



Preface

The purpose of this paper is to give DBAs and IDMS development technical staff, and for that matter their immediate management, a perspective and a set of tools with which to fight the “perceived wisdom” that IDMS is old out-of-date technology that is fit only to be replaced as soon as possible. We know that it is not true. We have watched the product evolve, encompass each generation of technology, hardware and software, and absorb such industry standards as SQL in its stride. It is a fully modern product and this paper is intended to promote this fact.

The majority of IDMS professionals are convinced that IDMS remains effective, up-to-date and a powerful engine for processing and protecting vital corporate information. We perceive serious errors being made by corporate decision makers who are not in possession of this information. However we often do not have the perspective of the wider IT world and have great difficulty getting the message through to a business oriented upper management that they are making a business error, not just a technical one. To a large extent this is because we are untrained and unskilled in presenting our case in fiscal terms. A major part of this document is focused on how to use such tools.

The case histories in this paper are based on a synthesis of experience from many user sources. These sources are not named by their own request because there are many corporate pressures (business, legal etc.), which resist such identification. The IUA is acting as a vehicle to get these ideas out to members. Nevertheless, it must be stated that this is a set of personal experiences, not a scientific study. It is intended to give ideas of how the forces working against IDMS can be resisted.

Consequently, assertions are made which we have been assured are based on research but we are not in a position to support them. Some are views expressed in the trade press and the reader is referred to this source. Especially in the case histories, they represent at least someone’s perception, which the reader is invited to challenge. Above all, situations change rapidly in IT and any such assertion should be treated sceptically. Comparative prices must be re-examined always in the local market place to confirm what they are in any particular circumstance.

This issue of the white paper must also be regarded as a preliminary release of the paper which will evolve and improve as we get feedback and comment from IDMS sources.

There is concern in a number of quarters that the paper could be used by those who make their income in persuading the gullible to convert from IDMS. Certainly the IUA and the author hold the copyright on its content and permission is not granted to reproduce or distribute the paper in anything other than its complete form. The author and IUA Board declare our complete commitment to IDMS as a product with unique and completely modern strengths, and resist any suggestion that the paper supports any other view.

The word “legacy” has, at some time, been used as a pejorative for all products that are mature and/or based on mainframe technology. It is used by marketers to describe any product that has a certain degree of maturity as compared with what new item they are now selling. The reality is that



new concepts are usually riddled with problems and about the time the bulk of these are worked out, someone is going to represent it as a “legacy” product. Some have suggested fighting fire with fire. If “legacy” has a negative connotation as something inherited but not taken by choice, then “heritage” has the connotation of that which we choose to keep from earlier time as having enduring value. Bernard Shaw said “If you cannot hide the skeletons in your cupboard, make them dance.” So, calling IDMS heritage is to acknowledge that it is a mature product but one which continues to yield great benefits. This at least is the view of the author.

Above all, if we find ourselves in such a situation (and most of us are), it cannot be overemphasised that our local CA organisation has both considerable resources available to it and a very strong incentive to challenge migration decisions. Involve them in the process of resisting the forces for migration.

Introduction

The purpose of this paper is to review the facts, arguments and techniques that may help members to deal with the situation where their organisation is considering the replacement of IDMS with another DBMS.

We wish to find a strategy to defend IDMS but there are some basic questions that must be asked before we go any further. These are:

- What is the status of IDMS in the view of management?
- Who are the decision makers we need to influence?
- How best can we influence these decision makers?
- What is the landscape in which decisions need to be made?
- What are the battles we can win?

It is clear that the decision makers are our management. This stands to reason but does not answer the question: Who are the actual decision makers and who are those others who will most influence a decision? It is also important to know who are the advocates for one position or another. These are the people we need to persuade or counter if we are to have any impact.

In talking to management there is little to be gained by arguments that are not fiscally based, justified and explained. Budget projection, not technical excellence, will always win and as technicians we must learn to speak in these terms if we wish to be heard. Many of us lack (or do not wish to admit to having) the ability to use this monetary method of valuation but lose credibility by failing to do so.

We must have a realistic view of the current economics of computing before we can hope to contribute to the argument. Things are changing fast and it is necessary to keep up. There are sources that we can cultivate which will give us a view of this.

Finally, we need to establish the profiles of the situations we can expect to encounter and, in particular, the ways we can win the arguments.

Consider the following:

Some Perspectives

We have sought opportunities for IDMS customers to explain their companies' decisions to keep or abandon IDMS as a "strategic" product. Without identifying these organisations, it is useful to consider some of their business profiles and the arguments they have used.

STRONGLY SUPPORTED ENVIRONMENTS

Some examples are ones where the justification of retention of IDMS is relatively obvious, though these are frequently at extremes of usage. Less clear-cut (less extreme) cases may nevertheless take relevant aspects of these examples to incorporate into their arguments.

Case 1 Large scale large volume user

A major company using IDMS in a very big way has a particular inertia when it comes to changing the database environment because its investment is large. It has most likely reviewed its commitment to the use of IDMS on a reasonably frequent basis but has so far endorsed the status quo. However, the choice is rarely arrived at lightly and is made on the basis of the economics, as well as on certain key technical ground.

Nevertheless, they do recognise a number of serious negatives associated with using IDMS:

1. The cost of being on the mainframe platform is significantly higher than some other platforms providing similar processing capacity.
2. The cost of new development on the mainframe is most significantly higher when compared to that of many modern object-oriented environments.
3. The cost of maintenance, both of IDMS database administration and of the mainframe applications and environment in general are also high compared to alternatives.
4. The costs of a server based application tends to be spread across several departments, not just the IT department, masking the true costs.

However, against these, the positive factors in favour of IDMS are also clear:

1. The robustness and power of the IDMS database engine is difficult to match.
2. The reliability and 24X7 operation of the mainframe platform is also exceptional.
3. The high cost of conversion to take existing applications over onto some other environment is very difficult to justify.
4. The high cost of risk to continued business operation of any failure to effect a successful conversion is also seen to be very significant.
5. The costs of a mainframe application may be lower than a server application if computed on a "per user" basis.
6. The integration technologies of ODBC /JDBC, TCP/IP and MQ Series enable access the IDMS database in the newer programming languages.

One aspect of these large users that appears significant is that the operating set of applications is likely to be very comprehensive, reflecting well the existing business operations. These are also likely to be highly integrated making piecemeal conversion relatively more difficult. Some large users have attempted benchmark conversions but have

mostly failed to find viable way to match existing performance while minimising risk. Because of the completeness and size of the application functionality, it is also difficult to devise a conversion that would provide much more than a function-by-function replication on the existing environment, i.e. large expenditure for little value added.

Development of new applications is a different matter. These are most likely to be predominantly n-tier, with Intel/Windows machines dominating for the client side and having distinct advantages also as servers. This is an area where new mainframe applications are most unlikely to win in a head on contest. As observed above, development costs on the mainframe are significantly higher, so management is unlikely to agree to new application development on that platform. For this reason, it may have been some years since there have been new developments under IDMS or even on the mainframe platforms itself.

Such a stable system will require little maintenance intervention and involve little system or application change. If the mainframe applications are not significantly enhanced their costs tend to be low, and their robustness and efficiency are significant assets. In such an environment the higher hardware costs may be offset by lower on-going costs. There is also a cost associated with having multiple DBMS engines. In this role, existing IDMS business logic can often continue to be utilised, by being invoked from client or server applications. The tools exist to effect connectivity and are improving rapidly. Hence there may be the opportunity to keep IDMS strategic as the DBMS engine for the third tier of new applications.

To keep IDMS in such large users is an argument that can be won. If the IDMS database has evolved to represent the natural data structure of the corporation then conversion can at best be a copying from one platform to another.

Case 2 Small scale high complexity user

If a database is the provider and manager of information that has very complex structure, then we also find an environment where there are strong arguments for the retention of IDMS. Some examples which have been seen are: complex risk, policy, cover information in the insurance industry; many level manufacturing Bill of Materials data; and providers of complex business data, where information is recorded to be retrieved in a variety of ad hoc ways in real time and perhaps directly by third parties.

Such systems, based on a mainframe and either CICS or IDMS/DC, are well suited to Web hosting (e.g. Websphere), to support of third tier applications or web services. Conversion to a purely relational view is likely to fail to deal effectively with data complexity but such structures are the strength of the CODASYL data model. To this can be added the power of the database engine, its speed, reliability and recoverability and there is a strong incentive for such a user to remain with IDMS. New development may be aimed at the web or client GUI interfaces but a cost/benefit analysis of such a situation may well render the option of conversion to RDBMS near to impossible to justify.



Relational advocates will counter that a relational database definition can handle any data structure and this is true. However while a “pure” relational solution to such data complexity is possible with such DBMS as DB2, Oracle, SQL Server or even Access, it is certain to be much more difficult to design, maintain, tune and keep efficiently organised. This derives from the nature of the file/table architecture of these tools and the user has no alternative to choosing between longer retrieval and processing times, or a greater numbers of servers and database engines. The ultimate argument for us in such instances is to promote a carefully focused benchmark.

New application code, in this class of IDMS environment, is most likely to be developed on the client or server platforms with perhaps periodic downloads of some extract information onto servers. However the benefit of IDMS’s ability to handle the very complex structures can be capitalised on.

Case 3 Small but mission critical applications

It is possible to have smaller IDMS users whose key IDMS application set is nonetheless a strategic to the overall corporation. If this is based on complex information structures then it has similarity to Case 2 but it may simply be the criticality of the functions that makes it difficult to justify conversion. This is especially true if the reason for proposing change is based on other than solid business reasons e.g. because IDMS and/or the mainframe are unfashionable technology.

Here the relative cost of conversion is added to the estimated cost of risk of failure. In some businesses the cost of just the disruption of service can be unacceptably high. These sites may however feel very keenly the cost and scarcity of development staff for the IDMS applications and to counter this an education programme may be proposed whose cost is likely to be small against the other measures. A benchmark should probably be avoided if possible because it might just lend credence to the viability of a converted application.

MORE DIFFICULT ENVIRONMENTS

Equally there are cases where the argument is more difficult and justifications more finely balanced. The following typify some of these situations and the strategies which could help to retain IDMS in these sites.

Case 4 Bundled DB2 with high IDMS licence cost

IBM has seemed increasingly eager to entice IDMS users to their DB2 DBMS and have made this more appealing by bundling DB2 into the price of hardware upgrades. In itself this is usually not a sufficient inducement because the IDMS programs in general require individual conversions from network to SQL. This often causes a serious impact on application logic.

However, some IDMS to DB2 conversions have been indeed been successful but conversion costs are found to be directly a function of application size, the number of lines of code and of the number of programs in applications. So, where a customer has a small commitment to IDMS applications, conversion might become a justifiable business strategy. The size of application determines this.

A factor in such sites may also be the size of the licence fee for IDMS. Some older and larger users have very favourable licence fees but smaller ones may not. In these instances, the fee savings can offset conversion costs, especially if conversion is amortised over multiple years. Furthermore, if the user has aspirations towards n-tier applications then the cost of SQL Option licence must also be counted against the IDMS retention.

Case 5 Move to Midrange (esp. Oracle) platforms

There are similarities to Case 4 except that the option of converting existing business logic is virtually eliminated. The user has to do a complete development of new business processes unless of course moving to some off the shelf environment in the ERP model. The same issues as in Case 4 are still present but, if it is the proposal is to build a completely new custom applications on the new platform, then conversion costs will be greater. This may often leave open a good chance of promoting co-habitation arrangement for IDMS which stays on as DBMS engine for existing applications. The connecting technologies such as MQ, TCP/IP or ODBC/JDBC make this feasible but the DBA may need to be judicious about which application components to fight for and which to concede.

The wish to use ERP or COTS packages as part of a business re-engineering or in order to standardise applications, is a very common additional reason for such a down sizing. The cost justification of such re-engineering may be useful to review but certainly the costs of changing business processes must be part of the calculation

There is furthermore a perception that it is cheaper to move to smaller computing platforms because of the lower development and maintenance costs they are promoted as having. This may be balanced against the mainframe costs. These costs, considered in isolation are likely to be more persuasive for change than incorporating them into the overall costs to the business.

Such argument may at least delay elimination of IDMS and hence there remains a possibility that the midrange solution will be problematic in practice.

Case 6 Move to web/n-tier technology

The Internet revolution has opened up a major alternative to traditional application design, the multi-platform distributed network of co-operating programs (n-tier). If this is the chosen business strategy, comparison of costs will need to incorporate these costs. It can also be seen as an opportunity to renew the corporate business processes and realistic costing of this change must be included into the decision basis.



Since not large or complex, it is also likely that a prioritised, component-by-component conversion strategy may be possible which reduces the risk of conversion. Still there can be a role for IDMS as a database engine and business process server.

Methods for Influencing Decisions

If we have one of the above situations then we may need to present our argument to management. There are some basics to keep in mind.

Talking to Management

Some general points about the approach to take are:

- Technical excellence is not an argument.
- There are situations where the business case may be against IDMS.
- Technicians have to fight the tendency to reject the advice of local technical staff.
- Arguments based on costs and benefits are primary.
- Comparison of relative cost/benefit are more likely to be heard.

The language of decision making

As DBAs and technicians, IUA members are generally sold on IDMS because of its technical characteristics. We talk about its data modelling sophistication, its robustness and resilience, its throughput rates and response times, its 24X7 capabilities and its fault tolerance. In fact none of these are likely to appear as more than side issues to the decision criteria used by management, which will revolve around money. Even advantages of key technical functions must be justified in this way and all decisions made will have a financial basis. If as DBAs we respond to financial arguments with a technical litany we can only expect to be side-lined.

Currency is not our natural communication medium but that is not to say that there are no financial arguments to be made or that technicians cannot learn these skills. We may believe that a particular technical feature is a decisive element in a decision about corporate computing strategy but if it is really as important as we think then it must be capable of being quantified.

Understanding the Real Business Issues

We sit in our technical corners, do our work and resolve our problems, but do we understand the issues facing the corporate leaders? Do we know what competitive threats our companies face or how IT figures within their corporate strategy? I suggest that mostly we do not. Yet if we do not know these we are unlikely to be able to influence the decision process.

How do we know if such discussions and decision making is taking place? This is going to be different for different organisations but ultimately the answer is to get involved with our management by being proactive. It is up to us to get ourselves involved.

We are however unlikely to be experienced in dealing at this level and help may be needed. A strategy would be to find a mentor from within some senior level of the corporate

organisation. This would have to be someone who understands the processes and concerns of those at the top, knows their key issues, and also has knowledge of the skills and presentation techniques that are likely to get a hearing. If no internal mentor is available then it would be advisable to seek someone with business skills from CA or elsewhere.

Techniques

Some specific techniques are described below which will be helpful to make many of our arguments.

Cost/Benefit Analysis

The most basic technique is the financial analysis of costs and benefits of any option being considered. We are more used to thinking about decisions as a comparison of technical functions but here we must deal with estimates rather than absolute criteria.

The first stage is simply to list all of the costs and all of the benefits, being as inclusive as possible. Quantifying each, especially benefits, is the most difficult part of this process. There will be a level of uncertainty for each and a need to work with windows of cost and with stated estimation assumptions.

Clearly, the costs go on one side of the balance sheet and the benefit values on the other. Brutal honesty and avoidance of bias is needed.

Proposal:			
Convert XYZ application from IDMS to DB2			
Costs		Benefits	
Convert data structure	£10K	SQL compatibility	£5K
Convert data files	£30K	Avail. of trained staff	£10K
... (etc.)		... (etc.)	

Cash Flow

This method builds on the data of the costs and benefits analysis but is based on providing a timeline against which to measure, and this can have a significant effect on the financial feasibility of a plan. It can give a much more realistic view of the impact on the business and on budgets. In the case where all the costs are up front and all the benefits down the road (and also uncertain), it shows how budgets may be affected in the short-term and long-term, in ways that do not show up in a simple list of costs and benefits.

Cash flow may also be used to spread costs over a longer period as well as to show the impact of time-based charges. We may show the need to have available alternative strategies for spreading cost to support our proposals.

Type	Jan	Feb	Mar	Apr	May	Jun	Total.
Costs	-£20K	-£50K	-£15K	-£10K	-£5K	-£5K	-£105K
Bene.	£0K	£0K	£5K	£10K	£50K	£10K	£10K
Net	-£20K	-£50K	-£10K	£0K	£45K	£5K	-£30K



ROI

This also derives from the cost/benefit data and is simply another way of viewing this information overall. It shows how the cost of the real investment needed to follow a particular strategy will generate a measurable return for the additional money expended in realisation. It can demonstrate that is our proposed return is good value for money or perhaps that the corporation could get a better return putting its money into the short-term money market.

Funds Invested: £1,000K	ROI: £ 0.1 per £1 invested
Benefits: £ 100K	

Risk Analysis

One of the more difficult parts in making an argument is to deal with unquantified “fear, uncertainty and doubt” (FUD in IBM-ese.). Such issues may be raised both for or against any side of an argument. This can involve extreme possibilities like:

- IBM will get out of the mainframe market.
- CA may withdraw support from IDMS.
- CA may sell IDMS to an unacceptable vendor.
- The corporate computer is a terrorist target.

However they can also be more mundane issues such as:

- The hardware does not reach the manufacturers throughput estimates.
- Business paradigm change is forced by competition.
- User requirements radically change during the process.

All that can be done in such instances is to produce a reasoned argument for the probability of the risk and proposed strategies in such an event. Such risk analysis has taken much more prominence because of recent events and it is necessary to explore in general terms what options would exist.

This is a cost analysis where each item has a probability factor and strategies developed for each outcome.

Risk	Response	Action	Cost
New software design has fundamental flaw	*Retain existing s/w for 3 months	*Keep old h/w *Keep old s/w licence *Temp storage increase	£100K £85K £100K
	*Proof of Concept with a 10% prototype	*Delay start s/w dev. *Resources for prot.	£20K £45K
(etc.)			



Comparison of Alternative Strategies

It is not sufficient to do Cost/Benefits analysis, etc. for just one of the competing strategies but to demonstrate comparisons. In some instances there will be existing data provided by an advocate of a competing strategy. This can be incorporated into our numbers but not uncritically. The objective is to show the relative balance between multiple alternatives as objectively as possible. It may be that our IDMS strategy simply does not measure up and if so, it must be accepted unless we feel that we have under-rated something.

Comparison can combine with any or all of the above analyses as in this example of the multi-year comparison of licence fee against conversion for Case 4 above.

Yr	Cost of IDMS	Cost of DB2
1	Licence fee year 1 SQL Option purchase	Conversion costs year 1
2	Licence fee year 2 (incl. SQL Opt.)	-
3	Licence fee year 3 (incl. SQL Opt.)	-

The Management Summary

As important as anything else that we learn is how to make a succinct summary of the points, so that on one single page is a statement of our proposal, our findings, our conclusions, complete but reduced to its essentials. Subsequent pages may support the summary but the manager must be able to get to grips with the essence of a proposal in the short time we are likely to be granted.

It is difficult to overstress how important a single page summary can be as this decision moves up the corporate ladder. Senior management does not want to spend too much of its time on the technical details. It is making the assumption that this information is relevant or lower management would not be bringing this to their attention.

Sources of Information

The above are standard techniques, and models described in books and practical examples found in most corporations. This may be one of the areas where a mentor can help us to put our analyses into the format that matches the corporate culture. For specific hardware costs and price/performance figures, the vendors will most likely be willing to give figures though these must be set against each other quite carefully. Salesmen have a habit of comparing apples with oranges.



A Realistic View of the World

The Mainframe

IDMS lives and dies with the IBM mainframe and the mainframe is widely viewed as an expensive computing platform. This is for both for costs of hardware, software licences and people costs for development and administration. A carefully-sized mainframe which supports a database engine with minimal administration and no development, can compete with smaller platforms when balanced against conversion costs. We must remember though that it is increasingly difficult to recruit mainframe people. This argument can be countered by use of the ODBC and JDBC. Offshore companies are also offering reduced costs and can be explored.

The price/performance costs are not in favour of the mainframe based on a variety of sources but must be assessed in any individual case or time. Various guidelines found in trade press, from consultant reports and from the platform manufacturers and distributors. We should recruit our CA liaison into this process as well.

Other Computing Platforms

The problem is to compare apples to apples and what must be sought is a reasonable standard test of cost to performance as a basis for a realistic comparison. However there are some cautions regarding how processing costs are estimated. A properly designed application for a particular distributed design is going to include components of savings by spreading processing over less expensive nodes of a computing network. However it will also introduce new delays into response time and processing overheads due to the increased numbers of transfers between computing nodes. These must be balanced.

Nevertheless, we are likely to see that the mainframe is still significantly more expensive for a given price/performance measure despite the great progress IBM has made in reducing hardware costs. Experience suggests that the Intel/Windows platforms are least expensive, either as clients or as servers, and that a server with something like four processors is optimal.

One item in favour of a mainframe solution is that there are usually more server DBAs per application than there are IDMS DBAs.

Operating Systems

As with the hardware, operating system costs are weighted against the mainframe. The maintenance costs of Z/OS for example are very high and combined with say USS for web server capabilities, they can become difficult for cost comparisons. Z/OS is particularly a problem for maintenance because of the high technical level required for its support, maintenance and generation. These do reduce if the mainframe is very stable and used principally as a database engine. Staff acquisition remains a serious issue for management.



Conversion Costs

Each conversion is unique. It will be influenced by the comparative platforms involved, the languages and design of the application (old and new). The costs may also be offset by less tangible corporate objectives like the opportunity to re-engineer the business processes.

At one end of the range is the conversion from DBMS to DBMS within the same mainframe platform type. Here the code may be to an extent re-usable and justification will come down to perceived conversion savings against conversion costs.

At the other end of the scale we may be changing platform as well. Here there may be very little business logic that can be carried over and in that case the cost and risk is the same as a complete re-write.

It was noted above that in a case where little or no additional functionality results from the conversion, then the cost will have no mitigating element of value added. Of course, this may be countered by the view that the mainframe of itself is becoming a corporate millstone and that there is uncertainty about its survival, as well as in finding affordable staffing to run them.

Calculation of conversion costs is likely to be complicated and subjective, but at least such consideration is in our natural area of expertise.

DB2 Conversion

This is a special case when it is likely that all aspects are within the skill set of the existing IDMS staff to make estimates and calculations. It is also one where it is likely to be a simple apples-to-apples comparison.

The complexity of the network schema, as noted earlier, is likely to make the conversion of code to use SQL more difficult. Be prepared however for the proposal of some multi-step conversion strategy.

For example, the conversion could pass through the steps:

1. Making the IDMS schema SQL compatible.
2. Progressively converting applications to using SQL in place of network DML.
3. Converting the whole application over to DB2 SQL.

Such an approach may be more difficult than it appears and does require the SQL Option to be installed, but this type of approach has worked in the past.

Performance is likely to be easy enough to measure when a benchmark is performed. If management is reluctant, we can give some examples where DB2 has been beaten badly (e.g. examples where DB2 required more than twice the computing power or has more than twice the response time). In a fair comparison, the network database has the ability to shine but it is important for us to make sure that the comparison is not distorted by, for instance, the DB2 test leaving out troublesome substructures that would increase processing time. Because of DB2's essentially index-oriented architecture an IDMS network will usually prevail.



IDMS Licence Costs

The annual renewal of the licence for IDMS can be quite high and this allows the argument that conversion cost can be offset against licence fee savings. This is a year-on-year predictable cost as against other more subjective costs such as for conversion.

However, it is also something that can be clearly presented to the vendor as a problem to the survival of the relationship, see if there is not a strategy that can counter this argument.

Application Development Costs

That mainframe development costs are much higher and take much longer is a truism. However, use of techniques such as RAD, XP or Agile methods in relation to ADS can certainly moderate these costs in the case of incremental development on an existing application base.

ERP

Many corporations have bought into the idea of standardised corporate-wide application processes and this is the reason for using such systems as SAP. Since the standardisation itself is perceived as having a value, it can be difficult to do real comparison because of the subjectivity of quantifying this value. All that we can do to counter such a decision, is the realistic presentation of the costs and description alternative strategies.

It is a great shame that so little effective work has been done to encourage the ERP providers to embrace the IDMS SQL implementation.

Web Applications

Development of web-based applications has been given as the reason in itself for moving away from IDMS. On the contrary, the proposition that IDMS can become a heavy-duty database engine, means that the tools are already available to allow the mainframe be the third tier of such a structure.

SQL Option is essential since without it mechanisms like ODBC or JDBC are not available, nor are Table Procedures or SQL Procedures. An unrelated advantage is the new TCP/IP support and 3rd party MQ products. These together make a variety of solutions available for connecting and integrating client/server applications to a back-end engine and of making use of existing application logic investments.

This approach has many attractive aspects. Because IDMS system can be both database engine and business logic repository, the conversion to web applications can be achieved in a relatively risk free way. Only when an application component is fully tested and ready, will it replace the existing ChUI application interface and the older solution may co-exist with the new, until all users are converted. There may also be flexibility in having a permanent situation where both new and old continue to co-exist.



Modern Development Environments

One thing is certain, new IDEs (Integrated Development Environments) leave everything on the mainframe looking old and tired. Combine these environments with the undoubted advantages of object orientation and extensive object libraries, and it is an unbeatable combination for application development.

A tool like ADS can achieve some of the productivity of these new environments if combined with modern RAD, XP and Agile methods, and incremental development techniques. It is not necessary to have the formal structure of an “object” to make use of hierarchies of subroutines linked around standard APIs. IDD allows dictionary entities such as records and code to be linked under a name just like an object, thus acting as a repository. It may not be as elegant as an IDE but it can get the work done.

What is certain is that modern corporate management is not willing or able to tolerate the IT development cycles of the past. Probably only limited development or enhancement is realistic on the mainframe.

Finding a Strategy

We have reviewed the issues, tools, etc., now we must define our strategy. It is necessary to examine carefully which aspects of the above we can use in defending the IDMS “heritage”. Here are some thoughts:

Know ourselves

We should review the assertions often made about IDMS, understand what they mean and be prepared to promote or refute them:

Note * above: scalability means different things in the two environments. On the mainframe it usually implies the growth in processing power of the processor or matrix of processors; in non-mainframe it means the ability to install more servers to clusters.

Pro:	Contrary:
<ul style="list-style-type: none">▪ Proven technology▪ High power/volume DB engine▪ Highest technological level▪ Mainframe power▪ Supporting full SQL database▪ Supporting full network database▪ CA support▪ Scalability (mainframe)*▪ Low cost of a stable system	<ul style="list-style-type: none">▪ Seen as yesterday’s technology▪ Difficulty of use and maintain▪ Old fashioned non-RDBMS▪ Mainframe costs▪ Licence costs▪ Scalability (non mainframe)*▪ Development costs▪ Maintenance costs▪ Hard to get/keep staff

Know the Enemy

We need to review the facts about the other options in the frame. These must be real facts, not just anecdotes. CA and other vendors, plus trade press can all help.

Understand Management Perspective

We must understand the reasons behind a management wish to change DBMS. Businessmen do not usually go out and spend lots of money capriciously (though it has happened), so we must discover what is causing them to contemplate such an expensive course of action. Reasons may be some of the following:

- Hardware costs.
- Software costs.
- Development costs.
- Staff costs and difficulty getting staff.
- Development times.
- Perceived advantages of new technology.
- Perceived limitations of IDMS or the mainframe.
- Business re-alignment (merger, take-over, etc.).
- Competitive pressure.
- Business process re-engineering.
- ERP

Whatever it is, we must find out these root causes. This needs to be done as straight fact finding, not arguing any case at this point. In some situations, the real reason may not be stated (or perhaps even known explicitly) by the management, and it may take some digging to come up with a complete list.

In some instances there may also be a difficulty in discovering who to deal with in IT management because the real organization structure is unclear. The person who will make the decision may not be the obvious one or the decision may be abrogated to someone such as a consultant. It is important to know that person.

Develop a Strategy

Knowing now what the issues are, one or more strategies can now be developed to deal with the issues.

We must complete the financial costs and benefits as completely and honestly as possible. Here is where our mentor will be invaluable to build up a consistent argument. There is some value to having the opposition put in their figures first but in the end we will win by having the best argument.

Selling Management

We should keep in mind these simple rules:

- Keep it simple.
- Focus on the REAL issues.

- Have the facts.
- Know the competition.
- Focus on money/resources.

Financial Arguments to Make

Conversion Costs

This is probably the best route of justification because it is very often underestimated. It includes hardware, software, personnel, process re-engineering, etc. and should include risk management costs as well. The costs must be trusted by all parties involved. If no agreement is possible, then we should put our argument to paper.

Fail Safe Costs

This is a contingency argument which points out the cost of coming back if the attempt to convert fails.

Leveraging Application Investment

How does the corporation benefit from moving working processes to a new platform if it is at the cost of one that is fully working on the mainframe. Strategies for saving this investment can involve keeping IDMS in house to continue to provide information and business process logic. No organisation should make such an investment without some measurable value added.

Propose a Benchmark

This has the potential for being a body blow against the opposition but must be done intelligently. It should be a final resort and normally be proposed only when there are specific assertions of capability which need to be challenged, such as:

- Hardware throughput rates
- Software load levels
- Ability to do specific functions

The trouble with benchmarks is that they can cost a significant amount of time and resources to perform. This has to be justified to management against the cost of failure of the changes proposed.

If we do enter into such a contest, it is necessary to design the benchmark carefully in order to avoid biasing it towards a particular outcome. There need to be specific objectives to be measured and truly comparable tests when the tests cannot be on the same platform.

Test Key Volume and Response Issues

Clearly this is one of the technical strengths of IDMS but is also an one where proponents of smaller processors have argued superiority because they can much more simply grow the computing power by adding new, low cost servers. In reality there are many cases where these numbers are grossly underestimated by them. Examples of horror stories can be found.

Application Development

To suggest doing development on IDMS is a difficult argument to undertake, not least because other platforms have cheap efficient OO environments which make development much simpler and cheaper. However if there is a new facility being added which could go onto either platform, it may be argued that it should be put under IDMS.

Web/Network Services

The concept of a Web Service is popular amongst the proponents of n-tier solutions. It is not beyond the capabilities of IDMS applications to create appropriate APIs to build these or at least to interface with service routines on a server. Features like Table Procedures are a viable interfacing mechanism. This allows IDMS code to have a vital place in the new n-tier applications

Costs of Multiple DBMS Maintenance

This must be used with caution because it could go the other way but where IDMS has a large established base of applications it can argued not to introduce new database engines. ERPs can be a special problem because they can be bundled with particular third party DBMSs.

Size and Complexity of Database

These are technical features and the representation of their benefits is not trivial. If such a situation exists, it seems that the best strategy is to fight for a benchmark or pilot test.

Outsourcing

Outsourcing may be considered as a way to reduce operating costs but can be a costly solution if not carefully contracted. By transferring some functions to a new location where running costs, maintenance costs and provision of trained staff can be more economic, this can possibly defuse arguments against the mainframe. It is not an option to go into unwarily but there are cases where it has worked.

Some final thoughts

Statistics and Sources

It cannot be assumed that the information about such things as the relative costs of operating a particular computing platform will be exactly as described here in

specific sites, geographic locations or changing technology and market. We can use them as guides but should confirm the in the specific case.

There are a variety of sources:

- The trade press
- Consultant reports
- Back issues of publications
- Monograph publications
- Technical journals
- Computer Associates
- IDMS Development
- IBM

However we must never underestimate the information we can get from the IDMS user community and IDMS-L is often the fastest ways to find people with existing work on this or similar subjects.

The Role of CA in Promoting IDMS

It is pointless to bemoan the lack of marketing effort on IDMS by Computer Associates. They are a business which must allocate resources according to the business priorities that they perceive. However, there have been signs that greater effort has been made in this area and the recent reports of the views of the new CEO, John Swainson, may mean that CA will become more active. This must be encouraged and seeking the assistance of our CA representatives can help to do so.

The difficulty of getting qualified mainframe staff

It is a well known fact that the community of IDMS trained and experienced technical staff is shrinking due to people moving into other areas, retirement, and few new people coming in. This is important in the fight to keep IDMS viable because it becomes increasingly a factor in persuading companies to abandon it. Visual DBA can help by allowing someone, who is more comfortable with the PC, to perform at least the common administration functions.

However in all this discussion it seems to be forgotten that new staff can be trained and with the latest releases the processes have become more user friendly, so that process is easier than it used to be. Finding staff may be a problem but it is just using it as an excuse if the process of training new staff is ignored as a solution.

The Wider Problem

The comment has been made that we are being too IDMS and database focused in the above discussion and that there is a problem in the whole issue of how corporations choose technology. Many of the opinions people carry are not based on fact because the facts are not available or are too hard to develop. The approach that is most common is to select whatever is fashionable. It was said in earlier times that no-one lost their job choosing IBM and while this may not be true today, we can substitute the current received wisdom of IT fashion and the sentiment still holds.



The decision makers have a vested interest in choosing the safe option and it is safe to do what everyone else is doing. It is possible to counter that IDMS is safe because it is well tested and tried, and has a low risk of failure. However fashion in technology can a hard thing to fight.

Conclusion

There are arguments that can successfully be made to keep IDMS in our companies in at least some strategic role. However they will not always work. This paper has sought to review the issues and the means to making the arguments, and will hopefully help in dealing with this difficult corporate issue for IDMS users.