Seagate Info

Introduction to On-line Analytical Processing (OLAP)

Overview

This document aims to help the new user understand the issues that need to be considered when starting to design an OLAP application. This is accomplished with the aid of a case study, in which the following topics are discussed:

- The Toys case study.
- Breaking down the problem.
- Identifying output, input and calculations.
- Common pitfalls to avoid.

Contents

Problem breakdown

An application will normally be driven by a business need. Someone in your company will have a business problem that they want to use OLAP tools to address, through the development of an application. This application must be designed to allow the target users to report on, model, and analyze data (which may currently be stored in a number of different systems) to gain meaningful information, highlight areas of key importance, and identify problems and opportunities.

The section below provides the background to the case study upon which the examples in this document are based.

Toys - Setting the Scene

"Toys" is a fictitious wholesale and distribution company, dealing with a range of toys and games brought in from manufacturers around the world. Retail outlets place regular orders with Toys, who delivers the required products throughout the country.

The company is organized into a number of semi-autonomous divisions, each of which is responsible for selling a range of products. There are currently four divisions – Collectibles, Soft Toys, Electronics and Computer Games. However, the company is keen to be responsive to the marketplace, and is likely to create new divisions, or merge existing ones, as time goes by.

A sensible starting point, when designing an application, is to try to break the problem down into a series of key components. From the small amount of background information available about Toys, you can begin to identify the essential components as being:

- **Products** – the company deals in a range of toys and games.
- **Manufacturers**, located throughout the world, provide toys and games to Toys.
- **Retailers**, possibly with a number of outlets each, place orders with the company.
- **Time** – orders are placed on a regular basis, so there must be some aspect of time in this analysis.
- **Divisions**, each of which is responsible for a range of products.

Identifying Output, Input and Calculations

Any new system will have a set of requirements it needs to fulfill. The extent and complexity of these will vary, but the steps that need to be taken to ensure they are met can be applied to any situation. The section below provides a high-level definition of requirements from Toys.

Requirements of the Toys Application

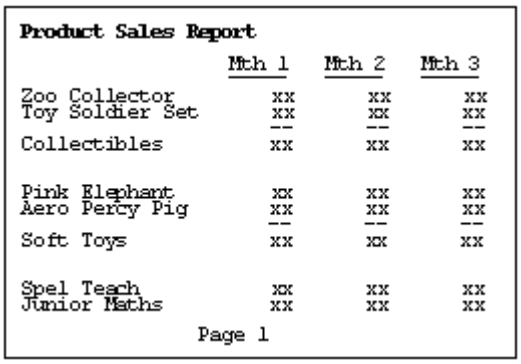
The Managing Director of Toys is concerned that, as the business grows, the company continues to manage profitability. He wants to use OLAP to help them to do so. Competition between the divisions is actively encouraged; therefore, one of the features of the new system should be its

ability to highlight which areas of the company are doing best (or worst!) according to certain criteria.

The factors against which divisions are to be measured include:

- **Sales.** This can be actual units sold, or the value of actual against forecast sales, on a monthly basis.
- **Margin**. Which divisions are contributing the most to company profits?
- **Costs.** Divisions incur direct costs of the products they sell, but also incur a share of the company overhead. Therefore, there may be a number of different cost measures that need to be examined.

The Finance department currently produces some paper-based reports of product sales by division, upon which the new system must be based and expand. Figure 1 shows an example of one of these reports.



Product Sales Report

	Mth 1	Mth 2	Mth 3
Zoo Collector	xx	xx	xx
Toy Soldier Set	xx	xx	xx
Collectibles	xx	xx	xx
Pink Elephant	xx	xx	xx
Aero Percy Pig	xx	xx	xx
Soft Toys	xx	xx	xx
Spel Teach	xx	xx	xx
Junior Maths	xx	xx	xx

Page 1

Figure 1 *Sample Product Sales Report*

The Managing Director is particularly keen to encourage more accurate forecasting and planning within the company. Divisional managers are required to prepare a five-year plan of their division's Profit and Loss figures, taking information from what has happened this year and projecting it forward. Any such plan will also need to forecast cash flow within the division.

Before starting to identify the output required by the new system, the information on Toys' requirements can be used to expand on your list of key components:

- Sales, Cost and Margin information must be made available. Direct costs will be at product level, while overhead costs are incurred by a division. Mention is also made of Profit and Cash Flow.
- Variables. Sales are monitored in units and value.
- Forecasts for product sales are prepared and used to compare against Actual.
- Time. The time frame for reporting sales is on a Monthly basis. Yet, there also appears to be scope for Years, to produce the forward plan.

It may help to begin a pictorial representation of the key components, as you identify them. Using a plain sheet of paper, work around the outside and draw a box around each set of key components you have identified, grouping similar ones together.

For example:

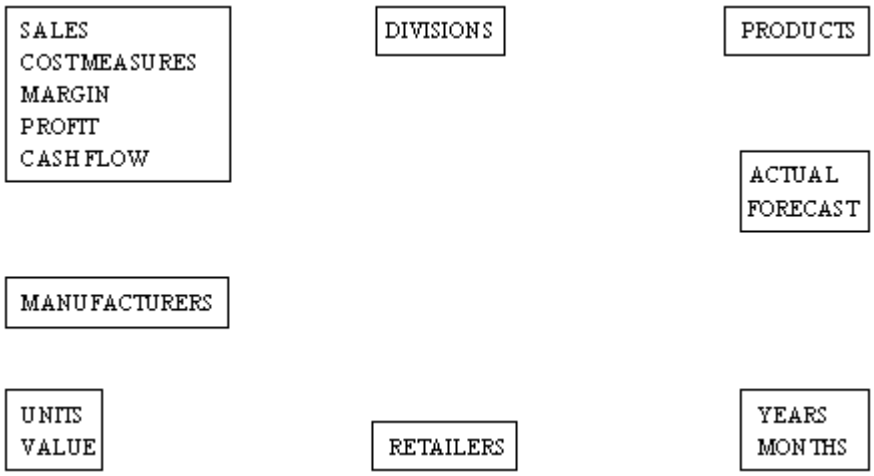


Figure 2

As more understanding of the problem and its required solution is gained through further analysis, you can work with a diagram such as the one above, to reflect changes.

Output

With a basic understanding of the key components of the business situation, you can use the output required as a starting point to define the scope of the design, and to provide further insight into analyzing the problem.

Existing reports that the company produces can be a good place to start. There are likely to be a number reports that people within the company find invaluable. The new application should aim to provide at least the same information as these, but with possible improvements in terms of layout, ease of use, etc.

Care should be taken not to reproduce redundant information. Just because a particular report has been available historically does not mean it contains important information. You should spend time understanding what the information is used for.

Product Sales Report

The existing Product Sales report shows, for each division, the number of products sold in each month of the year. This allows the sales of products within a division to be compared relatively easily, and also permits month-on-month comparisons of sales of a particular product. However, it does not make divisional comparisons easy, and the Toys requirements suggest that this should be an aspect of the system to be designed.

One output first required of the application may therefore be a report such as the following:

Unit Sales Report - Actual

	Mth 1	Mth 2	--	Mth 12
Collectibles	xx	xx		xx
Soft Toys	xx	xx		xx
Electronics	xx	xx		xx
Computer Games	xx	xx		xx
Holistic Toys	xx	xx		xx

Figure 3

This report shows data at a higher level than the one currently produced by Toys. It allows you to view sales data aggregated at division level, making comparisons between divisions a lot easier. With a report like this, you could then use the power of OLAP to drill down on a particular division, to view the sales of its products. For example:

Unit Sales Report - Actual

	Mth 1	Mth 2	--	Mth 12
Wing Commander	xx	xx		xx
Princess of Doom	xx	xx		xx
Holomania	xx	xx		xx
Computer Games	xx	xx		xx

Figure 4

A report in this format might also be useful to be viewed graphically:

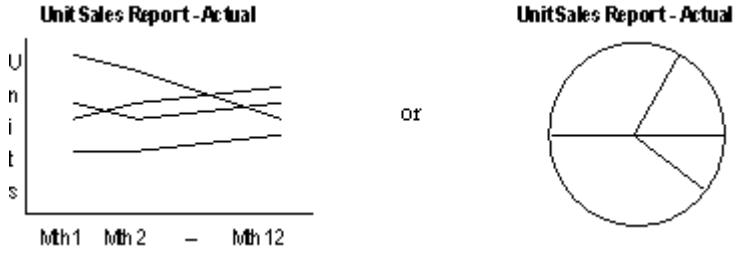


Figure 5

Looking in more detail at this first set of output should have provided you with additional information for your components diagram. Specifically, it now appears that divisions and products may be closely related. It is possible for you to drill down from a division to the products that it sells, because the sales figures for a particular division can be derived by summing the sales of all its products. It may be worth grouping products and divisions together on the diagram:

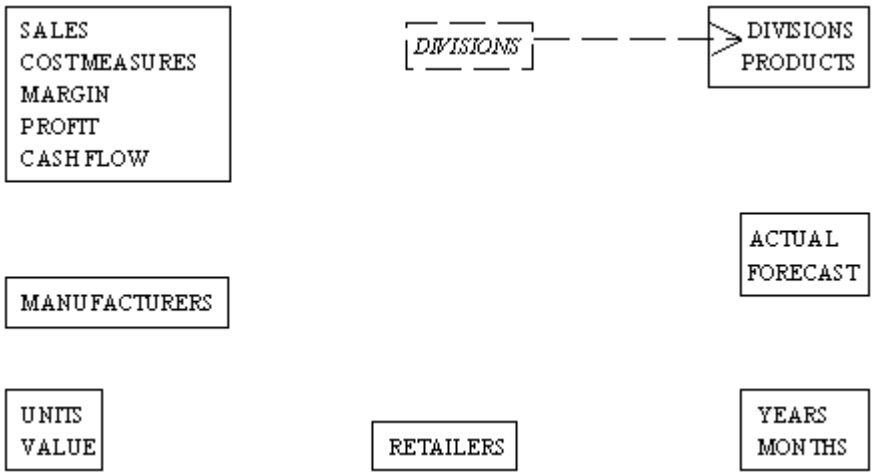


Figure 6

Moving on from existing reports, you can now look at those required output, which the system will be providing for the first time. In many cases, the output will be providing information, which is not currently available, and so you do not have any existing reports to use as a starting point. In cases such as this, you can draw up an initial design for a report, but it is important that feedback is obtained from the potential users as soon as possible. This process of refining the design of an application, by repeatedly building samples of the final application for comment by users, is known as 'iterative prototyping'.

OLAP provides the ideal environment for adopting an iterative prototyping approach to application development, and by creating screens, which the target audience can review and comment on, you can obtain useful additional information as part of the analysis process.

Divisional Comparison Report

The Managing Director of Toys wants to be able to compare divisional performance on factors such as Sales, Cost of Sales and Margin. A first draft of this report may look something like this:

Divisional Performance Report

	Sales	Cost of Sales	Margin
Collectibles	xx	xx	xx
Soft Toys	xx	xx	xx
Electronics	xx	xx	xx
Computer Games	xx	xx	xx

Figure 7

To facilitate ease of use, you could choose to sort the divisions based on, for example, the value of their sales; this would allow easy identification of those divisions that were doing best or worst.

Alternatively, it may be preferable to color-code the sales figures, to make problem areas instantly visible. This could be done based on the difference between each division's actual sales compared with their planned sales, showing those divisions that exceeded their target in green, and those that had not met their target in red.

These features of the report can be decided at a later stage, during the evolution of the prototype application.

Profit and Loss Planning

You are told that there will need to be some way of preparing a five-year plan of each division's Profit and Loss. This suggests the need for an interactive report which will show the detailed items that make up the division's profit and loss, on a five-year time-scale, such as:

Profit and Loss Report - Electronics Division

	This Year	Year + 1	Year + 2	Year + 3	Year + 4
Sales	xx	xx	xx	xx	xx
Cost of Sales	xx	xx	xx	xx	xx
Gross Profit	xx	xx	xx	xx	xx
Tax	xx	xx	xx	xx	xx
Profit after Tax	xx	xx	xx	xx	xx

Figure 8

Cash Flow Planning

Another requirement for the system is for it to assist with Cash Flow planning over a five-year period. This report is likely to be closely related to the Profit and Loss report, sharing a number of common data items.

The precise way that the company manages and reports on cash flow will determine the final design of this report, but an initial design of the Cash Flow report may be as follows:

Cash Flow Forecast - Soft Toys Division

	This Year	Year + 1	Year + 2	Year + 3	Year + 4
Opening Balance	xx	xx	xx	xx	xx
Receipts	xx	xx	xx	xx	xx
Payments	xx	xx	xx	xx	xx
Financing	xx	xx	xx	xx	xx
Tax	xx	xx	xx	xx	xx
Closing Balance	xx	xx	xx	xx	xx

Figure 9

This report could be used as a means of prompting the appropriate users to explain their requirements in more detail.

With the list of the output required from the system becoming clearer, you should return to your diagram of key components to try to define the scope of your application design.

Drawing a dotted line around those 'entities' that are included in the scope, you can see that this application does not appear to involve manufacturers or retailers in any way. These components should not yet be removed from the diagram, as further analysis of the business may suggest that they do belong within the application scope.

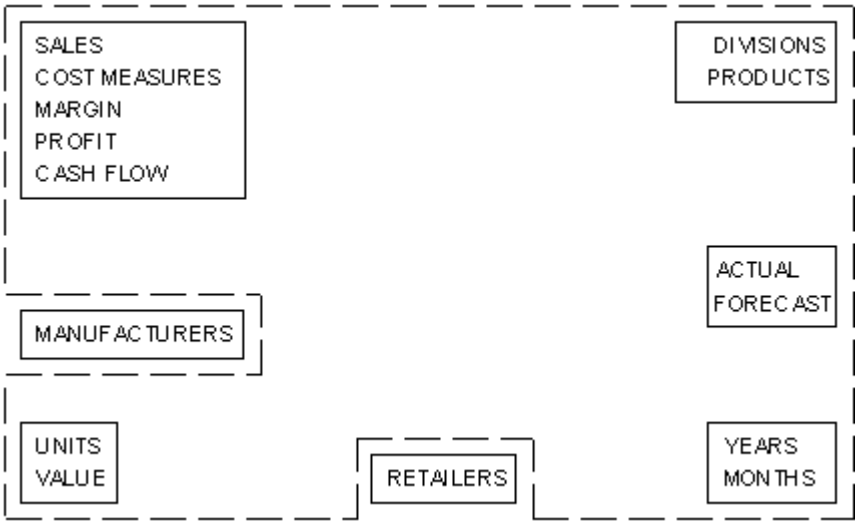


Figure 10

Input

Having defined the output required from the application, you can now have a look at the input that will be available. What you need to do, at this stage, is to identify those data sources that can be used in the application to support the specified requirements. You will effectively be mapping the input available to the output required.

Most companies will use a range of computer tools to manage their business. As part of the initial phase of the OLAP application development, data must therefore be gathered from each of these different sources to construct the OLAP models.

Following a series of fact-finding interviews with key Toys' personnel, you obtain details of data sources that may be used in the application.

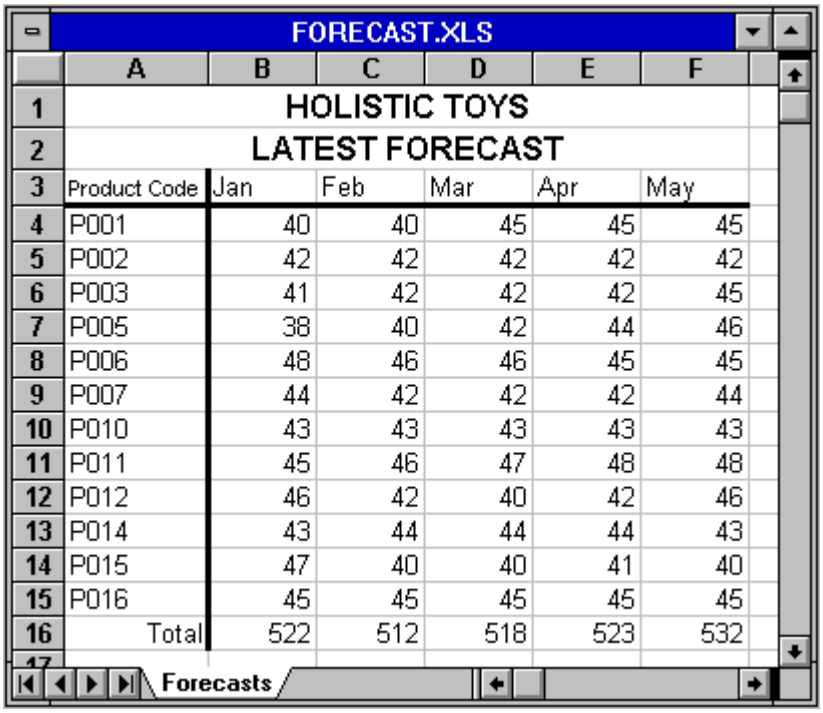
Spreadsheets

	A	B	C	D	E	F	G	H	J
1	Holistic Toys			Product List					
2									
3	Division			Product Code		Product Description		Cost Price	Selling Price
4									
5	Collectibles			P001		Zoo Collector		5	6.99
6	Collectibles			P002		Toy Soldier Set		6	10
7									
8	Soft Toys			P003		Pink Elephant		3	3.5
9	Soft Toys			P005		Aero Percy Pig		2	3
10									
11	Electronics			P006		Spel Teach		9	9.99
12	Electronics			P007		Junior Maths		4	8
13	Electronics			P010		Junior Counter		4	5.5
14	Electronics			P011		Compu Prof		10	11.99
15	Electronics			P012		Micro Count		14	15
16									
17	Computer Games			P014		Wing Commander		12	12.5
18	Computer Games			P015		Princess of Doom		8	8.95
19	Computer Games			P016		Holomania		18	20
20									

Figure 11

Data on the products that the company deals in is currently maintained in Microsoft Excel spreadsheet. Each new product is allocated a code identifier, and its cost and selling price recorded against it; which division of the company is responsible for the product is also recorded here.

Each division of the company produces a plan of how many units of each product they anticipate selling in each month of the financial year. These forecasts are pulled together into a spreadsheet in Excel by the Finance department, and maintained on a monthly basis, as shown on the next page.



FORECAST.XLS

	A	B	C	D	E	F
1	HOLISTIC TOYS					
2	LATEST FORECAST					
3	Product Code	Jan	Feb	Mar	Apr	May
4	P001	40	40	45	45	45
5	P002	42	42	42	42	42
6	P003	41	42	42	42	45
7	P005	38	40	42	44	46
8	P006	48	46	46	45	45
9	P007	44	42	42	42	44
10	P010	43	43	43	43	43
11	P011	45	46	47	48	48
12	P012	46	42	40	42	46
13	P014	43	44	44	44	43
14	P015	47	40	40	41	40
15	P016	45	45	45	45	45
16	Total	522	512	518	523	532

Forecasts

Figure 12

General Ledger

The company also has a general ledger system, which is based on a relational database. This system stores warehousing and distribution information, as well as maintaining records of product sales and divisional costs on a monthly basis.

Table descriptions for those database tables that will provide input data to the OLAP application are available.

The database table *current_sales* is made up of 13 columns. The first column contains a product code, and the remaining 12 columns represent unit sales of that product code in each month of the current year.

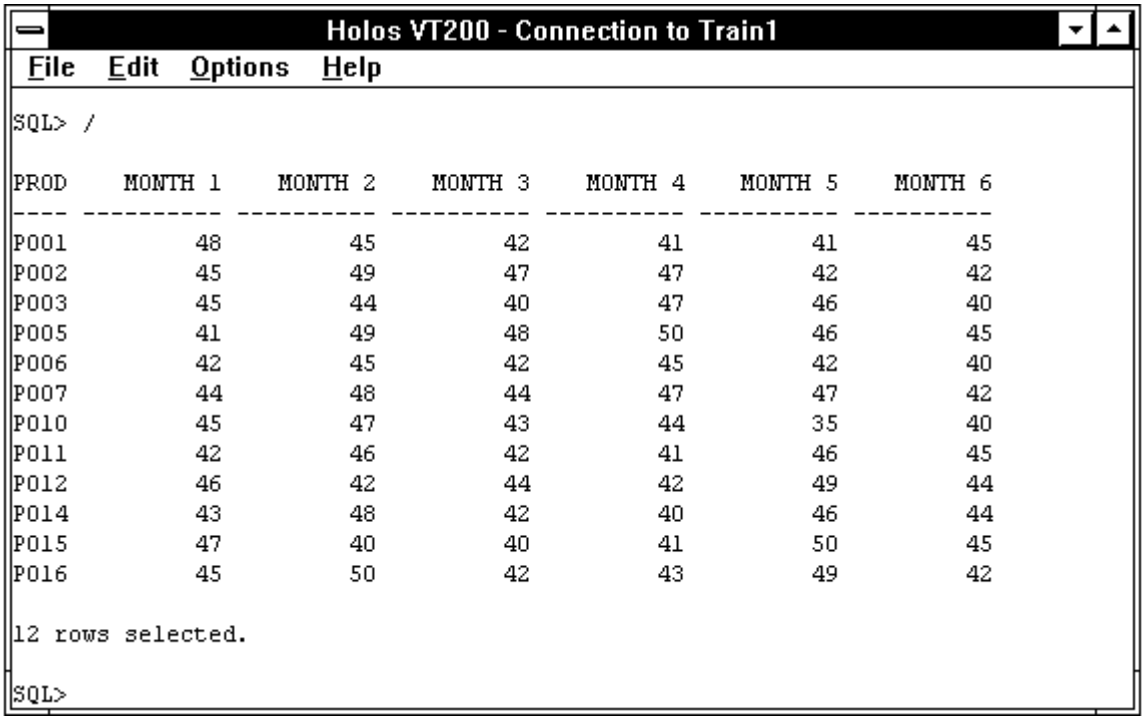


Figure 13

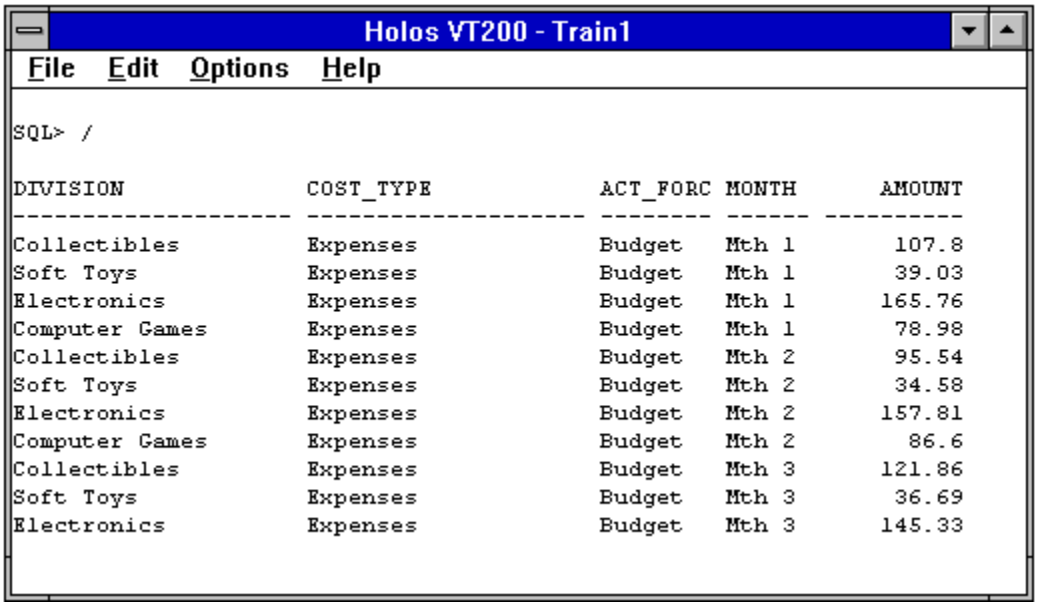


Figure 14 Table *division_cost* has 5 columns*: division, cost_type, month, amount* and *act_forc.*

Manual Input

The preparation of five-year plans will rely on additional information, which is not currently held in a convenient computerized form, being input to the application. This data is mainly to do with the cash flow analysis. Examples include:

- Purchases of fixed assets during the year.
- Interest received or paid.
- Changes in working capital.

Existing data sources are an invaluable source of information when designing an application, as they provide an insight into the logical representation of the data, and suggest ways in which the data could be stored within OLAP.

However, before going on to begin to define a logical design of the data, it is worth using these input as a cross-check against the key components identified earlier, to ensure that you have covered all the key items.

You are told that each product has a cost price and a selling price, which are recorded in a spreadsheet; these should be added to your diagram of the key components.

You can see, from the same spreadsheet, that each product has a unique code identifier as well as a description; this is very often the case, and can be dealt with easily by OLAP.

The spreadsheet also confirms the grouping of divisions and products together. Looking at this grouping in more detail, you can see that each product is only sold by one division. For example, *Zoo Collector* falls under the control of the *Collectibles* division and is not covered by any of the other divisions. From the division's point of view, each one can be responsible for a number of different products; for example, the *Collectibles* division covers the *Zoo Collector* and *Toy Soldier Set* products.

This kind of relationship or grouping is known as a 'one-to-many' relationship, and is often represented diagrammatically as shown below:

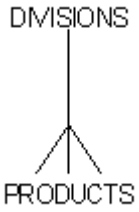


Figure 15

The single line at the divisions end represents the 'one', and the fork at the products end indicates 'many'. This line may be read in either direction, in order to understand the relationship that is being illustrated. Starting at divisions and working down the line to products, you can tell that one division has many products. Starting at products and working up the line to division, you can deduce that each product belongs to one division.

Quite easily, it could have been the case that this grouping of products and divisions was not so clearly defined. If a particular product could be sold by any number of divisions (as would be the

case if the divisions were regionally organized), then the relationship would have been 'many-to-many' and would be represented as follows:

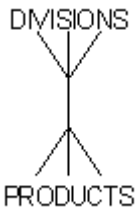


Figure 16

If this had been the case, then you would not group divisions and products together on your components diagram, but would have to change it back to its original form, where divisions and products were identified as separate components. This is because in this case it would not be possible to directly relate the sales of any particular product to one division, as products would not be unique to a division.

Looking at this another way, if you built up a grid which indicated which divisions could sell which products, then the situation at Toys, with its one-to-many relationship between divisions and products, might look something like:

	PRODUCT 1	PRODUCT 2	PRODUCT 3	PRODUCT 4
DIVISION 1	+			
DIVISION 2		+		
DIVISION 3			+	+

Figure 17

If the relationship had been many-to-many, the same grid would look more like the following:

	PRODUCT 1	PRODUCT 2	PRODUCT 3	PRODUCT 4
DIVISION 1	+		+	
DIVISION 2	+	+	+	
DIVISION 3			+	+

Figure 18

In this situation, if you knew Product 1's sales figures, you would need additional information before you could tell which division was responsible for those sales.

In the same way that you have defined a grouping of products to divisions, you can also group divisions into the company as a whole, and months into a year, as is shown in the diagram below.

You have been told about a number of manual input to the system that will be used in the preparation of five-year plans. Without needing to know specifically what they are at this stage, you should add these to your diagram of the key components.

In order to facilitate these manual input, you will also need to define an additional requirement of the system, namely, an input screen to allow divisional managers and their staff to add this information to the application. This should be added to the list of requirements you created earlier.

All of this information should be used to update your diagram:

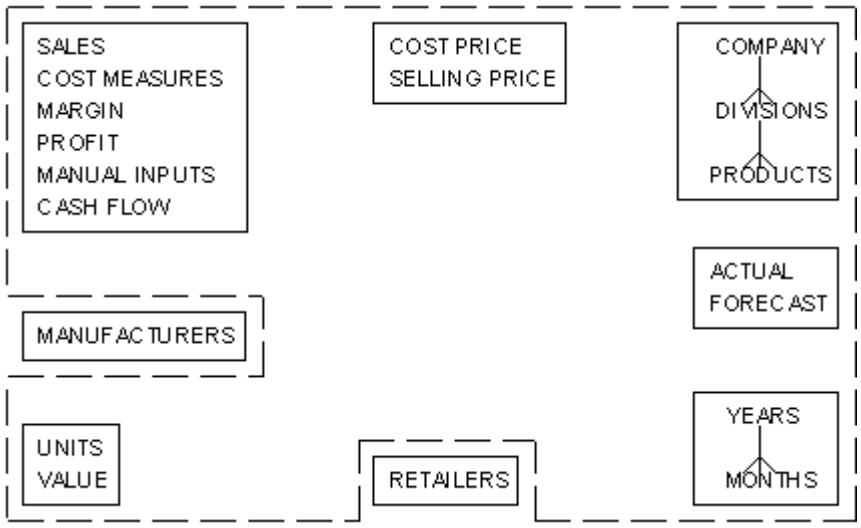


Figure 19

Having confirmed that your diagram now contains all the important entities or components, you can begin to link some of them to represent the data that will be contained in your application. All of the components you have drawn around the edge of your diagram are essentially different ways of looking at the Toys data. This next step will involve deciding how you want to see that data, and coming up with a logical design that will satisfy your requirements.

Logical Design

In this section, you will develop a logical design for the application. This involves understanding the natural shape of the data, which will be held in the system, and producing an abstract model of how this would be stored in an ideal world.

In a later step, you will consider the practical implementation of this in OLAP, in the form of a physical design. This will take into account any constraints, which may apply, to the system (e.g. required response times, memory availability etc.)

Because OLAP stores data in multi-dimensional objects, the logical design will involve identifying a set of these which will allow the storage of the application's data. However, at this stage, we will not be concerned about how the data will be held physically within your application. The OLAP environment provides a number of different ways for you to manage your data; each of these will be best suited to a particular scenario, and any number of these different methods can be combined within a single application. What you will be concentrating on here are the logical, or conceptual, definitions of the data.

In OLAP, the multi-dimensional data-structures are known simply as **structures** and the axes of these are known as **dimensions**. The items you have identified on the diagram so far will become the dimensions of your application. These dimensions will allow different views of the data to be

defined. Each of the structures you identify will be built up from one or more of these dimensions.

Price Data

As a first step, you can identify the need for a data structure to hold the current prices of each product; this will be largely similar to the spreadsheet currently maintained by the company. Each product has a value stored for cost price and selling price, so this data structure will need two dimensions – products and price categories – and will store price information.

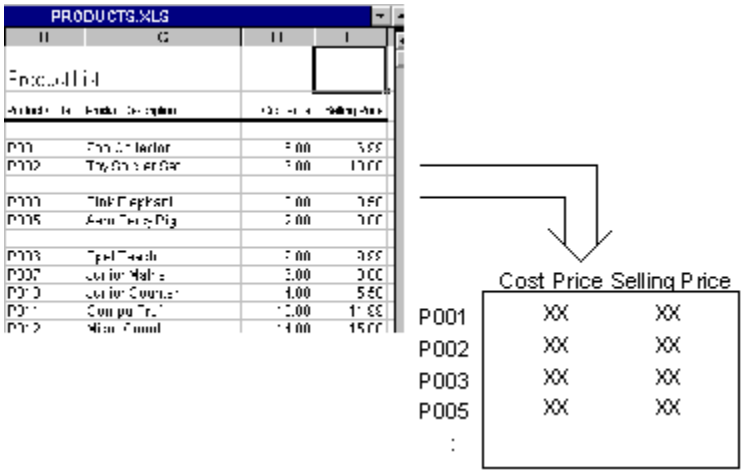


Figure 20

You can represent each logical data structure on your diagram by drawing a cube in the center of the page. You should draw lines from each cube to indicate the dimensions, which will be used in the structure the cube represents.

The cube marked 'A' in the diagram below represents the price data structure.

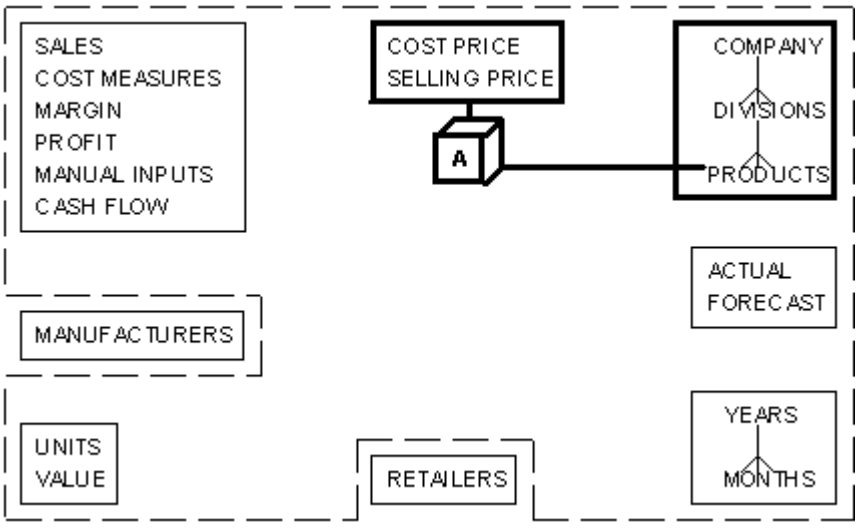


Figure 21

The diagram shows A as a two-dimensional data structure, because there are two lines connecting the cube with its component parts. Price categories will only relate to individual products, and not to divisions or the company as a whole. That is why one of the lines is drawn directly into products, and not to the box grouping products, divisions and the company.

Sales and Cost Data

You can now go on to define a second data structure to hold unit sales of each product. The *current_sales* database table from the Toys ledger system suggests that product sales are monitored on a monthly basis, so you can easily identify two axes for the data: products and months.

However, looking at the spreadsheet that the Finance department produces to collate forecast sales of each product, you find that this set of data also has the same 'dimensionality'. It would probably be worth expanding this data structure to include a third axis, or dimension, to store both actual and forecast figures.

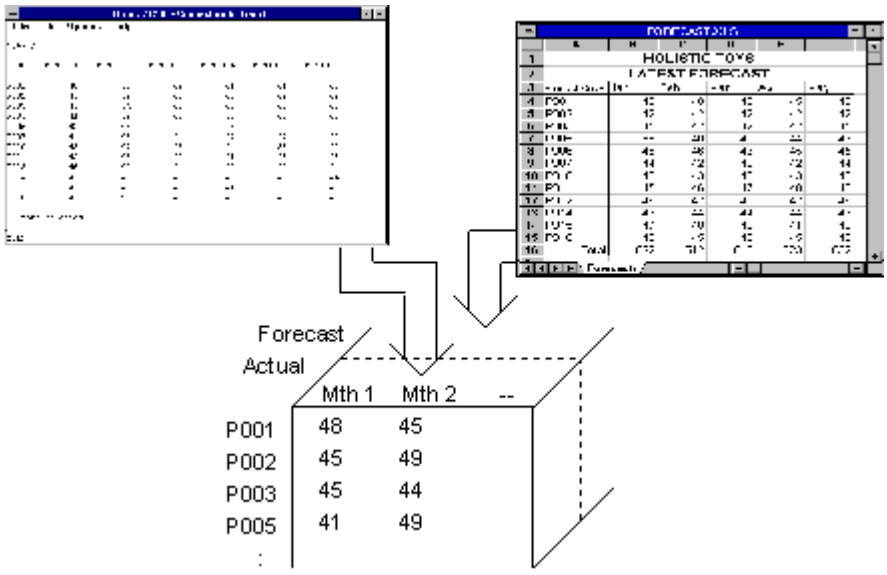


Figure 22

The cube marked 'B' in the diagram below would therefore represent a data structure containing actual and forecast unit sales for each product, by month.

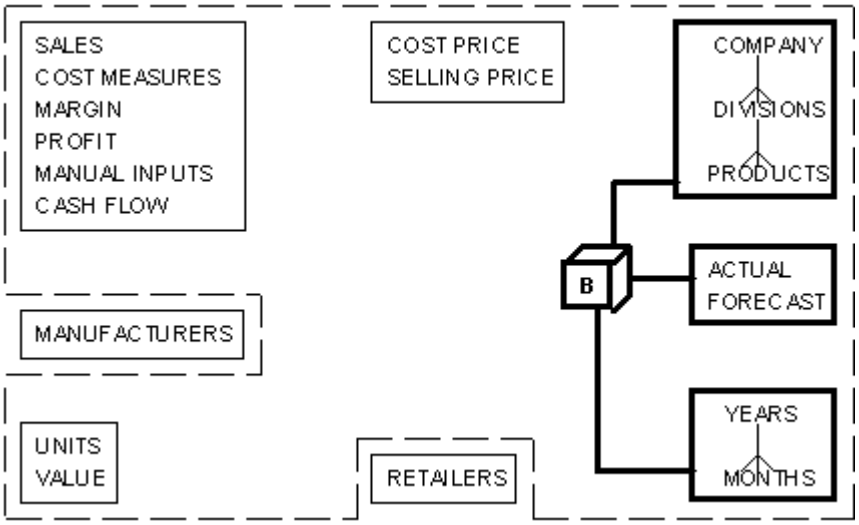


Figure 23

Note that all lines protruding from cube B are drawn to the box groupings, rather than directly to any one component. This is because unit sales can be viewed meaningfully at division and company level, and on an annual, as well as a monthly, basis.

It is not uncommon to find that a number of items of data appear to have the same dimensionality, as did actual and forecast unit sales. In these cases, it is possible to either keep these items as separate logical structures, or to add another axis to the logical structure already defined, as in the example just described. The most sensible design will usually fall out of the output requirements of the system. In the case of Toys, actual and forecast sales of a product are very similar data items and will need to be compared to each other, so you should keep them in the same logical structure.

If you had decided to keep actual and forecast unit sales as conceptually separate structures, the three-dimensional cube B in the last diagram would be replaced with two cubes, each representing a two-dimensional data structure. The first of these would hold the actual number of units of each product sold in each month; and the second would hold the forecast unit sales, by product and month.

Continuing with your analysis of the data, you find that product sales can also be represented in terms of their value. Should you now expand the last data structure, which will hold unit sales, to include a fourth axis, or dimension, to store both Units and Value figures? This option is illustrated in the diagram below.

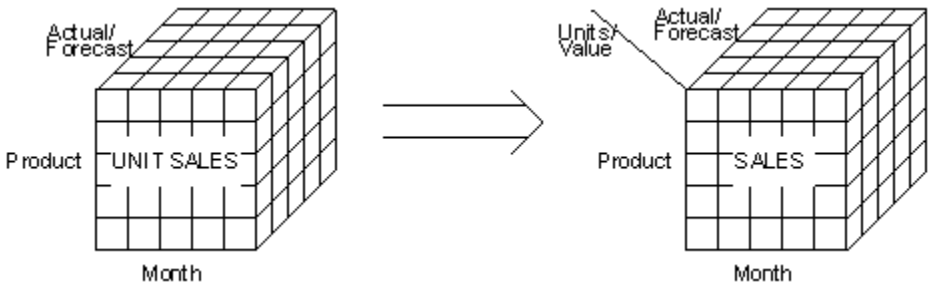


Figure 24

(Notice that although OLAP can support unlimited dimensions, it is difficult to represent any more than three meaningfully on paper. The fourth, and subsequent, dimensions in a structure tend to be drawn as lines protruding from the cube.)

Although this seems to be a natural thing to do there is a problem lurking round the corner. From the main structural diagram you can see that the 'sales' also occurs in the grouping at the top left of the diagram along with other Profit and Loss items such as cost measures and margin. As these items will also be held for actual\forecast and at divisional level, it is natural for these to be introduced as a new dimension in this structure...

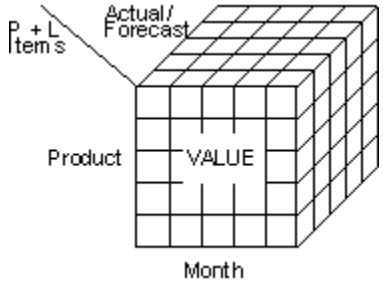


Figure 25

…or in terms of the structural diagram:

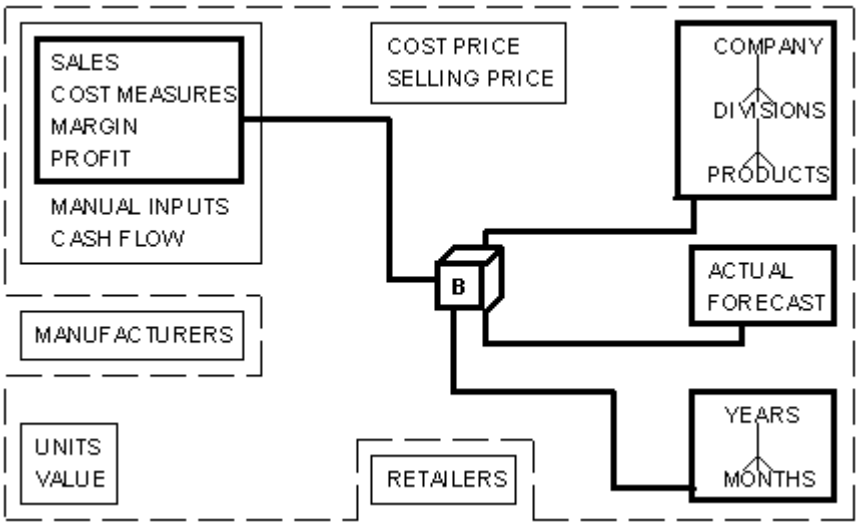


Figure 26

Unfortunately, while these new Profit and Loss items can represent 'value' type information, they cannot naturally represent 'units' type information and therefore a units\value dimension is not suitable for this structure.

However, the unit sales information will be held by actual\forecast, years\month and division\product and therefore has a very similar shape to the values structure (B):

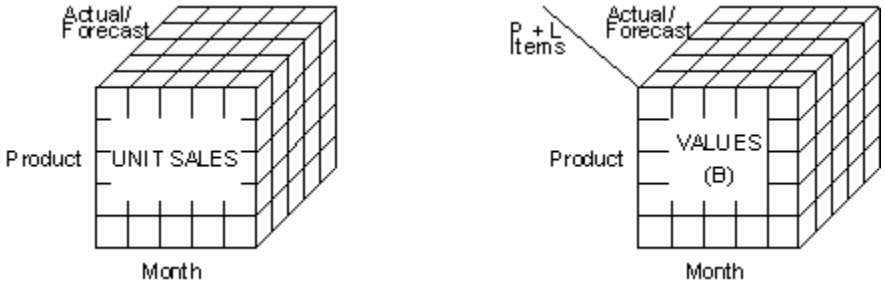


Figure 27

In fact, the only difference between the two is that the unit sales structure does not have the Profit and Loss dimension. In cases such as this, the smaller structure is in fact a multi-dimensional 'slice' of the larger structure and can be accommodated in the larger structure by expanding the non-shared dimension.

In this case, you can add a *unit sales* field to the Profit and Loss dimension to accommodate the unit sales figures:

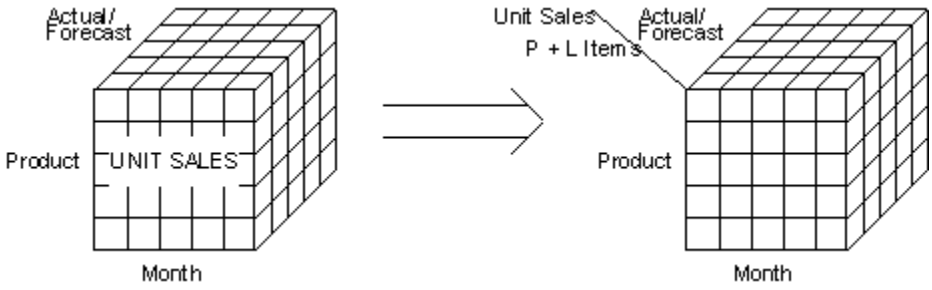


Figure 28

This change will require some changes to your structural diagram. The components box containing units and value should now be removed, and units must be added to the grouping of Profit and Loss line items.

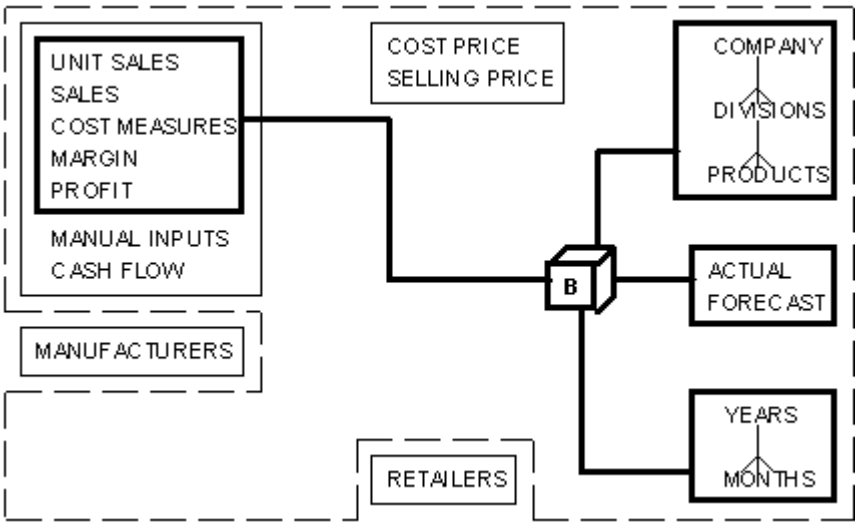


Figure 29

Thus the final structure, 'B' in the diagram, represents a four-dimensional structure holding monthly actual and forecast figures for toy sales, the value of those sales, and a series of other Profit and Loss items.

Five-Year Planning Data

A final logical structure will be required to support the preparation of five-year profit and loss and cash flow plans at divisional level. The full set of profit and loss line items will form one of the axes of this structure, and divisions a second. In addition to this, you will need to include a third axis, containing the five years that make up the planning period.

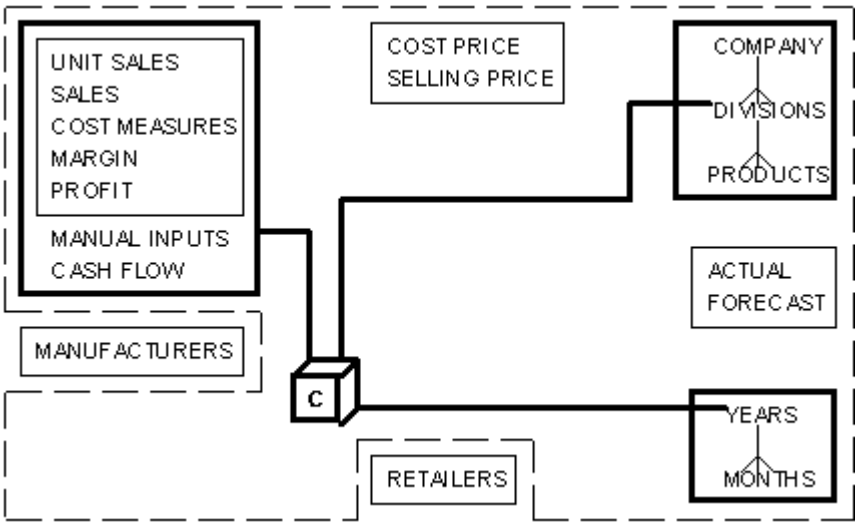


Figure 30

The diagram shows C as a three-dimensional data structure. The planning process is performed at divisional level, and not at base product level; because of this, one of the lines protruding from C is drawn directly into Divisions. One of the other lines is drawn directly into Years because the planning process will be performed on an annual time-scale.

You have now completed your diagrammatic representation of the important entities or components, and linked them to represent the data that will be contained in your application. Pulling all of the logical data representations together into one diagram can result in a diagram that appears, at first, to be rather complicated. However, by now you should be confident with each of its individual components, and should be able to understand how they all fit together.

The resulting diagram may look something like the one below:

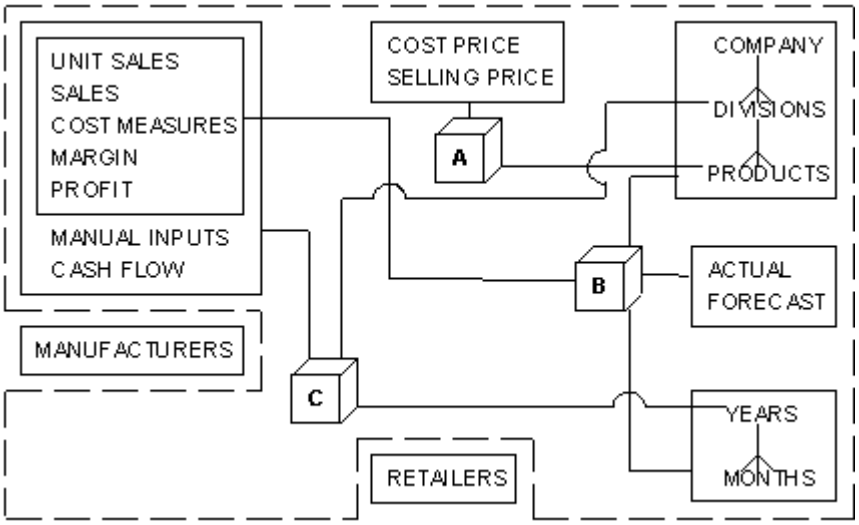


Figure 31

…where the structures are:

A - product price data

B - profit and loss and unit sales data

C - five year planning data

Now that you have a definitive list of the input to the system, along with the information gathered earlier on the output required from it, you can return to your diagram of key components to firmly define the scope of your application design.

You can confirm that all of the components within the dotted line should remain within the application, and manufacturers and retailers are not contained within its scope. These can therefore be removed from any subsequent versions of the diagram that you produce.

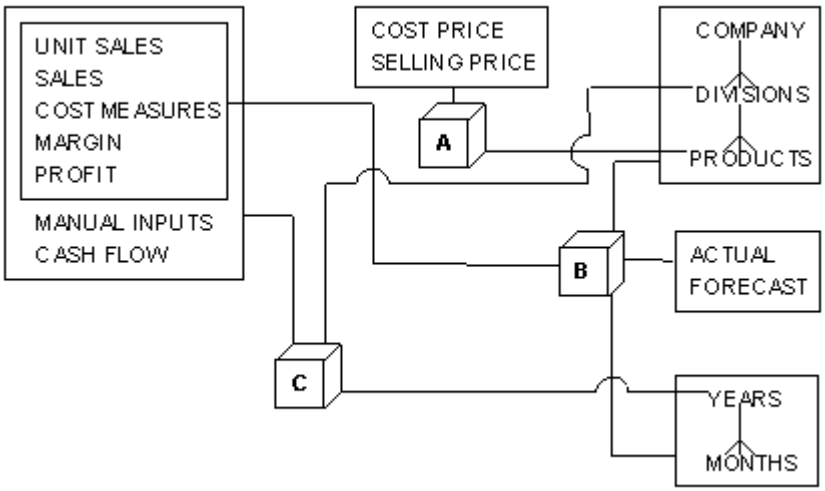


Figure 32

Calculations

Once you understand the output required from the application and have successfully mapped the input that are available to the output required, you can start to define any calculations that will need to be performed.

Taking each of the logical data structures that you have defined, in turn, you should be able to identify those data items that will need to be calculated by the application.

Product Price

The two-dimensional structure of product prices, marked as A on your diagram, will be initially populated in its entirety from one of the company's spreadsheets illustrated earlier. No calculations or manipulation of data will be required at this point.

Toys

The four-dimensional structure of toy unit sales, and profit and loss items for the current year, marked as B on your diagram, will be populated from three input data sources:

- the *current_sales* database table from the Toys ledger system will be used as a source for actual sales data
- the spreadsheet that the Finance department produces to collate forecast sales of each product will be used as a source for forecast sales data
- the *division_costs* database table will be used as a source of data relating to cost measures

Much of this data may be viewed at the aggregated division and company level, and consolidated for the year. You will therefore need to aggregate the incoming base data, from product (or division) and month level, up to the consolidated levels.

The *Unit Sales* report, specified as one of the output from the system at an earlier stage, could provide added value. This is if, as well as displaying figures for actual or forecast sales, it could also display a comparison of one against the other in the form of a variance calculation expressed as a percentage. Variances are also likely to be of use in the Divisional Comparison report.

The result of these calculations can be seen in the data structure:

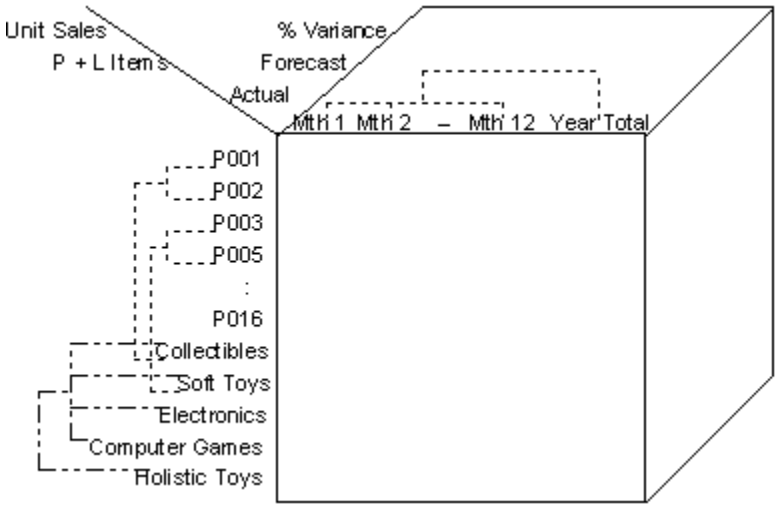


Figure 33

The variance 'measure' can easily be added to your components diagram, as an additional item in the boxed group containing actual and forecast.

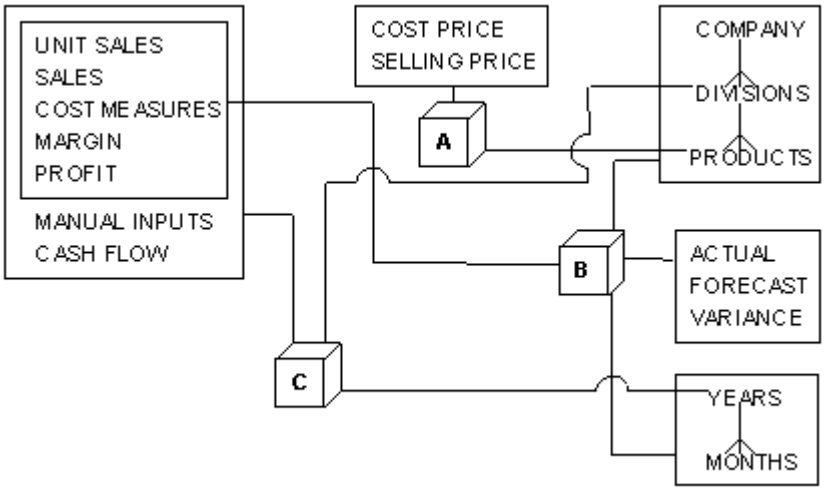


Figure 34

This structure will also require a number of further calculations to be performed on it, largely based around the derivation of the various profit and loss items.

You will be able to calculate the value of actual and forecast sales by taking the unit sales of toys from the structure, and multiplying them by the selling price from the product price structure (A).

Referring back to the diagram shown above, you can see that data structures A and B share a common axis, that of Products. Because of this commonality, OLAP allows you to easily relate the prices held in structure A with the units data held in structure B, for any particular product.

The results of this calculation can be stored directly in the structure, to make up the value of sales, as shown in the diagram on the next page.

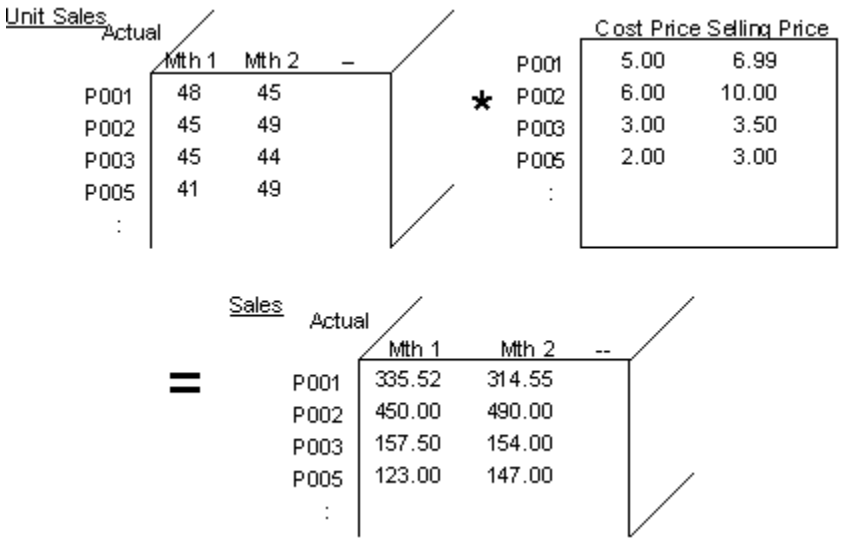


Figure 35

A similar calculation can be performed to derive the actual and forecast direct costs of sales in this structure. These values can be calculated by multiplying the unit sales in the toys structure (B) by the cost price from the product price structure (A).

You have now dealt with the value of sales, and the direct costs of sales, in the structure. As far as the other Profit and Loss items, you decide that you need more detailed information from Toys' personnel and conduct some further interviews.

Toys - Determining Profit and Cash Flow

The Finance department requires the ability to monitor the following Profit and Loss items, on a monthly basis for the current year:

> Sales, Cost of Sales, Gross Profit, Depreciation, Expenses, Operating Profit, Interest Payable, Net Profit, Tax, Profit after Tax.

For preparing five-year plans of profit and loss, the Finance department requires the divisional managers to be able to target the following value items:

> Sales, Operating Costs, Operating Profit, Depreciation, Interest Payable, Profit before Tax, Tax, Profit after Tax.

For preparing cash flow forecasts as part of the five-year plan, the Finance department requires the divisional managers to be able to monitor and target the following value items:

> Opening and Closing Balances, Operating Profit, Changes in Working Capital, Net Fixed Assets, Net Cash Flow from Operating Activities, Tax.

Concentrating on those items that will be used in the *toys* structure, that is, those that will be measured and monitored on a monthly basis, you have already defined calculations to derive *Sales* and *Cost of Sales*. Three of the remaining data items, *Expenses*, *Depreciation* and *Interest Payable*, may be sourced at division level, rather than product level, from the *division_costs* database table from the Toys ledger.

The company uses formulae to derive all remaining profit and loss line items, namely *Gross Profit*, *Operating Profit*, *Net Profit*, *Tax* and *Profit after Tax*. *Gross Profit*, which is calculated as the difference between *Sales* and *Cost of Sales*, can be produced at product level; all the other line items will be reported at division level.

You will need to set up calculations for each of these data items in the in-year profit and loss structure. You should also update your components diagram, to reflect these additional items.

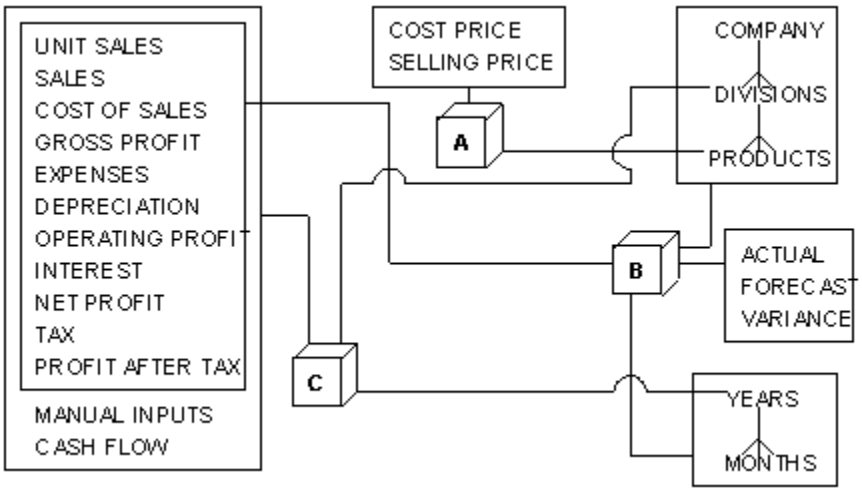


Figure 36

Planning

The three-dimensional structure that will store profit and loss and cash flow plans for the next 5 years, marked as C on your diagram, will also be based around a number of calculations and data manipulation operations.

Much of the source data for this year's figures in the planning structure will need to come from the in-year profit and loss data contained in the general toys structure. Therefore, you will need to set up routines that will allow the relevant data items to be easily retrieved from the latest in-year figures. As this may occur at any point in the year, these routines will need to retrieve actual data up to the current point in time, and forecast data for the remainder of the year.

As the five-year planning view satisfies a different requirement to the one-year monitoring view, the presentation of the Profit and Loss items also differs. These differences will be analyzed in more detail when you come to use the one-year structure to populate the planning structure.

You have already identified the requirement for an input screen that will allow divisional managers and their staff to type in additional information to be used in the planning process. Routines will have to be set up to store and process this manually input data, some of which will be used to calculate particular profit and loss or cash flow line items. The formulae that will be used to derive particular line items will depend, to a large extent, on the amount of data that will be able to be manually input to the system. This can be finalized during prototyping of the application development.

Updating your components diagram to include the additional line items required in the planning structure should result in:

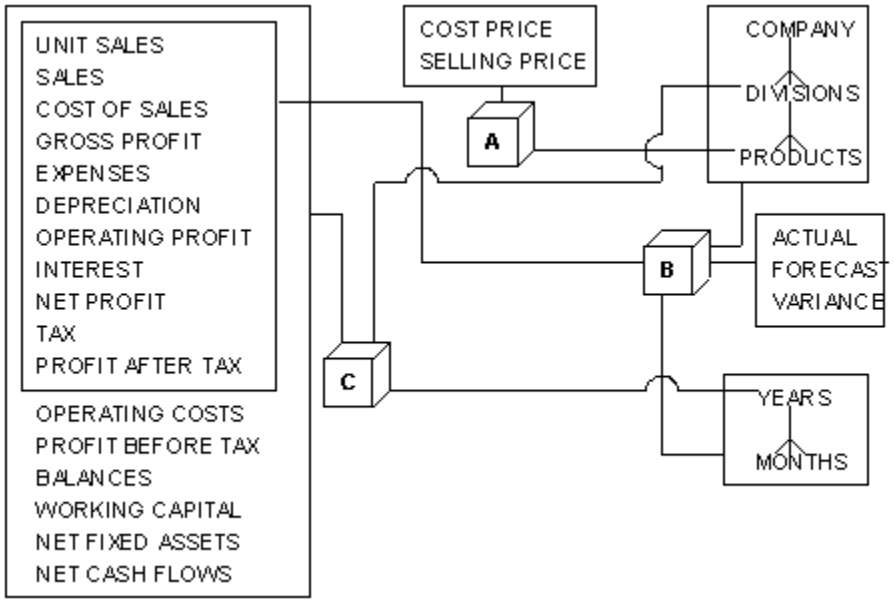


Figure 37

The analysis phase of your application is now complete. You have successfully broken down the business problem into a number of key components, and included these in a diagram which represents your logical design for the application to be developed.

Through the process of identifying the output required from the system, the input available to it, and the calculations which will need to performed as part of the application, you have been able to update the logical design to a point where you are ready to commence development of the application itself. The first stage in this process will involve translating your logical design into a physical one, and creating the objects that will form the basis of your application.

Common Pitfalls

There are a number of issues which can arise during the analysis stage of application development, which present additional challenges to the analyst. Being aware of potential problem areas, and some of the mistakes which can often be made at this stage, will help to ensure that you complete the process with a logical design that will provide for successful application development.

Scoping the Application

It is often tempting to try to target an application on a very wide user base, with a range of differing requirements, particularly when using OLAP for the first time. The power of the product, together with the range of potential uses to which it can be put, are likely to generate interest and enthusiasm within the business area.

It is important, however, to remain focused on the project in hand and to define realistic objectives for the application development. Trying to do everything at once can only complicate issues and is more likely to lead to a compromise solution, which is unlikely to meet the needs of the target users.

As a rule of thumb, it is often sensible for the first application which is implemented to be the smallest possible application which delivers real business benefit. The advantages of this are:

- **Simplicity** – the application will be as simple as possible and will provide a good foundation for the use of OLAP within the company.
- **Early feedback** – the application will provide a basis for early user feedback.
- **Quick delivery** – the time to delivery of the first application will be minimized, and with it an earlier tangible return on the investment.

User Involvement

It is important to maintain a high level of user involvement throughout the process. As your analysis evolves, you should take the opportunity to walk users through your logical design, to check your understanding and encourage discussion of the implications. A flexible design will allow changes or clarifications to requirements to be handled with ease.

This requires you to find the right people to speak to. During the analysis phase, you are sure to come up with a number of questions about existing data sources and company standards and procedures, among other things. The people who are best suited to answer these questions may not be those with whom you have had the majority of your dealings. It is up to you to find the people who can provide you with the best source of information regarding these issues.

One Structure to Suit All Requirements

One of the most common mistakes that people make during the analysis stage is to try to define a single logical structure to encompass all the data in the application. This can result in a large structure with many dimensions, only certain parts of which will be relevant to any particular data item.

The OLAP environment encourages you to define as many logical structures as your business situation requires. Your design should take into account the natural dimensionality of the data, as well as the requirements of the application. This will permit the application design to be a realistic representation of the business situation, making it much easier for it to be dynamic and data-driven, changing to reflect changes in the real world. Note that this concept is analogous to 'normalization' in the design of database tables.

It may initially feel that putting all the data into one large structure would be much easier; however, this is never the case in the long run, as data is always easiest to handle when it is structured in a natural way.

Data Sources

The quality of existing data, which will provide input to your application, often needs special consideration. You need to be sure that the application does not simply replicate data inaccuracies, which often exist in existing systems.

An added complication that often arises when developing an application, which gets its data from a number of different sources, is inconsistency. It is not uncommon to find that different systems have their own unique identifiers for particular items, such as product codes. These must be highlighted at an early stage so that all codes that identify, for example, a particular product can be 'translated' into a single identifier within your application.

In general, the issues of data accuracy and consistency can represent a substantial part of the project and the time and resources involved in this stage should not be underestimated.

Identifying Hierarchies

It is often the case that there are natural hierarchies contained within the data in an application. Identifying these at an early stage can greatly assist you in your understanding of the data, and prevent wasted effort later in the development process.

Having identified a potential hierarchy, you must then go on to decide whether the relationship between the data items is 'one-to-many' or 'many-to-many', as this will affect your logical design.

Ensure Requirements Can be Met

As you begin to design the logical data structures for the application, you should check each one to ensure that it will allow the reporting requirements of the application to be met in a simple manner.

This point can best be illustrated with an example. It is clear that there can be a hierarchy between months and years; holding this data together as one 'dimension' will make it easy to identify trends and forecast forwards. However, this design would not lend itself to a requirement to compare months from different years, such as:

	YEAR 1	YEAR 2	YEAR 3
MONTH 1	+	+	+
MONTH 2	+	+	+
MONTH 3	+	+	+

Figure 38

Maintaining a high level of user involvement throughout the analysis stage of application development will help to ensure that the resulting design meets the specified requirements.

Summary

The process involved, when starting to design a OLAP application, may be summarized by a simple flow chart:

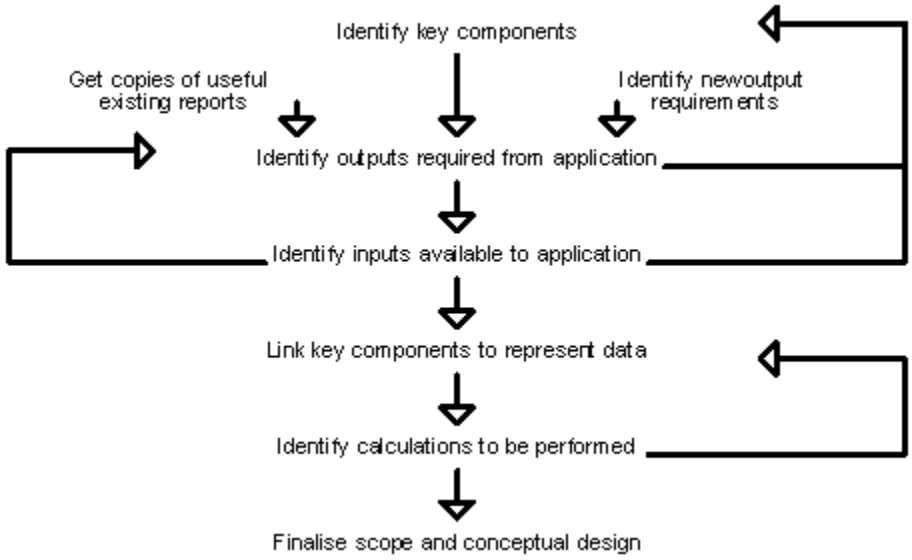


Figure 39

This process has led you to the following overview of the application, which will be required by Toys.

Toys – The Story so Far

Reports:

- Unit Sales report
- Divisional Performance report
- Profit and Loss report
- Cash Flow Forecast report
- Input screen

Data Structures:

- Product prices [product, price information]
- Toys [product, month, measure, profit\loss items]
- Planning [division, year, profit\loss items]

Calculations:

- Aggregation, where appropriate, from products to divisions, and from divisions to a company total
- Aggregation from months to a year total
- % Variance
- Sales
- Cost of Sales
- Various stages of Profit, including: Gross Profit, Operating Profit, Net Profit, Profit after Tax
- Tax
- Operating Costs
- Opening and Closing cash flow balances
- Net Fixed Assets
- Net Cash Flow