IMS/2 AT LOLA[1]

K.W.S. LEWIS
Database Manager , London On-line Local Authorities
Chairman, UK Guide IMS Project

Presentation to:

NCC DATABASE SOFTWARE CONFERENCE
Bristol, November 1973

1 This document was scanned from a contemporary photocopy, OCR'ed, proofed and formatted similar to the original style, plus additional footnotes, in August 2016 by Alan E Cooper. Unclear and additional words shown in [ ].

1. Introduction

This paper presents an overview of a total operational information management system under development for a consortium of local authority users. It examines specifically the implementation of a long term data base strategy based on IBM's IMS/2 Software. The implementation is considered to be the most advanced on-line system in British local government today.

Key themes in the development are:

i. An analysis of the data flows within the user areas to identify an integrated corporate data base for the long term.

ii. Supporting software to allow implementation of this data base in an evolutionary way.

iii. Recognition of data as a company resource in itself and the use of on-line updating and retrieval technique.

The participants in the consortium are the London Boroughs of Haringey, Hackney, Hillingdon and Tower Hamlets (foils 2 and 3), with a total population of nearly one million people, and an employee work force of some 34,000 people.

Installed hardware (foil 4) includes a l024K byte 370/158 with 20 2314 disk drives and 10 P.O.[2] leased lines supporting 40 2000 character visual display units. By June 1974 there will be 75 terminals situated in 25 locations.

2. Background to the LOLA Project

In 1967, the London Borough of Haringey commissioned the London Boroughs Management Services Unit to investigate the long term data processing requirements of a local authority. Some 10 man years of effort resulted in the following analysis*[3]:

- The general need for increasing emphasis on and use of information as a resource in itself.
- The requirement for a data base for the local authority which would be of use to all the legitimate interests concerned.
- That the volume of information being processed would continue to increase, as would the duplication of effort between different departments maintaining similar data.
- That using a manual system the efficiency would decrease, and further the quality of staff required would not be available.
- That an on-line computer system with corporate data bases would be required to handle this information.

Estimates (in 1969) indicated that at least £1m[4] of computer equipment would be required for this development, and some 80 man years of effort to develop a nucleus of applications.

A consortium of four boroughs was then formed to pursue the development, with the costs shared between them.

Fundamental to the long term data base thinking (foil 5) was the concept that all data which was used by more than one area would be stored once only, and cross referenced to be accessible in other ways. Three classes of data form this hub, from which individual applications draw as required. They are -

- Property: For example rating hereditaments, council houses, premises registered for various commercial or industrial uses.
- People and organisations: For example ratepayers, creditors, council tenants, electors, school children.
- Internal resources: For example revenue and capital budgets, staff, stores holdings, transport, financial investments.

Application data bases link in peripherally to the central indexes.

Clearly, one could not contemplate implementation of such a complex set

2 [Post Office, who at the time also ran the UK telephone system. 2016 note.]
3 *London Borough of Haringey long term computer project - report on the initial study.

4 [About £15m (US$ 20m or Euros 17m) in 2015 prices. 2016 note.]

of linked files at one point in tine, and so a major feature of the data base software would be its responsiveness to change in the data bases as the total system was implemented in an evolutionary way. We recognised also the need to protect our investment in programs, if and when the physical data base structure had to be modified.

3. Software Selection

Some of the features required in the data base management software are itemised in foil 6.

An important requirement was program/file independence (foil 7). This is the ability for an application program to see a logical picture of only the data which it requires to process. The mapping to the physical representation is then handled by the data base management software, allowing the physical data base to be changed as a result of a changing pattens of use.

Another major requirement was support for cross referencing of data automatically - the logical data base concept. (Foil 8).

There were, of course significant requirements of the data communications software as well, though these are not addressed in this paper.

The project timescale allowed only two years to implementation of the first on-line data base application. Although we considered developing our own software or having a manufacturer develop the software for us, we decided that the correct approach was to use packaged software which was already available. At that time IMS/2 was the only relevant software, and so the machine order went to IBM as well.

Events later proved how important this decision was. To have written own software would have had a significant cost in manpower and in delayed project implementation, and even IMS/2, with 200 man years of effort put into it and IBM's highest level of support (Class A) still had a significant number of bugs, as discussed later.

4. Organisation Structure

Foil 9 indicates the structure adopted for the organisation. Control of the data base is centred on the Application Support Group, which represents a new departure in DP terms. This was one of the means by which data security was ensured, since a PSB control block[5] had to be produced for any program processing the data base. Control could then be exercised also over logging of data base use.

It was found essential that the Application Support staff worked closely with the system programmers in the Technical Support Group, as the responsibilities of the two groups were designed to overlap in certain areas.

5. The First Application System - Rating

The rating system[6] was the first application we tackled because it provided most of the data needed to set up our central indexes of people and property. The system was designed so that most of the day to day work of rating office could be carried out by on-line updating and retrieval. Batch processing would only be used for large billing runs, and the processing of high volume payment data. The project had an in-built overhead since in addition to building indexes for future use, it was the installation's first exposure to IBM, OS, PL/1, data base and on-line updating via a TP network.

5.1 Timescale and Resources

The development of a major application using IMS with both teleprocessing and batch programs has a long timescale, even in an installation which already has experience of such work. In our case, it took [2][7] years from the decision to use IBM hardware and software through a live running in April l972.

---

5 [Program Specification Block that describe the use of logical terminals and logical data structures used by application programs. 2016 note.]

6 [Rates were the council's principal source of income, being a tax on properties. 2016 note.]

7 [The order for the h/w & s/w was placed in May 1970. See *Experiences in Using IMS 2 for a Consortium of Local Authorities*. 2016 note.]

Our first application included an overhead for creating the IMS system and learning how to design data bases and use IMS, in addition to the problems described above. The major allocation of resources was

System design and programming	20 man years
Data base design, IMS & 05 support	19 man years
General purpose facilities	6 man years
	---------------
	45 man years

This includes about 7 man years of support provided by IBM. For some time, there were at least four full time SE's plus a software CE, all very competent, working on the account. This was a heavy commitment to a single 360/50 user.

About 2,000 hours of machine time were used. For a subsequent application of equal complexity, we would expect these figures to drop by over 50%.

5.2 Data Base Design and Creating

Foil 10 & 11 give an indication of the complexity of logical relationships involved. For each borough, seven physical data bases were required, and using HISAM, HDAM and HIDAM access methods four of them had no less than 15 logical relationships between them. Four more logical data bases were built upon these, to be used by the application programmers. For performance reasons, the correct technique out of more than 24 possibilities had to be selected for each of the logical relationships[8], together with access method and physical pointer arrangements for the physical data base structures.

For physical relationships, combinations of hierarchical and physical child/physical twin relative disk address pointers were used. Hierarchical pointers were used where the major processing involved sequential reads of groups of segments; physical child/physical twin pointers when access to specific segments below root level was required and where logical relationships were involved.

8 [See document *LOLA's Database Rules*. 2016 note.]

Logical relationships were implemented using combinations of symbolic key pointers and direct address pointers with physical pairing and virtual logical pairing.

Fundamental to the data base design work was information on the way in which the application programs were to process the data, both in the major batch programs and in on-line transactions.

Another area to be addressed was file conversion. It was necessary to develop a suite of programs (foil 12) to manipulate and load the data to separate physical data bases, which after resolution of symbolic pointers and updating to insert direct address pointers, produced the necessary logical data bases.

5.3 Problems with IMS/2

The rates project overran significantly its manpower and machine time budget. A vast amount of additional support work was required to handle the following problem areas. (Foil 13).

5.3.1 Education and Manuals

We began at a stage when there were no formal courses for IMS2, and with manuals which were badly written. Many weeks were lost before we could get down to real work, in obtaining clarification of fundamentals such as:

- Contents of I/O area for a LC/LP concatenated segment
- Insert, delete and replace RULES
- Implementation restrictions for logical data bases
- Extent of program/data base independence

In support terms, particularly in the early stages we felt that our IBM SE's were learning with us, rather than well ahead of us.

5.3.2 Data Base Design and Proving

Originally, the problem was that we knew what the data bases should look like from a program point of view, but we did not know how best to implement the logical relationships, because of the multiplicity of ways and because of the two particularly badly written chapters in the System Application Design Guide and Utilities manuals. We firmed up our design using physical pairing to crosslink each of four physical data bases, two in HIDAM and two in HISAM. Immediately we began application program testing, we entered a period of 2-3 months of grave difficulty with DFSDLR0, the logical retrieve processor. Errors ranged from abends, through bad status codes to simply wrong data returned, and it became clear that we were the first users to attempt full blooded logical relationships with HISAM. After many fixes, plus 2.1 and switching the data base access methods to HIDAM, we got going again with our application program testing, having lost a considerable amount of time.

5.3.3 Program Design and Testing

We recognised early on that a very significant part of the CPU cost of running our programs would be accounted for by the DL/1 calls which formed the interface between the programs and the data base. DL/1 is designed as transaction driven software, and apart from the overhead of a generalised package, this added an extra penalty. Therefore it was desirable at program design stage to review the application program DL/1 call sequences with a view to minimising CPU cost.

Also batch programs had to be designed to handle their own checkpointing, since some of them would run for several hours. There was no such facility in IMS.

In the teleprocessing programming area, the ability for a program to hold a 'conversation' with a terminal user was a new facility in IMS/2 which LOLA relied on extensively. The local IBM support were totally inexperienced in this area, and we had to develop our own conventions for handling scratch pad areas when switching from one conversational transaction to another. We had also to ensure that any updating carried out in a conversation step was logically consistent, as if the conversation subsequently abended, only the last step could be backed out [automatically by IMS].

The biggest problem in transaction program design was caused by the system limit of number of transactions which, divided over four boroughs plus testing gave just over one hundred transactions per borough - considerably less than the number required in the long term plan. This meant that programs had to handle many separate functions, they were large (up to 300K), heavily over-layed and slow. The limit on the number of transactions thus impacted system performance very significantly.

For transaction program testing, a stage was missing between module testing and final on-line testing, since no batch simulator facility was available with IMS. Several programmers at terminals using some 300K of real core for a test IMS system was far too expensive, yet the IMS system was not robust enough to handle program development with production running.

5.3.4 Error Recovery in Batch

Although a considerable amount of effort was required to produce and test operating procedures to be used in error situations (foil 14) in the main the programs were provided by IMS2.

5.3.5 Data Base Maintenance and Re-organisation

Because of the complex logical relationships in our data bases, data base re-organisation took a significant amount of time (17 hours for 2½ 2314 packs[9]). Some data base errors are not detected by the utilities until prefix resolution stage, well to the end of a run. Two successive re-organisation attempts failed in this way due to software errors corrupting data. It was fortunate that the corruption was in a key field, otherwise corruption of data in the data base would have gone unnoticed.

A more serious problem subsequently occurred as a result of errors in the prefix resolution and batch backout utility programs, concerning counter maintenance. Prefix resolution produced counters which were too high, whilst backout produced low counters by failing to apply counter maintenance records correctly following a logical delete. Hit one of these and IMS abended on you there and then!

Although the original software bugs were rapidly fixed by IBM, we were

9 [Each 2314 disk held 29.17 million bytes of data. There were 3, 6 or 9 disks in a 2319 B disk sub-system. 2016 note.]

left to clean up the data bases ourselves. IBM recommended that we write our own unload/reload programs. We did: they cost in excess of 6 man months to develop and more than 80 hours of machine time to run.

5.3.6 Extra Facilities

We found it necessary to provide extra facilities in order to produce a working system. A very significant amount of support programming was expended in this area.

5.3.7 Maintenance

With new release levels arriving every 4 months on average, and maintenance releases every 8-9 weeks much effort was required to keep the system up-to-date. In order to implement version 2.2 into production, no less than 60 IBM fixes and 22 local fixes were applied to bring this level to a production state over a three month period. Then, however, the software ran almost bug free in production use.

Despite all these problems the rates project went live on time and has been running in production now for l8 months.

6 Further Applications

The development plan has three elements, reflected in the following implementations.

6.1 Major Projects

We have just installed a comprehensive Financial Management application, including the areas of accountancy and budgetary control. The application, with a further 5 physical data bases per borough, forms another major part of the central files in our data base strategy. It represents a large part of the internal resources area.

We shall shortly be implementing a system for Purchasing Control and Payment of Creditors. This system draws from the central indexes of people and properties for the names and addresses of people and organisations seen as creditors by this application. Amendment of the existing data base to incorporate the extra pointer segments and logical relationships has proved to be a straight forward task.

6.2 Development of Facilities

These are based on rapidly developed extensions to the existing data bases giving a speedy implementation and quickly realised benefits to the users. An example is the Rates/Rents interface - to make selected rating data relating to council-owned property available to the Housing Department plus facilities to hold details of council tenants.

Other examples are:

- Property physical characteristics, a further expansion of the property index to assist the Housing Department in the areas of Lettings and property maintenance.
- Property progressing and monitoring, facilities to allow a variety of departments to record significant activities against properties for progressing and 'diary' keeping requirements.

Similar possibilities exist relating to the people central index.

6.3 Use of existing application packages

6.4 Development rate

Fast development on a broad front is now achievable due firstly to the evolutionary approach in our total data base concept. Secondly we have built a nucleus of data to support one application area. That data is accurate. extremely relevant for and can be made available to other legitimate users within the local authority.

This is the real pay-off from a data base system.

7. The future of IMS

Having implemented the early phases of our long term plan to build an Operational Management Information System, we are convinced that Data Base/Data Communications is the technology which we need to maximise the value of information as a corporate resource. We in DP must never lose sight of the fact that we are here to support the end user.

Certain additional facilities in the supporting software are still required to enable the job of data base administration to be done effectively. There is presently no effective means for providing program/file independence for sequential files.

There is also a need for an automated data dictionary. However we believe that there exists a problem of a different perspective.

IMS, along with other fully fledged general purpose DB/DC monitors is expensive in processor power. The 360/50 upon which our system first went live quickly became overcommitted, and it is now clear that our 370/158 throughput will also be limited by the capacity of its CPU.

It is possible to relieve the main frame of teleprocessing network control and ultimately message queuing by installing a TP front end machine. VTAM will go some way to meeting this objective. But data base management too is CPU intensive, and as noted before DL/1 does not perform well in batch. What is required is data base 'front ending' technology. Thus the main frame would request a logical segment from a data base in the same way as it would obtain a queued input message from a network front ending machine. Such a philosophy would also lend ease to the implementation of localised data bases perhaps at factory level, with a corporate main processing machine at head office.

8. Experience Gained

The fact that our projects have gone live on time, been accepted by user departments and have run successfully in production for more than 18 months proves that IMS2 DB and DC can be made into a working proposition. We have gained a vast amount of experience from our involvement, which will be of significant value in implementing future projects quickly.

From a programming point of view, writing programs to process data bases is a comparatively straightforward task. No longer need a programmer think in terms of file structure, overflow records and so forth. He can concentrate on the data which he needs to see and that alone. Thus we also benefit in that extra segments or logical relationships can be added to physical data bases without impacting application programs. In the same way we carry out performance tuning modifications without impacting running programs.

From the end users point of view the fact that all the data is held together with no redundancy means less work. Only one set of data has to be maintained and updated and there is only one level of currency. More important perhaps is the fact that his data is available immediately, and that updates can be completed in seconds in one train of thought. This allows the user to be more effective in his job and actually increases his job satisfaction.

However, quality control of the IMS software is an area about which we are still very concerned. With such a complex package, the greater is the exposure unless the software is extremely reliable. Early releases of IMS2 were far from foolproof - some 2,800 APAR solutions have been produced by the Development Team in a period of 21 months. On the other hand, IMS is slowly becoming more comprehensive, and future users should not have to expend as much support programming effort as we did, since more and more of those facilities are becoming available in the package or as IUP's.

We draw the following morals from our experience (foil 18):

* A nucleus of IMS experts is an essential prerequisite to successful implementation. To these, add all the technical SE support from IBM which <u>you</u> think you need.
* Then obtain expertise by allowing support staff evaluate the package for a while. Otherwise mistakes will be expensive.
* Pick a non-critical batch project as a live pilot study.
* The logical data base and program design phases are critical to the success of the project. Effort spent validating your designs using the appropriate tools will repay handsomely in eliminating future pitfalls and providing a system which will perform well.

9. Conclusions

We believe that the viability of the data base concept has been proved out at LOLA, but at a higher cost than was necessary.

We are now at a major point in our development, we have a springboard for rapid advancement. The next year should justify the far-sightedness of our constituent authorities and their courage in supporting innovation.

Tower Point North,
Sydney Road,
Enfield,
Middx.

K.W.S. Lewis
November 1973