">---SHIP---</th> </tr> <tr> <th colspan="2">---O---</th> </tr> <tr> <th>E</th> <th>ALLAREAS</th> </tr> <tr> <th colspan="2">-----</th> </tr> </thead> <tbody> <tr> <td>IN_TOWN</td> <td>44.86</td> </tr> <tr> <td>OUT_OF_TOWN</td> <td>25.83</td> </tr> </tbody> </table>			CASE: OPTIMISTIC		I:	SUMMER	J:	NORMAL	K:	NUTS	---SHIP---		---O---		E	ALLAREAS	-----		IN_TOWN	44.86	OUT_OF_TOWN	25.83	<table border="1"> <thead> <tr> <th colspan="5">CASE: OPTIMISTIC</th> </tr> <tr> <th>I:</th> <th colspan="4">SUMMER</th> </tr> <tr> <th>J:</th> <th colspan="4">NORMAL</th> </tr> <tr> <th>K:</th> <th colspan="4">NUTS</th> </tr> <tr> <th colspan="5">-----SHIP-----</th> </tr> <tr> <th colspan="5">---O---</th> </tr> <tr> <th>E</th> <th>NORTH</th> <th>SOUTH</th> <th>EAST</th> <th>WEST</th> </tr> <tr> <th colspan="5">-----</th> </tr> </thead> <tbody> <tr> <td>IN_TOWN</td> <td>0.05</td> <td>47.31</td> <td>0.00</td> <td>9.51</td> </tr> <tr> <td>OUT_OF_TOWN</td> <td>11.02</td> <td>0.00</td> <td>14.82</td> <td>0.00</td> </tr> </tbody> </table>				CASE: OPTIMISTIC					I:	SUMMER				J:	NORMAL				K:	NUTS				-----SHIP-----					---O---					E	NORTH	SOUTH	EAST	WEST	-----					IN_TOWN	0.05	47.31	0.00	9.51	OUT_OF_TOWN	11.02	0.00	14.82	0.00			
CASE: OPTIMISTIC																																																																															
I:	SUMMER																																																																														
J:	NORMAL																																																																														
K:	NUTS																																																																														
---SHIP---																																																																															
---O---																																																																															
E	ALLAREAS																																																																														

IN_TOWN	44.86																																																																														
OUT_OF_TOWN	25.83																																																																														
CASE: OPTIMISTIC																																																																															
I:	SUMMER																																																																														
J:	NORMAL																																																																														
K:	NUTS																																																																														
-----SHIP-----																																																																															
---O---																																																																															
E	NORTH	SOUTH	EAST	WEST																																																																											

IN_TOWN	0.05	47.31	0.00	9.51																																																																											
OUT_OF_TOWN	11.02	0.00	14.82	0.00																																																																											

Fig. 32. Selected optimisation results from roll-up and drill-down modes for the optimistic case.

"Calculate the sum of the past demands
ROLLUP old.demand OVER o USING o.total

"then use only the ALLAREAS value
LIMIT o TO allareas
"do the forecast
demand.program

"average the transportation costs
areas.program

"undo any 'limit' commands from the programs
LIMIT o TO allareas

C.3.2.2. Listing 3. Program for preparing the database in drill-down mode.

"Program for setting the status of the appropriate
"dimensions, and calling the forecasting program
in
"DRILL-DOWN Mode

"Use the areas without the total
LIMIT o to first 4

"Do the forecast
demand.program

"Calculate the totals for each area, so user
" can see total if needed
ROLLUP fcst.demand OVER o USING o.total

"undo any 'limit' commands for areas
LIMIT o TO FIRST 4

C.3.2.3. Listing 4. Program for adding another (season, year) value to the conjoint dimension.

"Program for adding (new) dimension values
" for the conjoint dimension i.year

```
OKFORLIMIT = yes
FOR year i
DO
IF ISVALUE (i.year, < i year >)
THEN GOTO end
ELSE maintain i.year add < i year >
end:
DOEND
```

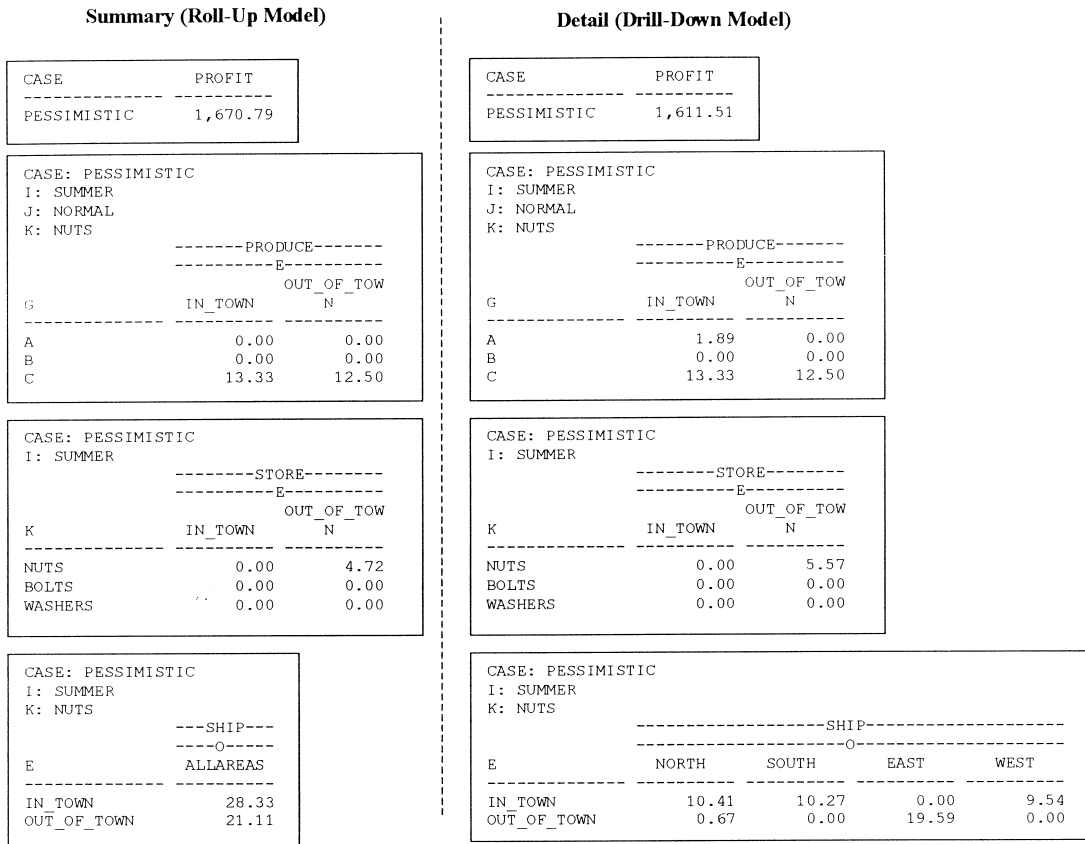


Fig. 33. Selected optimisation results from roll-up and drill-down modes for the pessimistic case.

C.3.2.4. Listing 5. Program for calculating the average of transportation costs when the Sales areas are in roll-up mode.

```
"Program for computing the average of
" the transportation costs

"Use the areas, but not their total
limit o to first 4
" for all other dimensions
for k i e
do
"calculate the average cost and store it in
allareas
trancst(o allareas) = average(trancst e)
doend
"Restore dimension status
limit o to all
```

C.3.3. LP-optimisation model: MPL code

```
TITLE nutsbolts
INDEX
i = INDEXFILE(Time.DAT)
j = INDEXFILE(Modes.DAT)
k = INDEXFILE(Products.DAT)
g = INDEXFILE(Machine.DAT)
o = INDEXFILE(Outlet.DAT)
e = INDEXFILE(Site.DAT)
DATA
TimeUs[i,j,k,g,e] := SPARSEFILE
("TimeUs.dat")
{Time Usage}

TimeAvl[i,j,g,e] := SPARSEFILE
("TimeAvl.dat")
{Time Available}
```

Price[i,k] := SPARSEFILE("Price.dat")
 {Selling Price}

demand[i,k,o] := SPARSEFILE
 ("demand.dat")
 {Demand at Outlet}

costor[k,e] := SPARSEFILE("costor.dat")
 {Storage Cost}

StorCap[k,e] := SPARSEFILE
 ("StorCap.dat")
 {Storage Capacity}

ProdCost[i,j,k,g,e] := SPARSEFILE
 ("ProdCost.dat")
 {Production Cost}

trancst[i,k,o,e] := SPARSEFILE
 ("trancst.dat")
 {Shipping Cost}

DECISION
 Produce[i,j,k,g,e] Where TimeUs $< > 0$ and
 ProdCost $< > 0$

EXPORT to Sparsefile("Produce.dat")
 {Production at site}

Store[i,k,e] EXPORT to Sparsefile("Store.dat")
 {Storage at site}

Ship[i,k,e,o] EXPORT to Sparsefile
 ("Ship.dat")
 {Shipment}

Shortage[i,k,o] EXPORT to Sparsefile("Shortage.dat")
 {Shortage variable}

MODEL

MAX Profit = Sum(i,k,o,e :Price[i,k]*
 Ship[i,k,e,o] – Sum(i,j,k,g,e :ProdCost*
 Produce[i,j,k,g,e] – Sum(i,k,o,e :trancst*
 Ship[i,k,e,o] – Sum(i,k,e :costor*
 Store[i,k,e] – Sum(i,k,o :15*
 Shortage[i,k,o])

SUBJECT TO

TimeCon[i,j,g,e] EXPORT Slack ShadowPrice
 RHSValue To SparseFile("TimeCon.dat"):
 {time Availability Constraint}

Sum(k :TimeUs[i,j,k,g,e] * Produce[i,j,k,g,e])
 \leq TimeAvl;
 StockB1[i,k,e] Where $i = 1$ EXPORT Slack
 ShadowPrice RHSValue To
 SparseFile("StockB1.dat"):
 {Stock balance Equation $tp = 1$ }

Sum($j,g:1$ * Produce[$i := 1,j,k,g,e$]) – 1 * Store
 [$i := 1,k,e$] – Sum($o:1$ * Ship[$i := 1,k,e,o$]) = 0;
 StockB2[i,k,e] Where $i = 2$ EXPORT Slack
 ShadowPrice RHSValue To
 SparseFile("StockB2.dat"):
 {Stock Balance Equation $tp = 2$ }

Sum($j,g:1$ * Produce[$i := 2,j,k,g$]) + 1 * Store
 [$i := 1,k$] – 1 * Store[$i := 2,k,e$] –
 Sum($o:1$ * Ship[$i := 2,k,e,o$]) = 0;
 demship[i,k,o] EXPORT Slack ShadowPrice
 RHSValue To SparseFile("demship.dat"):
 {Demand Shipping constraint}

Sum($e:1$ * Ship[i,k,e,o]) + 1 * Shortage[i,k,o]
 $> =$ demand;

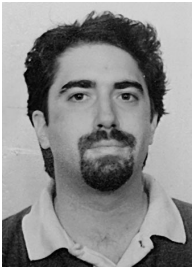
BOUNDS

Store[i,k,e] \leq StorCap;
 END

References

- [1] Arbor Software, Multidimensional Analysis: Converting Corporate Data Into Strategic Information, Arbor Software, 1993.
- [2] Arbor Software, Relational OLAP: Expectations and Reality, Arbor Software, 1995.
- [3] A. Berson, S.J. Smith, Data Warehousing, Data Mining, and OLAP, McGraw-Hill, 1997.
- [4] F.A. Buytendjik, OLAP: Playing for Keeps, <http://www.xs4all.nl/~fab/olapkeep.html>, July 1995.
- [5] E.F. Codd, S.B. Codd, C.T. Salley, Providing On-Line Analytical Processing to User-Analysts: An IT Mandate, E.F. Codd and Associates, 1993.
- [6] E.F. Codd, S.B. Codd, OLAP with TM/1, E.F. Codd and Associates, 1994.
- [7] M. Demarest, Building the Data Mart, DBMS, July 1994.
- [8] K.M. Decker, S. Focardi, Technology Overview: A Report

- on Data Mining, CSCS TR-95-02, Swiss Scientific Computing Center, May 29, 1995.
- [9] E.F.D. Ellison, G. Mitra, UIMP: user interface for mathematical programming, *ACM Transactions on Mathematical Software* 8 (3) (1982) .
- [10] S. Filbin, Successful decision support in the retail sector, in: *Proceedings of 'EIS AND OLAP'*, UNICOM Seminars, London, July 1995.
- [11] R. Finkelstein, Understanding the need for on-line analytical servers, Arbor Software, 1994.
- [12] A.M. Geoffrion, An introduction to structured modelling, *Management Science* 33 (5) (1987) .
- [13] H.J. Greenberg, F.H. Murphy, Views of mathematical programming models and their instances, *Decision Support Systems* 13 (1) (1995) .
- [14] C.W. Holsapple, A.B. Whinston, The environmental approach to decision support, in: G. Mitra (Ed.), *Mathematical Models for Decision Support*, NATO ASI Series, Vol. F48, Springer-Verlag, 1988.
- [15] M. Holsheimer, A.P.J.M. Siebes, Data mining: the search for knowledge in databases, CS-R9406 1994, Computer Science/Department of Algorithmics and Architecture, Centrum voor Wiskunde en Informatica (CWI), 1994.
- [16] W.H. Inmon, What is a Data Warehouse?, http://www.cait.wustl.edu/cait/papers/prism/vol1_no1/, 1995.
- [17] W.H. Inmon, J.D. Welch, K.L. Glassey, *Managing the Data Warehouse*, Wiley, 1997.
- [18] C. Jones, *Visualization and Optimization*, Kluwer Academic Publishers, 1997.
- [19] C. Jones, *Visualization and Optimization*, <http://www.chesapeake2.com/cvj/itorms.html> version of Ref. [17].
- [20] Kenan Technologies, *Multidimensional Database Technology and Data Warehousing*, Kenan Systems, 1995.
- [21] Kenan Technologies, *An Introduction to Multidimensional Database Technology*, Kenan Systems, 1995.
- [22] R. Kimball, *The Data Warehouse Toolkit*, Wiley, 1996.
- [23] N.-S. Koutsoukis, G. Mitra, S.A. Moody, Information systems in support of decision making, Technical Report TR/01/96, Brunel University, Department of Mathematics and Statistics, UK, January 1996.
- [24] N.-S. Koutsoukis, G. Mitra, C. Lucas, Adapting on-line analytical processing (OLAP) for decision modelling: the interaction of information and decision technologies, Technical Report TR/14/97, Brunel University, Department of Mathematics and Statistics, UK, Revised February 1998.
- [25] Maximal Software, *MPL Modelling System: Release 4.0*, Maximal Software 1996.
- [26] S. McLean, B. Scotney, The data mining report, Information Technology Report, UNICOM Seminars, 1996.
- [27] MicroStrategy, *The Case for Relational OLAP*, Microstrategy, 1995.
- [28] G. Mitra, C. Lucas, S.A. Moody, B. Kristjansson, Sets and indices in linear programming and their integration with relational data models, *Computational Optimization and Applications* 4 (1995) 263–283.
- [29] R.H. Möhring, R. Müller, F.J. Radermacher, Advanced DSS for scheduling: software engineering aspects and the role of eigenmodels, in: *Proceedings of the 27th Hawaii International Conference on System Sciences*, Wailea, HI, IEEE Computer Society, January 4–7, 1994.
- [30] H. Mousavi, G. Mitra, C. Lucas, Data and optimization modelling: a tool for elicitation and browsing (DOME), in: R.S. Barr, R.V. Helgasm, J.L. Kennington (Eds.), *Interfaces in Computer Science and Operations Research: Advances in Metaheuristics, Optimization and Stochastic Modelling Technologies*, Kluwer Academic Publishers, 1996, pp. 297–324.
- [31] J. O'Brien, *Management Information Systems: Managing Technology in the Networked Enterprise*, 3rd edn., IRWIN 1996.
- [32] OLAP Council, *OLAP Glossary*, <http://www.olapcouncil.org/research/glossaryly.htm>, OLAP Council 1995.
- [33] ORACLE, *Personal Express User's Guide*, ORACLE 1996.
- [34] ORACLE, *Express Language Reference*, Vols. I and II, ORACLE 1996.
- [35] K.H. Palmer, N.K. Boudwin, H.A. Patton, A.J. Rowland, J.D. Sammes, D.M. Smith, *A Model Management Framework for Mathematical Programming*, An EXXON Monograph, Wiley, 1984.
- [36] G. Piatetsky-Shapiro, W.J. Frawley (Eds.), *Knowledge Discovery in Databases*, AAAI Press/MIT Press, 1991.
- [37] Pilot Software, *An Introduction to OLAP: An Explanation of Multidimensional Terminology and Technology*, Pilot Software, 1995.
- [38] N. Pendse, R.C. Creeth, *The OLAP report*, Business Intelligence, 1995.
- [39] S. Raghunathan, A structured modelling methodology to design decision support systems, *Decision Support Systems* 17 (4) (1996) .
- [40] R.G. Ramirez, U.R. Kulkarni, K.A. Moser, Derived data for decision support systems, *Decision Support Systems* 17 (2) (1996) .
- [41] Red Brick, *The Data Warehouse: Competitive Advantage for the 90s*, Red Brick Systems, July 27, 1994.
- [42] R. Rock-Evans, *Middleware: tools and cross-platform integration*, in: *Proceedings of Data Mining and Data Warehouse '96*, UNICOM Seminars, London, November 1996.
- [43] D.F. Rogers, R.D. Plante, R.T. Wong, J.R. Evans, Aggregation and disaggregation techniques and methodology in optimisation, *Operations Research* 39 (4) (1991) .
- [44] T. Sach, I. Collins, J. Page, *The data warehouse report*, Information Technology Report, UNICOM Seminars, 1996.
- [45] V. Sauter, *Decision Support Systems: An Applied Managerial Approach*, Wiley, 1997.
- [46] S.L. Savage, *Fast QM: Fundamental Analytic Spreadsheet Tools for Quantitative Management*, McGraw-Hill, 1993.
- [47] J.F. Shapiro, *The decision database*, Working Paper, Sloan WP#3570-93-MSA, June 1993.
- [48] W.G. Sullivan, W. Claycombe, *Fundamentals of Forecasting*, Reston Publishing, 1977.
- [49] N.T. Thomopoulos, *Applied Forecasting Methods*, Prentice-Hall, 1980.
- [50] E. Thomsen, *OLAP Solutions: Building Multidimensional Information Systems*, Wiley, 1997.



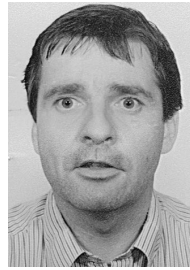
Dr. Nikitas-Spiros Koutsoukis has a first class honours BSc in accounting from the Technological Education Institute in Messologhi in Greece, and a Masters degree in Decision Modelling and Information Systems, from the Department of Mathematics and Statistics at Brunel University where he completed his Ph.D. Research. Dr. Koutsoukis lectures on the MSc course in Data and Information Modelling, and is mainly interested in Information Systems (IS) and models

for Decision Support (DS). In particular his research investigations cover the interaction between IS and DS tools. He is also investigating the application of Agent Technologies and Multi-Agent architectures applied to EIS/OLAP and enterprise information systems.



Professor Gautam Mitra has a first degree BEE from Jadavpur University, Calcutta, and MS and a PhD from London University. He is the head of the department of Mathematics at Brunel University and leads a research group in the area of Computational Optimisation and Modelling. Professor Mitra's research interests cover a wide field including Linear Programming, Integer Programming, solution algorithms and software systems for serial as well as

parallel computers. He is also interested in the issues of modelling and knowledge representation in general and in particular that of interaction between IS, modelling and solution technologies. Professor Mitra has written over 80 journal articles and one book and has edited five other books. He is a fellow of the IMA and BCS and is member of INFROMS, USA, ORSOC, GB and the MP Society International.



Dr. Cormac Lucas obtained his BSc and PhD at Brunel University. He is a lecturer in Operational Research and Statistics in the Department of Mathematics and Statistics at Brunel University and he is member of the Brunel Computational Optimisation and Modelling Research Group. His research interests cover the development of integrated software systems for modelling of Linear Programming and Integer Programming Application. He is currently involved on a number of industrial research projects that combine IS

with DS. He is also interested in modelling and solving of Stochastic Programming problems and their use in DS and IS.