Survey & Statistical Computing

LATS 2001: London Area Transport Survey

LATS Database Design Study

Database Design Report

Prepared for:

LATS Unit, Transport *for* London

Survey & Statistical Computing
119 Florence Road, Stroud Green, London N4 4DL, UK
Andrew Westlake, MA, MSc, MBCS, FSS

Phone: +44 (0) 20 8292 2005
Fax: +44 (0) 87 0055 2953
E-Mail: AJW@SaSC.co.uk
Web: WWW.SaSC.co.uk

Document Control

Project Title:	LATS Database Design Study
Document Title:	Database Design Report
File Reference:	\\Tower\Client Home\Katalysis\LATS\LATS DDR Pub.doc
Author:	Andrew Westlake
Other Contributors:	Miles Logie, Hugh Neffendorf
Reviewers:	

Issue History

Version Number	Date	File	Comment	Distribution
1	9/6/2001	Database Design Report.doc	Draft, section 5 not completed	MC, HN, ML
1.1	14/6/2001	Database Design Report.doc	Draft, new section 5 (not complete), section 6 in outline	MC, HN, ML, TfL discussion
1.2	27/7/2001	Database Design Report.doc	Draft, Section 5 complete, section 6 and glossary not much improved.	MC, HN, EM, ML, TfL discussion
1.3	18/8/2001	Database Design Report.doc	Complete Draft	MC, HN, EM, ML
1.4	8/11/2001	Database Design Report.doc	Final Version	TfL
1.5	22/11/2001	LATS DDR Pub.doc	Public version	Public

Acknowledgements

Many of the ideas presented in this report have been developed following discussions with other workers in the field of statistical databases. Particularly worthy of mention are Chris de Vany and Reinhard Karge (in the context of the IMIM project), and Simon Musgrave and Jostein Ryssevik (and their colleagues in the Faster project). Most of these people are now involved in the MetaNet group (which is attempting to coordinate advances in statistical metadata), and Survey & Statistical Computing is a Partner in that team. All these activities have received funding from the EC through the DOSIS and EPROS programmes of Eurostat.

This report has been developed with the guidance and assistance of the staff and consultants in the LATS team. However, the conclusions and recommendations are the responsibility of the author, and neither LATS nor TfL has committed to implementing all or any of the structure described here.

This version of the report has been prepared for public distribution. A few, sensitive, items have been omitted, compared with the version delivered to TfL as the final report of the project.

Contents

Tables

Figures

LATS Database Design Study

Database Design Report

1 Introduction

The LATS[1] database system is intended to be a dynamic resource containing information about travel in London. It will contain information about demand, use and attitudes and will cover all modes of transport. It is complementary to various other databases about transport facilities in London, and will have facilities to co-operate with them.

This document presents various issues relating to the design and use of such a database. It considers the objectives of the database, the use and users of the database, the main requirements for features and functionality (with considerable detail in some areas), and some technologies relevant to the implementation.

It is the main output from a design study to investigate the architectural and functional characteristics required for the database. This report does not attempt to make estimates of costs for the various task components proposed, since it has not gone into that level of detail on implementation. The review is wide-ranging and LATS may wish to adapt the programme considerably. Once LATS has developed its' more detailed requirements, it will be appropriate to establish the cost estimates.

1.1 Contents

All parts of this document address the underlying issues of:

- the material available to form the content of the database,
- the data organisation required to hold that information and give the flexibility to add new types of information as needs arise,
- the technical functionality required to manipulate and explore the information resource efficiently, and
- the interface functionality and resources needed to support effective use by different classes of user.

The report also considers some of the structural and functional aspects of current technologies for handling different types of statistical and transport data. These are related to the specific requirements of the LATS database system, with identification of useful ideas and missing features.

Other chapters identify specific requirements for the system.

- Content: the different types of information available for the database are discussed, in terms of both their structural and functional requirements.
- Functionality for using and maintaining the resource.
- Classes of users and their different requirements for access and functionality.
- Access control, including confidentiality and data protection issues.
- A summary of the conclusions and implications from a separate report about synthetic estimation.

[1] www.lats.org.uk

1.2 Key Points

- The overall vision is ambitious and will involve research, development and implementation over a period of years.
- The development of the full system can be viewed as a series of phases, each of which will produce a self-consistent system, allowing a sequence of decisions about continued development.
- Interim deliverables are required, particularly in the early phases, so that the database can be useful as soon as information is available. These will concentrate on clean, weighted, observed data.
- Further development will include synthesised data to provide improved, consistent estimates with associated precision information, combining the multiple data sources, and including a full base description of all travel in London. This will be maintained and updated with new surveys.
- The system will support a ‘conventional’ view of data, but provide much more extensive capabilities, building on developments in databases, metadata, statistics and modelling.
- The document introduces the concept of synthesised data, which will form a significant part of the database’s contents. Synthetic estimates in the database will often be as important as observed data in obtaining a picture of travel in London.
- The user interface must be flexible and scaleable to accommodate ease of use for discovery and display of information for the browser user, but with full search, manipulation and extract facilities for the more advanced.
- Links to other systems will be important, with exchange of information (and perhaps functionality) in both directions. Tight links with the TfL Planning and Information Database (PID) will be particularly important.
- The software approach is likely to involve a hybrid solution, not just a standard database tool.
- It is unlikely that the range of skills required to build, populate and operate the system will be found in a single organisation.
- There will be substantial continuing work to maintain and develop the information content of the system.

1.3 Background

This report builds on previous studies for the LATS project.

Prior to the foundation of TfL, a team of specialists was commissioned by LT to review the likely travel demand data needs for the new organisation. This work, which involved consultation with many senior staff who were scheduled to join TfL, reported in July 2000. Among the findings and recommendations, those that are relevant to this note are:

- The needs for information in TfL are wide-ranging, from operational and demand data to details of patterns of travel by all modes and information to support planning and forecasting techniques. A particular emphasis was placed on attitudinal data, which was felt to be inadequate.
- The users also vary widely. While most data has been used by transport specialists who could manipulate complex data structures, there is seen to be a growing need for accessible and friendly information that can be used by those requiring rapid answers to varied questions on all aspects of travel in London.
- There is also an important data user community outside TfL, including Boroughs, consultants, universities, journalists, etc.
- There is a need for base-line travel demand data and a consistent means of updating this data on a regular basis. It was recognised that sample survey observations alone could never provide a com-

plete, consistent and reliable source of the information required. The report concluded that synthetic estimation methods were needed to harmonise surveys and to complete the picture where partial surveys naturally left gaps. Such methods would give a complete and consistent best estimate of the current demand pattern at the required detail, and also form the basis of forecasts.

- The Data Needs report recommended the establishment of an integrated database system. This would be designed to ensure that data is handled efficiently and correctly, that information needs are met in a timely manner and that a high level of access to data is available.
- It was also recommended that TfL should establish an information unit to manage the programme, building on existing resources and skills.

LATS 2001 is the fourth such major decennial transport survey in London. The planning for LATS has sought to learn the lessons from previous surveys, notably that of LATS 1991, primarily with respect to ensuring that full value is obtained from the significant investment required to collect and collate the travel data. The planning for a database system for LATS 2001 has aimed to increase value through several means:

- Removing issues of (inevitable) inconsistencies in the observed data through statistical methods;
- Providing a methodology that allows the data to be meaningfully and regularly updated in a straightforward manner, again through the application of innovative statistical estimation methods;
- Developing a database system that provides wide accessibility of the information, both institutionally and for users of differing interests and skill levels;

1.4 Consultation

Earlier stages of this study have involved discussions with potential internal and external users and with suppliers of related software. Comments from the users have been incorporated into the User Needs listed in the next section, and a more explicit listing, gleaned from the minutes of those discussions, is stored in the Requirements database constructed during this study. Some comments from software suppliers appear as appendices.

2 User Needs and Design Objectives

2.1 Overview

The LATS database system is intended to be a dynamic resource containing information about the use of transport in London. It will contain information about demand, use and attitudes and will cover all modes of transport. It is complementary to various other databases about transport facilities in London, and will have facilities to cooperate with them.

The system is intended to be a developing archive of data, results, processes and conclusions about travel and transport use in London. It is seen as a resource for all those interested in information about transport in London, whatever the basis of that interest or the level of technical expertise.

It is proposed to produce a system that will provide functionality for people to use, explore and analyse information about transport. The database will be the core of the system, but the user interface will include suitable exploratory tools. It will be possible for authorised users to extract data from the system for further analysis in other software, but there will also be facilities for users to perform basic analyses and exploration directly within the system.

Although we talk about the system as a single entity, it is more likely that it will be a hybrid system, built from a combination of tools and technologies, and using more than one data store. However, the primary interface exposed to users will unify these different components.

It will be implemented in stages, so that information can be made available as soon as it is ready. It will be created initially using information collected during the 2001 round of the LATS project, but will be updated in subsequent years as further, similar and related information becomes available. In particular, this will tie in with the evolution of LATS into a continuous process. It will also be possible to add older data (such as information from the 1991 LATS) and data from other sources, where this contributes to the objectives of the system.

Management of the information resource, and the operational aspects of providing the information service, are further significant issues that have to be addressed in these proposals.

2.2 Management of Requirements

This report concentrates on four groups of requirement for the database system.

- **User Needs**: things that users need to be able to do, expressed in terms that apply to the use of the system.
- **Design Components**: components of functionality that need to be provided within the system, to produce a complete and rounded set of facilities that meet the user needs.
- **Solution Components**: specific facilities that need to be implemented to construct the system to achieve the design.
- **Implementation** Phases: groupings of solution components that give a possible implementation sequence, recognising dependencies between components.

The various items identified under these groupings are listed in the following sections (with considerable detail over the Design Components) and showing the main links between them.

The items have also been entered into a Requirements database (implemented in MS Access), and various other classes of information or requirement from this report have also been entered there, together with many of the links between the different items. A full listing of all the requirements and links from this database is available as a .PDF file, but the database itself can also be used as a dynamic resource for reviewing the items and links, and it can be used to support subsequent stages of the development.

2.3 Users and their Needs

We identify three main groups of potential users of the database (though real users may overlap these groups).

1. Transport Specialists, with special understanding and skills related to transport information
2. Policy makers and others responsible for decisions that relate to or depend on information about transport in London
3. Others with a general interest in summary information about transport in London (not service information), including the general public, researchers and journalists.

These groups overlap in their needs, but we can identify various broad classes of use and their importance to each group. The following table lists user requirements and activities that have been identified through consultation and by investigating other systems, projects and proposals, together with their likely importance to different classes of user. The final column of the table links to the components of the design (in the next section) that provide these facilities.

Table 1 User Needs

User Needs	**Importance** (1 – 3, 0 = not provided)			**Links to Design Components**
	Specialists	**Policy**	**Others**	
Discovery				
UNR:1 Discover resources using general terminology and common names	1	3	3	DCR:13, DCR:2
UNR:2 Discover resources using technical terminology and specialised references	3	1	1	DCR:13, DCR:2
UNR:3 Discover resources by following links from one item of information to another	2	3	3	DCR:13, DCR:2, DCR:3
UNR:4 Review definitions and other background information related to figures	3	1	2	DCR:13, DCR:2
Display				
UNR:5 View summary information as figures, and focus in on details of interest	2	1	3	DCR:4
UNR:6 View summary information as charts and diagrams	1	3	2	DCR:4
UNR:7 Read analyses and conclusions based on the data, produced by others	2	3	2	DCR:4
UNR:8 Integrated interface to different types of information	1	3	3	DCR:4, DCR:8
UNR:9 Presentation of information personalised to experience and preferences of user	1	3	3	DCR:4, DCR:8
UNR:10 Choose the level of interaction with the complexity of the system, from standard or base views and extracts, down to the full internal structure.	3	1	2	DCR:4, DCR:8
UNR:11 View the precision or confidence associated with any displayed information	2	2	2	DCR:4, DCR:5
UNR:12 Set bookmarks, to ease return to information that has been found	2	3	3	DCR:13, DCR:3, DCR:4, DCR:8
Manipulation				
UNR:13 Manipulate aggregate information to change focus or level of detail	2	1	3	DCR:5
UNR:14 Derive new summary measures based on ones already available	3	1	2	DCR:5
UNR:15 Build new aggregate summaries from micro data in the database	2	0	0	DCR:5
UNR:16 Browse through individual data records	?	0	0	DCR:4, DCR:5

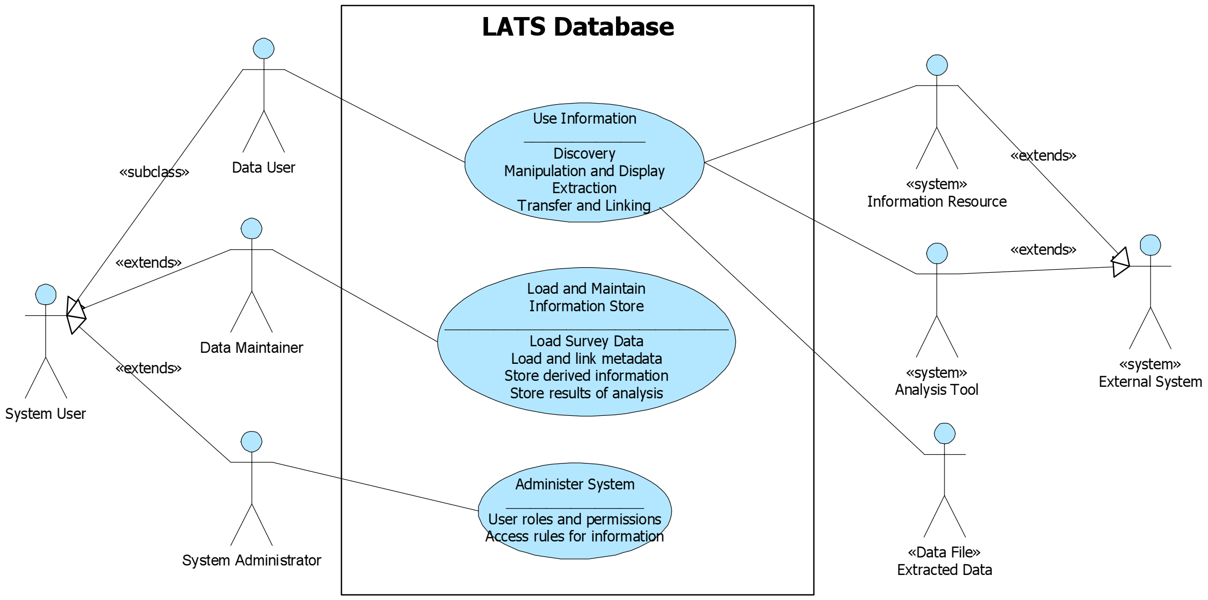
User Needs		Importance (1 – 3, 0 = not provided)			Links to Design Components
		Specialists	Policy	Others	
UNR:17	Store new results (views and derivations) for future use	3	2	3	DCR:4, DCR:5, DCR:7, DCR:8, DCR:9
Session Management					
UNR:18	Break off in the middle of exploration or derivation and return to the same place at the next session	2	2	3	DCR:4, DCR:6, DCR:7, DCR:8
UNR:19	Review the steps taken to reach a particular selection or view or arrangement of information	2	1	3	DCR:13, DCR:5, DCR:6
UNR:20	Rerun the steps taken to reach information, when any of the input components have been changed	3	1	2	DCR:13, DCR:5, DCR:8, DCR:9
UNR:21	Where source information is updated, be able to refer explicitly to particular older versions (to maintain consistency with extracted information)	3	1	2	DCR:4, DCR:5, DCR:6, DCR:9
System Links					
UNR:22	Access information within the database from other related analysis software and systems (eg GIS facilities in PID)	3	1	2	DCR:11
UNR:23	Access information from other related systems from within the database system (eg PID, RODS, Super BODS)	3	1	2	DCR:11
UNR:24	Transfer information out of the system in a suitable format for use in other related systems	3	1	2	DCR:10
Modelling					
UNR:25	Support for modelling (specification, fitting, synthesis and exploration)	3	1	1	DCR:12

There is one additional group of people involved with the system, those who maintain and operate the system and keep its content and performance up to date. Their needs are dealt with in the detail sections.

2.4 Design Objectives

To meet these needs we believe we need various components in the design (listed in Table 2). The intended functionality of the system is summarised in the following diagram.

Figure 1 Internal Functionality and External Connections



This is an example of a standard form of diagram used in systems design, called a 'Use Case' diagram (where 'use' is a noun). The 'Actors' (the stick figures) represent roles, systems or objects that are external to the system being designed (the LATS Database) but which interact with it. The ovals are Use Cases, which identify functionality within the system.

The first group of components in the following table reflects the user needs directly (mostly the first oval in the figure), and further components are directed at the construction, architecture, operation and management of the system, including loading information and building up the resource of conclusions and analyses. The final column shows links to the User Needs that motivate the component, and to other components that are in some way related to each one. Each component is elaborated in the next chapter.

Table 2 Design Components

Reference	Name	Description	Links
DCR:1	**Content**	The system will contain information from the original source surveys, aggregated and summarised information, synthetic information produced by modelling (including standard base matrices), and comment, analysis and conclusions based on this information. The content will develop dynamically as more information is collected in the future and more analysis is performed. All this will be supported by extensive metadata, to describe, inform and support use of the content.	DCR:12, DCR:13, DCR:18, DCR:7
DCR:2	**Discovery**	Different tools will be aimed at different groups of users. Catalogues using standard terminology will be aimed at subject specialists, a thesaurus component using common terms and alternatives will support non-specialists, and a general free text search facility over the descriptions, labels and analysis content should be useful to all.	DCR:13, DCR:3, UNR:1, UNR:2, UNR:3, UNR:4

Reference	Name	Description	Links
DCR:3	**Bookmarks and Links**	A general bookmark facility will allow users to remember where useful information is found, and will also support the construction of subject-specific indexes and catalogues aimed at particular groups of user. This should be essentially the same mechanism as is used for links between metadata and the referenced objects in the database.	DCR:18, DCR:2, UNR:12, UNR:3
DCR:4	**Presentation**	All information can be displayed in suitable forms, including numeric tables and charts. Different forms or levels of detail can be used (based on the idea of views) with default choices related to user groups and preferences. Basic displays should be possible using a standard web browser, with more advanced functionality requiring plug-ins or specialised client software.	DCR:13, UNR:10, UNR:11, UNR:12, UNR:16, UNR:17, UNR:18, UNR:21, UNR:5, UNR:6, UNR:7, UNR:8, UNR:9
DCR:5	**Manipulation**	Standard facilities for basic statistical manipulation of micro and aggregate information, including synthesised results.	DCR:13, UNR:11, UNR:13, UNR:14, UNR:15, UNR:16, UNR:17, UNR:19, UNR:20, UNR:21
DCR:6	**History**	A general history (or audit trail) facility will keep track of operations performed by users, allowing them to be reviewed or re-run.	DCR:13, DCR:18, DCR:8, DCR:9, UNR:18, UNR:19, UNR:21
DCR:7	**User and Access Management**	Named users will need to be registered and assigned to different skill or requirement groups. Access to information sets or functionality can be controlled at the level of groups or individual users. Issues of security, data integrity and confidentiality will need to be considered.	DCR:1, DCR:19, DCR:8, UNR:17, UNR:18
DCR:8	**User Interface and Resources**	Users should experience default settings based on membership of user groups, but be able to set their own preferences for various components of the interface. Private user storage is needed for bookmarks, histories, versions of summaries and analyses, and for returning to incomplete investigations. Ideally we need a workbench approach to the interface.	DCR:6, DCR:7, DCR:9, UNR:10, UNR:12, UNR:17, UNR:18, UNR:20, UNR:8, UNR:9
DCR:9	**Version Control**	As a dynamic resource, information in the database will change, and it is important to know which version was used for particular conclusions or derivations, and to be able to revisit earlier versions. This will operate both at a system-wide level, and for individual users.	DCR:13, DCR:6, DCR:8, UNR:17, UNR:20, UNR:21
DCR:10	**Export Facility**	There will be an export facility, allowing information to be placed into files in a suitable format for use in other systems.	DCR:13, DCR:14, DCR:17, UNR:24
DCR:11	**Links to and from other systems**	Experienced users should be able to make use of information from the system in external analysis systems, and to use information from external systems within this one. There are clearly standardisation issues associated with this requirement. The initial target for linking will be the TfL Planning and Information Database (PID) currently being implemented.	DCR:13, DCR:14, UNR:22, UNR:23

Reference	Name	Description	Links
DCR:12	**Modelling**	We intend the system to be a major resource for transport modelling, and the details of this are in a separate report. Facilities in the system will need to support specification, storage and fitting (parameter estimation) for models, and the subsequent generation of synthetic information.	DCR:1, DCR:13, UNR:25
DCR:13	**Metadata**	Metadata will be handled as an explicit resource within the database, taking a very general view that extends all the way from coding lists and labels for source data (codebook and data dictionary ideas), through operational aspects of derivation and manipulation (related to the history concept), to abstract concepts that relate to the subject matter of the various resources (related to the thesaurus concept).	DCR:1, DCR:10, DCR:11, DCR:12, DCR:2, DCR:4, DCR:5, DCR:6, DCR:9, UNR:1, UNR:12, UNR:19, UNR:2, UNR:20, UNR:3, UNR:4
DCR:14	**Standards**	As far as possible the system should make use of standard structures and protocols. There is much activity in the area of standards for statistical and geographical structures and metadata at present, with some standards emerging and others being discussed, so the development process will need to contribute to and learn from these activities.	DCR:10, DCR:11
DCR:15	**Hybrid Architecture**	Various software technologies exist that address some of the objectives for this system, but no single one addresses them all. We thus expect that the implementation of this system will involve a combination of component technologies, rather than a monolithic whole.	
DCR:16	**Dynamic Functionality**	The content of the system will evolve over time, so the architecture must support this dynamic. But the functionality will also develop, so the architecture must also support the extension and modification of functionality over time. In the shorter term this will apply to the staged construction of the initial system, so that some functionality is available early, with other features added later. However, change will extend to the longer term, as new ideas and methods for handling transport and statistical information are developed.	DCR:19
DCR:17	**System Integrity and Quality Assurance**	Suitable backup and disaster recovery procedures will be needed. The integrity of information within the system will be covered by access control, the history mechanism, plus procedures to control the introduction and updating of source information. However, there may be a need to ensure and protect the integrity of material extracted from the system (such as data files or documents containing analyses and conclusions).	DCR:10, DCR:18, DCR:19

Refer-ence	Name	Description	Links
DCR:18	**Information Loading and Maintenance**	Loading information, both initially and ongoing (as new, revised or updated information arrives) will be a significant task. Each block of information will require the instantiation of suitable storage objects, and the construction (or importing) of metadata to make the information accessible within the system. Basic analysis (either new or standardised) of all new information will be needed to construct the summaries available to non-expert users. Once in the system, information needs maintenance to ensure that links and references between items are kept valid, and that new understanding is applied to old information. This will affect information used for classification and searching, as well as conclusions and analyses.	DCR:1, DCR:17, DCR:19, DCR:3, DCR:6
DCR:19	**Operation and Management**	Ongoing tasks include: managing users, groups and resource permissions, reviewing the database content (particularly summaries and commentary) for timeliness, correctness and relevance, promoting the use of the system to potential users outside TfL, establishing and supporting links to and from other resources, extending the usefulness of the resource by incorporating related information (particularly commentary and analyses), as well as managing the loading and updating of data sources. This will be a substantial and continuing volume of work, and supports the recommendation from the Data Needs review that an in-house Information Unit should manage the development and application of the database, with appropriate support from contractors.	DCR:16, DCR:17, DCR:18, DCR:7

These ideas are elaborated in the following chapters, and some of the underlying concepts are explained in more detail later.

3 Elaboration of the Design Components

This chapter takes each Design Component, as listed in Table 2, and explores its characteristics in greater detail.

3.1 DCR:1-Content

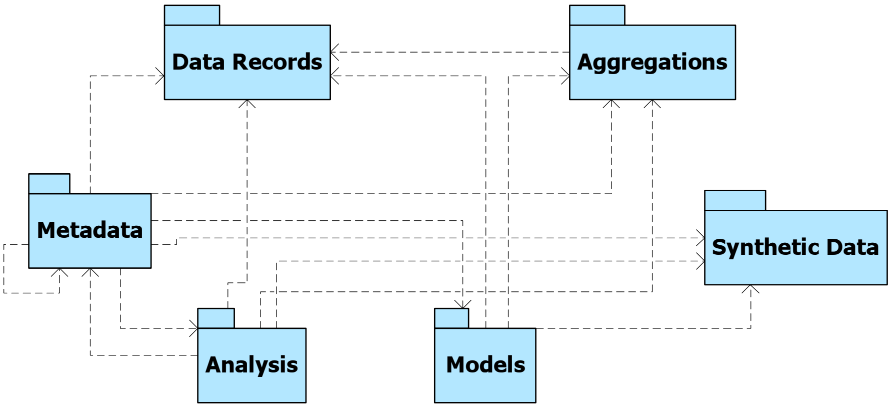
> *The system will contain information from the original source surveys, aggregated and summarised information, synthetic information produced by modelling (including standard base matrices), and comment, analysis and conclusions based on this information. The content will develop dynamically as more information is collected in the future and more analysis is performed. All this will be supported by extensive metadata, to describe, inform and support use of the content.*

The system will hold a variety of different forms of information of potential interest to users.

- It will contain both individual data records and aggregated data (with suitable facilities to prevent disclosure of confidential information, if needed), and will be flexible enough to hold other, more complex, data structures as needed. Potential sources of information are listed in section 4.1. There should be sufficient detail to be able to get useful answers to questions at least down to the level of major flows within (as well as between) boroughs (precision will be lower for smaller areas).
- It will be able to hold synthetic data produced by various modelling processes, and will allow different versions of estimates, produced at different times or under different assumptions, to be stored and retrieved. This will include approved base (or standard) matrices containing the 'best' estimates of the overall transport pattern, and even the possibility of synthetic populations. Issues relating to synthetic information are discussed in section 4.5.
- It will include analyses, results and conclusions, expressed in both numerical and diagram form. These will be supported by textual commentary and descriptions, and linked to sources within the system.
- The database will contain structured descriptions of the contents of the data resource in physical, operational and conceptual terms (usually called metadata), linked to the information described. This will support various uses by people and software, extending from searching for information about particular subjects to the export of information to other processing systems.
- The system will hold other useful textual material, including information about the content of the database, or the processing of the data into its stored form, or guidance on the interpretation of any information. This is actually a special case of the metadata, though it consists entirely of unstructured text, so will participate in the standard metadata linking and searching functionality.
- Depending on the extent to which modelling is supported by the system, it may be possible to hold model specifications as an explicit part of the system.

These different types of information are summarised in the following diagram.

Figure 2 Types of Information in the LATS Database



The arrows in the diagram show some of the links and dependencies between the different types of information. For example, the metadata component contains information that refers to all the other types of information (including links between different metadata elements).

3.2 DCR:2-Discovery

Different tools will be aimed at different groups of users. Catalogues using standard terminology will be aimed at subject specialists, a thesaurus component using common terms and alternatives will support non-specialists, and a general free text search facility over the descriptions, labels and analysis content should be useful to all.

The database will contain a catalogue component that lists the various resources using standard terminology and classifications. This is intended to provide rapid access to users who are familiar with the terminology.

There may be other catalogues (or similar structured lists) that are directed at particular groups of users and present a more restricted view of the system contents, but more highly structured in terms of subject. Users will be able to construct their own lists (favourites) using the Bookmark facility.

There will also be a general search facility, designed to support users who are less familiar or sure about their requirements. This will take at least one of the following forms:

- Free-text search over (all or selected) resource descriptions in the database.
- Thesaurus-based search.
  With this, the user's search string is examined for terms or concepts (or their synonyms) that have been extracted into the thesaurus. Resources are classified using the thesaurus concepts, and so can be found from the search string. This approach is less dependent on the user using standard terminology, but more dependent on the thesaurus being comprehensive and the resources being well classified.

3.3 DCR:3-Bookmarks and Links

A general bookmark facility will allow users to remember where useful information is found, and will also support the construction of subject-specific indexes and catalogues aimed at particular groups of user.

This should be essentially the same mechanism as is used for links between metadata and the referenced objects in the database.

It is essential that resources within the system are linked, for discovery through metadata (so that, for example, it is possible to move from reading the description of something to looking at the thing described), and also for statistical information (with links for aggregate data from the dimensions of a data cube (see 6.5) to the variables in the source data that were aggregated, and to the classifications used to define the possible groupings for each dimension). This can be thought of as similar to the concept of hyperlinks on the Web.

A likely solution for this is to treat all resources in the database as objects that have identity and so can be referenced – the standard approach in object-oriented systems. This would provide:

- Links between related resources within the system.
- Catalogues as lists of links
- User bookmarks, organised as 'favourites' or as personal (or shared) lists.

The construction and maintenance of the links between resources is a significant task.

3.4 DCR:4-Presentation

> *All information can be displayed in suitable forms, including numeric tables and charts. Different forms or levels of detail can be used (based on the idea of views) with default choices related to user groups and preferences. Basic displays should be possible using a standard web browser, with more advanced functionality requiring plug-ins or specialised client software.*

Using the standard interface (probably based on a web browser) the user will be able to:

- View and explore summary information, both aggregated from particular data sources and synthesised across multiple sources, with facilities to reduce or expand detail and coverage, in terms of (at least) area, time period, mode and passenger classifications.
- View information as charts or graphs, as required.
- Review reports, analyses and conclusions based on the data, in the form of charts, spreadsheets or other displays, accompanied by analytical commentary on the results.
- Explore the definitions and sources of data, classifications, processes and adjustments (via the metadata).
- Make personal choices about various aspects of the style and form of presentation.
- All numeric information can be accompanied by information about its precision or confidence.
- View all types of information through an integrated interface. The form of presentation will be different for different types of item, but the style will flow from the type of item selected, rather than the user having to choose a particular type of item before making a selection.
- The choices of information available to users will be affected by the groups to which users are assigned and by their personal preferences. Various 'views' of the same information will be available, designed to meet the needs of different groups of users. Where several views are available for an item, the default view for a user can depend on the preset characteristics of the user and any expressed preferences.
- Bookmarks can be stored for presentations, including form and style as well as content.

This basic access should be possible through a standard web browser. More advanced functionality may require specialised plug-ins for a browser, or separate client software. The precise location of the boundary between server-side functionality accessible through a standard browser, and client-side functionality for speed and flexibility, is an issue for the later specification stage, or as an early component of the development.

The concept of 'views' will be used to present information reorganised in standard ways, even if this is not the form ultimately used to store the information. Sophisticated users can have access to the full complexity of the system if they require it. In general, users will be able to choose the level of interaction with the complexity of the system, from standard or base views and extracts, down to the full internal structure. The concept of alternative views applies at multiple levels, for example, simple numeric information can be presented as figures or as a diagram, and other summary information could be presented with different levels of detail, such as including precision information, or increasing the amount of descriptive labelling included.

We do not intend to include facilities for general users to browse individual data records. It may be necessary to have disclosure control mechanisms to prevent identification of individuals through aggregate information, but it is not yet clear whether any of the information in the system will be sensitive.

3.5 DCR:5-Manipulation

> *Standard facilities for basic statistical manipulation of micro and aggregate information, including synthesised results.*

Some forms of manipulation are likely to need client-side functionality in the form of plug-in components for a browser, or more specialised software.

- Manipulate aggregate information (real and synthesised) including deriving new variables, changing the level of detail in classifications and focussing in on subsets.
- Users with appropriate permissions will be able to define new aggregations based on micro data.
- The system should automatically derive precision information for the results of the manipulations (or at least ensure that the precision can be evaluated if requested).
- The results of manipulating information can be stored (as can the steps, see 3.6), and bookmarked.
- It would be desirable to include functionality to help users avoid foolish manipulations, such as adding counts that are not disjoint, or averaging classification codes, but this is recognised as a difficult problem.

We are not treating mapping or statistical analysis facilities as high priority for the standard interface, since many programs that will be able to use information from the system do provide such facilities, often with considerable depth. However, some of the software systems that can be considered as candidates for implementing parts of the system do already provide some such capability, and we will take advantage of this. It will certainly be important to identify how these important functions can be provided for users.

3.6 DCR:6-History

> *A general history (or audit trail) facility will keep track of operations performed by users, allowing them to be reviewed or re-run.*

The system will need a general mechanism for keeping track of the processes that have been applied to reach particular arrangements or presentations of information. There are various reasons for this.

- To allow users to explore how information has been processed by others to reach a particular form. This will provide checks on sources, filters, adjustments, or any other aspect that affects the meaning and interpretation of the presented information.
- To allow users to review their own steps in reaching a particular presentation of information. This will support both confirmation of the steps used, and stepping back when a particular path is revealed as not useful.
- To allow processes to be rerun, perhaps because the source data has been updated, or because some aspect of the process is to be revised.

This mechanism will automatically record the activities undertaken within the system. This can have various uses, but in particular it will make it possible for a user to store the sequence of steps used to reach a particular presentation or derivation of information. This will allow the steps to be repeated (perhaps if the underlying information changes), or for the steps to be reviewed by another user.

Note that use of this mechanism will extend to modelling and the production of synthetic information for the database.

- Whenever information is stored in the system, the history of the steps used to reach the information is stored (as a history object) and the information is linked to the history. This provides the review and rerun functionalities mentioned above. Note that a simple textual log of steps is not sufficient, as it is essential that the history contain all the details needed to repeat the steps.

Some information from a process will become properties (metadata) of objects created by the process, so will not need to be in the history. For example, where a new measure is derived for a summary, the specification of the measure becomes part of its metadata, so may not need to be recorded in the history.

- All points within a history can be bookmarked (by default the end point will be chosen) and going to a bookmark on a history will execute the steps needed to get to that point (this is the rerun functionality). Note that when the history relates to the production of information that has been saved, it will usually be more sensible to bookmark the stored information.
- All active users will have an active history of their current steps, which will be automatically saved when they log out (or the connection is lost), with a default bookmark established to return them to that point when they restart.
- History recording should be switched on by default, but users with sufficient authority should be able to switch it off temporarily, for example when performing some form of large-scale update to the system.
- The history mechanism will be dependent on the version control facility (see 3.9), in that where versions exist the history must know which one was used, and whether the reference is to an absolute one (identified, say, by date) or a relative one (latest, previous, first, etc.).
- The development process will need to give careful consideration to the technical issues of scope and garbage collection related to histories – under what circumstances is a history no longer in use or usable?

3.7 DCR:7-User and Access Management

> *Named users will need to be registered and assigned to different skill or requirement groups. Access to information sets or functionality can be controlled at the level of groups or individual users. Issues of security, data integrity and confidentiality will need to be considered.*

As part of the operation and management of the system (see 3.19), all users of the system will need to use a registered ID and password. Registered users will be given personal resources and access permissions. Similarly, access restrictions (and perhaps charges) will need to be allocated to information resources. This mechanism will probably focus on Groups, to which users can be given membership, and against which access rights are allocated. A useful model and implementation of an Access Rights system for data archive resources is being developed as part of the Faster[2] project. Particular groups can represent collections of information that share the same access restrictions, or they can represent collections of users that share the same group of permissions (as a ‘Role’).

- User identification is necessary in order to allocate resources to users (see 3.8) for storing user information and preferences.
- Different information resources are likely to have different usage permissions, so facilities are needed to allocate these to resources and relate them to users (or user groups). The context in which resources

[2] www.faster-data.org

are referenced can also be important – for example, end users may not have permission to read micro data, but they can rerun a summary based on that micro data.

- Access control may extend to functionality, as well as information.
- Other issues, such as disclosure control, may need to be linked to user status (in that it is less critical for internal users than for external ones, for example).
- It can be useful to have an anonymous (or Guest) account with limited permissions (and no resources) for initial visitors to the system.
- Functionality to record the usage of the system will be important, and should be related to users, particular information and (perhaps) context. This would also provide the basis for charging, if that were required.

3.8 DCR:8-User Interface and Resources

Users should experience default settings based on membership of user groups, but be able to set their own preferences for various components of the interface. Private user storage is needed for bookmarks, histories, versions of summaries and analyses, and for returning to incomplete investigations. Ideally we need a workbench approach to the interface.

The facilities of the user interface will be designed to support various types of user and various types of inquiry (with suitable access control mechanisms).

With identified users, we hope to implement user facilities through a ‘workbench’ paradigm, where the user can organise the content and form of their working environment within the system, and store it for subsequent use. Users will have storage resources within the system that can be used with various related facilities to assist users in managing their use of the information that is presented.

- Make personal choices about various aspects of the style and form of presentation, stored in the form of a user Profile.
- Store bookmarks to particular presentations (or other resources), including form and style as well as content. Some basic form of management, as in a Favourites list, would be needed.
- The system should automatically maintain a ‘Recently Used’ list, of resources visited. Automatic pruning of the list should be related to size and frequency of use, not just age.
- Store new results (new views or derived information) and any associated commentary or conclusions. The version control mechanism (see 3.9) will apply to user objects.
- Keep track of how information was discovered or manipulated through history elements in personal storage.
- Share information with colleagues. This could be done by allowing users permission to assign rights to access their own stored information to selected groups of users.
- Store the current context for each user (as a history) and retain it at logoff or loss of connection, so that a user can restart from that point, and perform tasks that extend over more than one session.

Usability (ease of use) of the interface is an important issue, and must be specifically addressed at the detailed design stage. For the moment we just mention some facets that will need to be considered.

- Different users, with different needs and skills (probably identified through a variety of user ‘Roles’) need different functionality. For example, experienced users often know exactly what they want from the system, whereas more casual users want to discover what information is available on particular topics.
- Users with lower levels of skill need less detail in functionality for manipulation of information, but more in-depth work on content and presentation, so that interesting information can be found easily.

This has implications not only for the design of the user interface, but also for the maintenance and presentation of suitable content.

- The primary interface to the system will need to be flexible, with various routes for further progression into the resources, appropriate for different types of user.
- The initial style of the interface should be selected on the basis of user roles, but users will have means to vary the content of their personal interface, in terms of style, form and content. The user's personal interface need not be the first screen displayed after logging in, but should be immediately accessible from the first one.
- It is intended that the resource will be accessible both internally over TfL networks and externally over the Internet to approved users.

The system will be complex, and the need to protect some users from this complexity is recognised. Procedures for doing this within the user interface have been studied (under the heading of User-Centred Design) and are now fairly well understood. Within the database structure, the concept of 'views' will be used to present information reorganised in standard ways, even if this is not the form ultimately used to store the information. Sophisticated users can have access to the full complexity of the system.

3.9 DCR:9-Version Control

As a dynamic resource, information in the database will change, and it is important to know which version was used for particular conclusions or derivations, and to be able to revisit earlier versions. This will operate both at a system-wide level, and for individual users.

In a dynamic database, where resources get updated and change is of interest, it is essential that change is explicitly recognised, through *version control* facilities.

In some situations we can build this into the underlying data structures. For example, with annual surveys, we can build separate datasets for each year, or we can have a combined dataset with the year explicitly included in each record.

However, in general this is not an adequate approach, particularly where change is more continuous, and where analysis and derivations can be performed on the resource at will. Then we need to build in version control at a more generic and procedural level (rather than relying on the structure of individual resources). This can have several components:

- On a regular basis a version image (snapshot) of the system is taken (probably on the basis of differences) and at any time in the future a user can refer back to the state of any element of the resource at any version. The timing and frequency of the versions will be determined by the resource administrators on the basis of the rate of change of the resource and on the occurrence of significant change events. Where an object is changed more than once between two versions, only the final state of the object will be retained, though the history of the changes might be saved, since the history could be an object in its own right (see 3.6).
- A user can at any time explicitly save a version of any object over which they have appropriate rights (which will always include any objects created in a personal work-space, subject to resource limits), and can refer back to it in the same way as system resource versions.
- The history mechanism will automatically include reference to the version of any resource used. This will enable positive identification of the version when reviewing any history. If a history is re-run, the user will be able to reuse the original versions, or the current ones, or any other ones.
- It may be necessary to recognise different types of version. For example, a version that differs from another by using a different version of the source information might be considered differently from a version based on the same source information but using different parameters in the processing.

3.10 DCR:10-Export Facility

> *There will be an export facility, allowing information to be placed into files in a suitable format for use in other systems.*

The system will support the transfer of data to other processing systems.

- Transfer can involve either the extraction of information into files, for physical movement to the other application, or linking (see 3.11), so that the other application can connect to the LATS system and request information as it needs it.
- Selection of subsets will be possible prior to transfer.
- Numeric information can be accompanied by suitable descriptive information (metadata), such as variable names and labels, value labels, etc, through to sampling information and derivation rules.
- Disclosure control mechanisms will need to be considered, linked to user access rights.

The transfer and use of information that can be represented in flat-file (or relational) form should be relatively straightforward. This will certainly include all individual data records and may cover much aggregate data as well. Other data structures may need the receiving software to have more understanding of the problem domain. Work on standards for statistical data (see 6.8), particularly those using XML, may facilitate such transfer.

3.11 DCR:11-Links to and from other systems

> *Experienced users should be able to make use of information from the system in external analysis systems, and to use information from external systems within this one. There are clearly standardisation issues associated with this requirement. The initial target for linking will be the TfL Planning and Information Database (PID) currently being implemented.*

It is clear that there are various other information resources that could be used in conjunction with the LATS system. Of particular importance is the TfL Planning and Information database (the PID, for Infrastructure, Service and Geographical data). Other examples of information that would also be of value include land use data and a proposed system for road traffic automatic monitoring data. In addition, there are other established databases available for particular modes of transport in London such as buses (BODS) and the Underground (RODS).

It is unlikely (though not yet decided) that the ability to physically transfer information to or from such systems (as described above) will be sufficient. Rather, additional, more dynamic, features will be needed. These should allow other resources to be used with the LATS information within the context of the LATS database, and other systems to dynamically access information from the LATS resource.

- As far as possible the system will link to other existing resources on related topics, such as GIS facilities and the TfL Planning and Information database (PID).
- Links will be supported in both directions, so that the database system can make use of information in other systems, and other systems can make use of information in the LATS database.
- Links will use standard mechanisms, such as ODBC or XML-based standards, not proprietary or special protocols.
- Access to basic information in the LATS database should be possible from any software that supports the chosen mechanism, subject to user authentication. This should include many standard statistical packages, GIS software, Office tools such as Excel and Access, as well as more specialised software. In particular, the ODBC protocol should enable other systems to access information stored in a relational form within LATS, which should in turn include all micro data and most aggregate information, whether original or synthesised.

- Access to more structured information will require more sophisticated linking mechanisms, particularly where metadata is to be included in the link. Various proposals, mostly based on XML, are being discussed in the statistical and geographical domains, so the development will need to draw on (and quite possibly contribute to) these initiatives.
- In the same way, it should be relatively straightforward for the LATS system to make use of information in relational form in other cooperating systems, but agreement will be needed for more complex structures.
- Information retrieved from other systems will not have been subject to the same quality processes as internal information. In particular, it may not be accompanied by the same extent or type of metadata, and it may use different versions of classifications or measures. The LATS system will need to handle these situations, and at least make the user aware of such differences.
- Disclosure control mechanisms will need to be considered, linked to user access rights.
- The use of resources through linking is seen as a specialist activity, not one that is directly supported for casual end users. Thus, for example, it may be necessary for users to prepare information into appropriate views or forms within the database before it can usefully be accessed from other systems. If a suitable component architecture can be developed it might be possible to invoke these components of the LATS system from within other systems, but such functionality is not included in our early objectives for the system.

Information transfer is not itself particularly problematic, since various mechanisms are now well established, but the key to providing useful functionality is the effective exchange and use of metadata. This is a potentially complex area, with considerable depth. Effective interchange requires agreement on standards, and covers issues from exchange protocols through to structures and coding. The system should make use of suitable standards where they exist, but this may need involvement in a standard setting process, with suitable groups. The LATS system may establish a lead in its approach to the linking and exchange of information with other resources, since many other systems that are candidates for linking have been established for several years, and even the newer ones have not taken such a broad view of interchange possibilities. It is thus important that there are extensive technical discussions with other potential cooperating resources about objectives, means and standards.

3.12 DCR:12-Modelling

> *We intend the system to be a major resource for transport modelling, and the details of this are in a separate report. Facilities in the system will need to support specification, storage and fitting (parameter estimation) for models, and the subsequent generation of synthetic information.*

Synthetic information is fundamental to our view of the database as a general resource, with an important component being standard, validated, updateable Base matrices. These will be targeted for use by modellers and others, but, in turn, we wish to support the modelling process by which such synthetic information (base matrices and others) will be produced.

The implementation of the data synthesis (modelling) methodology is a significant undertaking that will be the subject of a separate contract. The facilities of the LATS database will provide the important data handling capabilities, including the ability to access and aggregate data in varied ways and to keep audit trails. This means that work on the synthesis component can focus on methodological and algorithmic matters, and so the marginal cost of the data synthesis development is accordingly considerably reduced. If the chosen solution for the database uses component architecture (see 6.9), then it should be possible to directly include the synthesis component as part of the main system. If not, it can operate using the linking facilities described in section 3.11.

Our objective in the database system is to provide facilities that support the methods and procedures recommended by the synthesis report (see section 4.5). At a basic level this requirement is largely covered by the storage, history and version control features already described, though there may be additional data structures (beyond micro and aggregate data) needed for synthesised results.

At a more ambitious level, we hope eventually to be able to provide more general facilities that will provide the means to both express and fit general models within the system. To do this the database needs various additional components.

- Structures to store the mathematical components of the model, in a form that can be executed (evaluated with suitable input values) and manipulated (for example, differentiated). MathML or CWM may provide useful support for this (see 3.14).
- The overall form of models will be complex, containing distributions for variables (with parameters), linking functions between variables (model formulae with parameters), dependencies between the distributions of variables (with parameters), and prior information about all the parameters (with the same degree of complexity in the relationships of the parameters and their distributional forms). It is most unlikely that simple, closed forms will be available for most complete models.
- Functionality (processes and algorithms) to fit a model to data in the system. This is likely to require computationally intensive iterative procedures using combinations of MCMC and EM (or similar) approaches (see section 4.5.5).
- A fitted model can be represented by the set of parameter estimates that specify its current state. Note that this can also be the starting point for further fits (cf Bookmarks and Histories).
- We need functionality to generate synthetic information from a fitted model, to make forecasts and to explore the impact of change on the model.

Realistically, it will not be possible to achieve all these more ambitious objectives in the short term, but some basic level of support for modelling is anticipated.

3.13 DCR:13-Metadata

> *Metadata will be handled as an explicit resource within the database, taking a very general view that extends all the way from coding lists and labels for source data (codebook and data dictionary ideas), through operational aspects of derivation and manipulation (related to the history concept), to abstract concepts that relate to the subject matter of the various resources (related to the thesaurus concept).*

The term *metadata* is used widely in the context of database and statistical systems, though there is not general agreement on exactly what it means. We take a very broad view:

> *Metadata is anything that you need to know to make proper and correct use of the real data, in terms of reading, processing, interpreting, analysing and presenting the information. Thus metadata includes file descriptions, codebooks, processing details, sample designs, fieldwork reports, conceptual motivations, etc., in other words, anything that might influence the way in which the core information is used.*

Metadata can be used informally by people who read it (and use it to affect the way they work with or interpret information), and formally by software to guide and control the way information is processed. Processes can also generate metadata.

Various attempts have been made to classify statistical metadata, none of them totally adequate. A useful, though crude, partition is into three groups:

- Physical metadata – what is in the data files, format, layout, coding, etc? This is what allows the system to present information or use it in processes without detailed specification by the users.
- Operational (or Process) metadata – how was the data obtained, how were variables derived, etc? This is largely the area covered by the history mechanism.

- Conceptual metadata – why was that particular data collected, why was a question worded in a particular way, what does something mean, how are things related? This is often seen as informal and unstructured, but in this system more formality will be available. The bookmark facility provides functional links between items; commentary and other descriptive material will be linked to the source information; and the catalogue and thesaurus components will provide structure.

Almost all parts of the system will be dependent on the metadata component, and further discussion appears in section 6.6

Because the metadata is central to the interchange of information with other systems it is essential to use suitable standards for organising the metadata.

3.14 DCR:14-Standards

> *As far as possible the system should make use of standard structures and protocols. There is much activity in the area of standards for statistical structures and metadata at present, with some standards emerging and others being discussed, so the development process will need to contribute to and learn from these activities.*

There have been efforts to agree on standards for statistical data and metadata structure and definition for some years, with considerable funding input from National Statistical Offices and Eurostat over the last decade. This is beginning to bear fruit, with some agreement on principles emerging. In addition, various groups have been proposing standards in particular specialised areas, an activity much motivated by agreement on the XML[3] (eXtended Markup Language) standard (see 6.8).

There is much to be gained by using XML to implement the description of data structures that are generally agreed. This is happening in varied industries. Important projects for survey data are:

- the Codebook proposal, from the Data Documentation Initiative[4] project based at the University of Michigan,
- the IMIM (Integrated Metadata Information Management) project for a metadata repository, initiated by Bo Sundgren from Statistics Sweden, and now being extended in the Bridge software from Run Software Werkstat[5], and
- the IQML[6] (Intelligent Questionnaire Markup Language) project for questionnaire design led by Edinburgh University.

A further project funded by Eurostat (MetaNet[7]) started at the beginning of 2001, with the objective of drawing together various activities and initiatives on statistical metadata and structures, and producing a more integrated overview of achievements and prospects.

The Common Warehouse Metamodel[8] (CWM), developed by the Object Management Group (OMG) covers many areas of relevance to LATS, including the manipulation of aggregate data (also referred to as OLAP), the representation of data transformations and expressions (in the Transformations component), and in the

[3] www.w3.org

[4] www.icpsr.umich.edu/DDI

[5] www.run-software.com

[6] www.epros.ed.ac.uk/iqml

[7] www.epros.ed.ac.uk/metanet

[8] www.omg.org/technology/cwm

handling of classification structures. It is possible that MathML[9] (one of the longer-established systems built using XML) may also be useful for representing expressions and models.

The ebXML initiative (for electronic business) is seen (by the European Central Bank and Eurostat) as offering new standards for data exchange, potentially replacing the GESMES component of the EdiFact system.

This list by no means exhausts the activities that can be relevant for the LATS database. Here we have focussed on the statistical side of metadata, but there will be similar initiatives on the transport and geography side to consider, such as the GML[10] (Geographical Markup Language) proposal from the OpenGIS[11] group. Other initiatives, in other specific or broad domains (such as the UK e-Government Interoperability Framework proposals[12]), address related issues, and so may be of value.

We do not expect that all the standards needed will be in place for building the database, so the approach will be incremental, using suitable standards where they exist, delaying implementation for less critical components, and making interim (but revisable) arrangements where implementation is needed at an early stage. LATS also provides an opportunity to contribute to the establishment of standards that have value to the transport modelling community beyond the immediate interests of LATS.

3.15 DCR:15-Hybrid Architecture

> *Various software and database technologies exist that address some of the objectives for this system, but no single one addresses them all. We thus expect that the implementation of this system will involve a combination of component technologies, rather than a monolithic whole.*

The database is intended to hold several different types of information, so it is probable that a hybrid solution will be needed, drawing on different types of standard software, with an additional layer providing integration between these components.

Figure 3 shows links between the different types of information proposed for the database and developing technologies that provide much of the required functionality, plus some examples of software systems that implement some of this functionality (not an exhaustive collection). It is clear that there is unlikely to be any single system that will provide a full range of functionality in the immediate future.

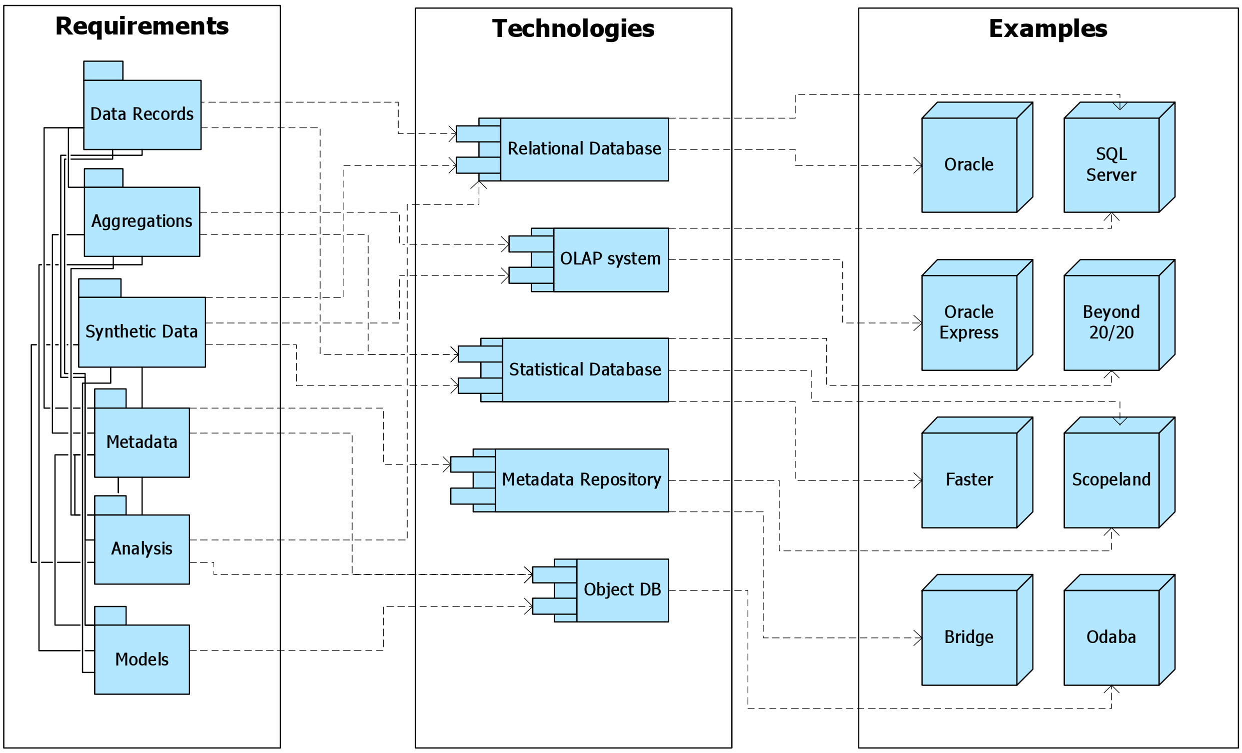
[9] www.w3.org/Math

[10] www.opengis.org/techno/specs/00-029/GML.html

[11] www.opengis.org

[12] UK e-Government Interoperability Framework - www.govtalk.gov.uk/egif/home.html

Figure 3 Requirements and Technologies



There are many similarities (and some differences) between the requirements of the LATS database and the systems of many Statistical Offices. Some have tried to build their own statistical database systems, or have made proposals for integrated metadata systems[13], but none have been completely successful. It is clearly beyond the resources of the LATS project to design and build a complete system from scratch. So the solution has to be one that draws on general-purpose tools built elsewhere for at least part of the system.

At a deeper level in the software architecture the idea of component objects is being widely promoted. This uses standardised interfaces based on COM or CORBA standards (from Microsoft and the Object Management Group, respectively), and is the basis of the ActiveX technology and JavaBean technologies used to add components to Windows and Internet applications. The most recent manifestation of this movement is the major .NET initiative from Microsoft that underlies the recent Office and Windows XP products.

The important idea is that once the interface is defined, the object can be replaced by a different version (that might provide additional functions for users) without the rest of the system being affected. If such an approach were considered suitable for the LATS database it could greatly ease the issues of incremental and dynamic development.

3.16 DCR:16-Dynamic Functionality

The content of the system will evolve over time, so the architecture must support this dynamic. But the functionality will also develop, so the architecture must also support the extension and modification of functionality over time. In the shorter term this will apply to the staged construction of the initial system, so that some functionality is available early, with other features added later. However, change will extend to the longer term, as new ideas and methods for handling transport and statistical information are developed.

[13] Jean-Pierre Kent, Jelke Bethlehem, Ad Willeboordse, Winfried Ypma (2000): On the Use of Metadata in Statistical Data Processing, Statistics Netherlands

The project is ambitious, and it is probably not possible to achieve all the desirable objectives, at least in the short term. However, one objective is that the system should have a long life, and develop over time, so this is consistent with an incremental approach to its design and facilities. It is important that information be made accessible through the database as soon as it is available from the data collection and processing parts of the LATS project, but it is less critical if some of the functionality for easy or extended use of the information is not available until later.

- The development and implementation plan must be incremental, so that users can start using the system before everything is finished. This is driven by available information resources, but also applies to functionality. Early stages must provide the basic functionality to use (view, manipulate and extract) the early resources in useful ways, but further functionality for these resources can come later.
- The later addition of functionality must not generate significant additional work related to existing information. For example, it would be unacceptable for the later introduction of the history mechanism to require that histories had to be generated by hand for all existing summaries.
- The underlying architecture of the system must support continuing development and change in terms of data structures and functionality (as well as information) as understanding of the area improves and new ideas and tools appear.
- The implementation must be based on a widely-available methodology that supports incremental development and implementation and that is capable of transfer between various parties involved in the development and maintenance of the system at various times.

3.17 DCR:17-System Integrity and Quality Assurance

Suitable backup and disaster recovery procedures will be needed. The integrity of information within the system will be covered by access control, the history mechanism, plus procedures to control the introduction and updating of source information. However, there may be a need to ensure and protect the integrity of material extracted from the system (such as files or documents containing analyses and conclusions).

It is important that users of the system can have confidence in the validity of the information in the database. The access control features should prevent the unauthorised alteration of information within the system, and the history mechanism should keep track of the processes applied to the figures presented.

It is important that the source information introduced into the system is well supported with documentation of the procedures used to select and collect it, and of the processing done prior to importing.

When information is taken away from the system (whether as exported files, or as results or conclusions quoted elsewhere), it is difficult to ensure that information is used or quoted correctly. What we can be sure of (through the history mechanism) is what was extracted or presented for the user. For this reason it might be decided that the history information from user manipulations should be retained in the system for longer rather than shorter periods.

- The construction process for the system must include testing processes that prove that the history mechanism produces information that correctly records (and can reproduce) the operations performed.
- Information should not be accepted into the system unless it is accompanied by adequate documentation about its provenance and quality.
- Ideally, information should only be accepted into the database if it meets appropriate standards for provenance, processing, coding, etc. In practice, information that does not meet the standards but for which the quality, etc, is known is more useful than no information at all.
- There are bound to be links between items of information in the system that cannot be captured by the history mechanism, so it is important that the operation of the system ensures that such links are iden-

tified and recorded (through the bookmark link facility). They must also be reviewed and revised from time to time, particularly when source information is updated.

As part of the specification stage for the project, consideration will also be given to any issues of Data Protection legislation that may relate to the system. In particular, it needs to be decided whether disclosure control procedures[14] are needed to prevent the linking of sensitive information to identifiable individuals. It may be argued that none of the information collected is sensitive, so no control is necessary, but a firm decision is required.

- If disclosure control mechanisms are needed (to prevent identification of the source of particular records in the database) these must be available both before data sets are loaded (providing static protection for complete datasets) and at any point when data are retrieved or analysed (giving dynamic protection for selected subsets and presentations).
- Different degrees of protection may be applicable for different classes of users. In particular, if unprotected (unaltered) data is present in the system it may be appropriate for internal administrators and specialist users to have unprotected access, but not for general users.

3.18 DCR:18-Information Loading and Maintenance

Loading information, both initially and ongoing (as new, revised or updated information arrives) will be a significant task. Each block of information will require the instantiation of suitable storage objects, and the construction (or importing) of metadata to make the information accessible within the system. Basic analysis (either new or standardised) of all new information will be needed to construct the summaries available to non-expert users.

Once in the system, information needs maintenance to ensure that links and references between items are kept valid, and that new understanding is applied to old information. This will affect information used for classification and searching, as well as conclusions and analyses.

Initialisation of the system will be a major task, involving loading data from the LATS surveys, constructing initial summaries from this information, defining the subjects and concepts that form the basis of the catalogue and thesaurus discovery systems, and then setting up all the metadata to describe and link the information together. Some of the work can be done by the development and implementation contractors, but LATS staff (probably assisted by consultants) will need to be responsible for much of the planning and conceptual development.

- The system requires facilities for loading information as an ongoing task. This will cover both new information and updates to existing information. Loading will include related metadata as well as numerical information, and suitable quality assurance procedures (see 3.17).
- When new information is loaded, additional metadata will be needed to link and integrate it into the system as a whole (such as adding it to catalogues, setting up links to related numeric and conceptual information). It is not likely that this process can be automated.
- The specification stage for the system should produce a detailed plan for the information that will be introduced into the system as a direct output from the LATS surveys. An outline list of data sources is available in section 4.1, but this will need to be elaborated to the level of the entities and fields to be included, supported by information about classifications and other metadata.
- In the outline implementation phases discussed later (see 5.3) the loading and manipulation of information is treated as a separate phase from the initial development, as they need different skills and so may use separate contractors. However, in order to meet the requirement for early availability of early

[14] Experience in Disclosure Control methods has been developed at ONS, and in the SDC project (www.cbs.nl/sdc), funded by Eurostat and hosted at Statistics Netherlands.

information, it will be essential that the initial development and loading phases overlap, requiring good cooperation between the two contractors. A draft plan for the availability of the data from the cleaning and weighting stage of the LATS surveys should be available shortly, but deviations in content and timing are to be expected.

- The specification stage will need to produce an outline tabulation (or manipulation and presentation) plan for the summaries to be derived from each tranche of source information after it is loaded into the system. This plan will need elaboration by LATS staff (with assistance from consultants) during the development process. The loading contractors will execute the tabulation plan (together with the construction of the metadata that is needed to incorporate the summaries into the system), following the loading of the related information.
- LATS staff (with consultants) will need to plan the subject matter catalogues and classifications to be used as the basis of the discovery system. The initial implementation of this information into the system can be done jointly between LATS and the contractors. The initial version should cover adequately the initial uses to be made of the (early releases of the) system, but depth can be added later as more users are added to the system, and in response to observed usage patterns and needs. The LATS team will need to work closely with the interface design specialists of the contractors for this component of the system.
- It is expected that a separate contract will be placed for the generation of Synthetic Information (see 4.5 for discussion of the methodology). This contract should produce a substantial body of information to be loaded and linked into the system, but that work should be covered by the contract itself.

The loading and elaboration process will take place in stages, as aspects of the system are implemented and deployed. The LATS team will gradually move into maintenance mode, as the proportion of the initial LATS data sources that are loaded and activated increases. Maintenance will involve the following tasks:

- Loading new versions of existing information, and other new information.
- Updating summaries, conclusions and metadata relating to updated (new versions of existing) information, and to new information.
- Adding depth to the subject matter classifications used for catalogue and thesaurus discovery.
- Loading up analytical material in the form of summaries, charts, graphs, discussions and commentaries. These will need to be classified, described and linked into the existing information – where figures and charts have been derived within the system much of this description and linking should be automatically generated. This implies a mechanism to expose analyses prepared by individuals to the wider community of users of the system.
- Modelling results and generated synthetic information should be covered in a similar way.
- Reviewing existing material (links, conclusions, etc.) for quality and correctness as new information (data sources, versions and analyses) are introduced into the system.

3.19 DCR:19-Operation and Management

Ongoing tasks include: managing users, groups and resource permissions, reviewing the database content (particularly summaries and commentary) for timeliness, correctness and relevance, promoting the use of the system to potential users outside TfL, establishing and supporting links to and from other resources, extending the usefulness of the resource by incorporating related information (particularly commentary and analyses), as well as managing the loading and updating of data sources.

This will be a substantial and continuing volume of work, and supports the recommendation from the Data Needs review that an in-house Information Unit should manage the development and application of the database, with appropriate support from contractors.

The operation of the system will involve various substantial tasks, which start during the implementation project and continue as long as the system is in use.

- Information Loading and Maintenance (see 3.18).
- User and Access Management (see 3.7).
- System Integrity and Quality Assurance (see 3.17).

It has been proposed that the LATS concept should be changed from a series of decennial but stand-alone projects to a continuous process in which updates take place between the large decennial surveys, perhaps with an intermediate survey every five years, and with continuous updating based on a variation of the LRTS surveys and other sources of transport information in London.

We expect that there will be developments in transport modelling and synthesis methodologies over the next few years, together with developments in the ideas, tools and technologies used for analysing transport data, and the LATS database system will need development to allow users of transport information to take advantage of these.

The prospect of these developments, plus the need for a specialist team to work on the maintenance and enhancement of the information resource in the LATS system, is consistent with the view that there is a need for a dedicated Information Unit in TfL to manage and oversee the operation and development of the system.

The team would be responsible for enhancing and exploiting the information content of the system, would need special understanding of the operation of the system as implemented, and would take responsibility for managing any additional developments to the system. It would thus need transport and statistical specialists as well as IT skills, and would draw on contractors for special tasks. The Unit could grow naturally out of the existing LATS project team.

4 Data Sources and Related Issues

The LATS database system is intended to be a dynamic resource containing information about travel in London. The objective is to construct a developing archive of data, results, processes and conclusions about transport use. This is seen as a resource for all those interested in information about transport in London, whatever the basis of that interest or the level of technical expertise.

The system will contain information from the original source surveys, aggregated and summarised information, synthetic information produced by modelling (including standard base matrices), and comment, analysis and conclusions based on this information (see 3.1). The content will develop dynamically as more information is collected in the future and more analysis is performed. All this will be supported by extensive metadata, to describe, inform and support use of the content.

In this section we discuss some additional issues related to the various types of content for the database.

4.1 Data Sources

The current situation with LATS is as follows:

- All major fieldwork is under contract and final minor surveys (e.g. taxis, coaches) are near to being commissioned.
- Data entry and geo-coding contracts are in place for the main surveys that were commissioned by LATS. Contractual decisions .on further processing and geo-coding of secondary data (particularly bus and underground surveys) have still to be taken.
- LATS is moving away from a ten-yearly cycle and toward a rolling programme of surveys, following the report of an expert team into the subject.

The likely data sources available for consideration for direct inclusion in the database include the following (details are subject to change as the surveys progress).

Table 3 LATS 2001 and Related ongoing Surveys

Data Survey	Comment	Content
LATS Household survey within the M25	Paper, Target of 20,000 households in 2001, smaller numbers in subsequent years	Household (Size, vehicle details, tenure, HH income) and individual (Demographics, work, transport, income) questions, Travel Diary - Recall yesterday, plus self-administered (SA) diary for one specified future day, details of trip stages). Non-response survey.
LATS Household survey in collar outside the M25	CAPI, Target of 5760 household from 24 clusters	Slimmer version of London HH survey, with trips but not stages.
LATS Roadside interviews	750 sites with collar, M25, central London and inner London cordons, connected by screen lines	Counts + Interview + SA diary
LATS Underground (RODS)	At stations, plus DLR	Station entrance, hand out, post back (>20% response)
LATS Overground Rail	At stations	Entry station (on train for intercity), SA. Origin/Destination, trip type, purpose, terminal modes: demographics, residence, work (~ 30% response). Plus counts.
LATS Buses (BODS)	On buses, part of rolling programme	Origin, Destination. Demographics, Ticket type, journey purpose, access, egress mode
LATS Taxi drivers	Spring 2002	
LATS Coaches	Spring 2002	

Table 3 LATS 2001 and Related ongoing Surveys

Data Survey	Comment	Content
Underground station entry counts	All stations, Annual, single day.	Full count per 15 mins, separate wheelchairs.
UTS Underground gate counts	On-going, by ticket type	Full entry and exit counts, per gate, per 15 mins, by 15 ticket type groups
CAPC (LUL, NR, DLR)	Underground exit count, NR, DLR arrival (=exit) count. Annual, one day.	Number entering central area (~zone 1), all modes. Influx, not traffic.
CAPC (Road)	Cordon influx count. Annual, one day.	LT Bus, Taxi, Car, M/C, P/C passenger count, Minibus, Coach vehicle count.
GLBPS	On bus interviews. Continuous	Route details. Ticket, O/D stages
LRTS	CATI Household-based interview of single respondent. Ongoing (8000 hh pa).	Household (Size, Age groups, vehicles, HoH income, SEG), Individual (Demographics, work, work PC, driving licence, travel frequency by mode, vehicle access, disability), Diary (Recall yesterday, mode, ticket, O/D, purpose, group structure, luggage, choice reasons, issues, preferences).
Various	Includes ad hoc surveys focused on specific locations and interests.	

Other relevant sources will also be considered, such as appropriate information from the 1991 LATS round.

4.2 Overview of LATS Data Processing

The plans for processing the data from the various LATS sources are being developed separately, and can be loosely classified into five stages. These Stages need to interface to the various database Implementation Phases discussed in section 5.3.

4.2.1 LATS DP Stages

Stage 1 is the initial survey data entry and edit. The data collection contractors usually undertake this. In the case of the roadside survey, it is the subject of a separate contract. Stage 1 also covers the processing undertaken prior to the supply of data from other agencies, e.g. RODS data from LUL. The level of processing undertaken at this stage varies between surveys.

Stage 2 is geocoding. All the LATS surveys (other than counts) collect trip end addresses. The addresses are data entered as part of stage 1 processing, either as written on the questionnaire or, more often, with elements of 'intelligent' data entry to enhance the quality or, at least, standardise the format. These then have to be geocoded; i.e. a list of geo-codes, including postcodes, LATS traffic zones and grid references, has to be attached to each. A contract has been let to undertake this work. It will be undertaken using an automated system of address recognition, the 'LATS Coding System' (LCS), supported by a manual interface for dealing with addresses that the system cannot match automatically.

Stage 3 is the processing following on from stages 1 and 2, requiring understanding of transport planning data requirements and detailed knowledge of LATS and its objectives. It involves processing stages 1 and 2 outputs into a form suitable for provision to data customers and for input to database production and early data analysis. The tasks involved will vary from survey to survey. The processes of managing and undertaking this stage will usually involve an interface with stages 1 and 2 procedures.

Stage 4 is the production of combined trip matrices from stage 3 outputs.

Stage 5 is the database development and production. This is the longer-term database, which will be linked to the rolling survey programme design. There is a need to take stage 5 data and documentation requirements into account when conducting stages 3 and 4.

The boundaries between the stages can be fuzzy, and the level and type of processing required at each can vary between surveys. The purpose of defining the stages is solely to identify a general structure that will assist in mapping out tasks and identifying the management structures and resources needed to take them forward.

Although stage 3 data processing as a concept can be defined in only a general way because of variation between surveys, the tasks should generally be recognisable as such. They will normally include:

- Liasing with the stages 1 and 2 contractors, obtaining the data, preparing an assessment of quality and identifying any specific problems.
- Preparing a specification for further data cleaning, correction, imputation, etc.
- Carrying out the data cleaning, correction and imputation tasks identified as required.
- Bias correction, using survey specific data and special surveys undertaken for the purpose.
- Data expansion and indexation to the 'estimation period' using survey specific data such as counts and sample frames. The indexation process for the household surveys will involve reference to the 2001 census date, but the travel estimates will usually be for a 2001 average weekday, excluding school holiday periods.

Reference to exogenous data sources for bias correction and expansion purposes will be involved whenever appropriate. Examples are independent traffic counts, public transport ticketing data and the 2001 census.

Stage 3 will not normally include any form of inter-survey referencing.

4.2.2 Correspondence between DP Stages and development phases

This section relates to the implementation phases proposed in section 5.3.

The output of Stage 3 above will form part of the input to Phase 2, the data loading phase. Stage 4 will be part of the separate modelling study, and will also use the results of Stage 3 and provide input to Phase 2. However, subsequent modelling and synthesis activities will be more closely linked to the database system, as the later phases proceed.

Stage 5 above is essentially the whole database system as described in the current report, but with a much more limited view as to its purpose.

4.3 Links with other resources

There are various other information resources that could be used in conjunction with this system. Of particular importance is the TfL Planning and Information database (the PID, for Infrastructure, Service and Geographical data). Other examples of information that would also be of value include land use data and a proposed system for road traffic automatic monitoring data, and other established databases available for particular modes of transport in London such as buses (BODS) and the Underground (RODS).

Links are needed in both directions, so that LATS information can be used in other systems (both from systems providing additional analysis and presentation functionality, and systems with other information resources) and so that information from other systems can be used from within the LATS database system. The ability to physically transfer information to such systems will be valuable, but is unlikely to be sufficient. More dynamic, features will be needed that allow other resources to be used with the LATS information within the context of the LATS database, and other systems, to dynamically access information from the LATS resource.

The PID design (which is focussed on an Oracle database) explicitly includes a component for introducing information from external sources. This will almost certainly not be sufficient for communication with the

LATS system without further development. Users of LATS will want access to the service and location information in PID, as well as to the GIS functionality that it provides, when analysing and interpreting the LATS information, and users of PID will want access to summary information that has been processed in LATS. This particular interface will act as an important focus for establishing interface methods and standards for the LATS database, and for establishing cooperation with the providers of related information resources. Technical issues of standards and protocols for linking are discussed elsewhere.

The design components described in the previous sections are an important prerequisite for information exchange with other, cooperating resources. Data exchange mechanisms such as ODBC are likely to be important for such linking, but are unlikely to be sufficient, not least because they are limited to relational database structures, and will not know about the extended metadata available in the LATS system. So we will also need work on suitable exchange mechanisms, on procedural agreements with other resource providers, as well as on functionality to allow LATS users to make use of information from other resources, and for other systems to make use of information from LATS.

Apart from the technical issues, there are philosophical ones that arise from improved interchange of information. As an explicit example, the RODS information about underground usage is used to produce a best model of underground traffic. However, this model does not take account of other modes of transport in London. So when the RODS data are used for modelling within the LATS context of integrated mode information the ‘best estimate’ of underground usage will be different. In one sense it will be a better estimate, since it will make use of more information, but in a different sense it is not as good, since it differs from the established RODS best estimates. The differences can be identified and explained through suitable metadata, but there is still the problem of deciding which is more appropriate for use in different contexts.

4.4 Documents and analyses

It is intended that the system will include analyses, results and conclusions, expressed in both numerical and diagram form. These will be supported by textual commentary and descriptions, and linked to sources within the system. In some cases this can be achieved by attaching notes (as labelling metadata, with little formatting) to tables or charts used for presenting the core information. However, often the commentary will be the main component of the analysis, and will come as a fully formatted document, with referenced information included as diagrams or notes to confirm or validate the substantive policy conclusions. It will be important to be able to store these complete documents in the database, still making use of the referencing and discovery facilities.

It is possible to store complex (binary, not data) objects in a database. Examples are Word documents, Excel spreadsheets and PowerPoint charts and diagrams. The objects as a whole can easily partake in the general facilities of the database, including being classified, referenced in catalogues and by bookmarks. It is not generally easy to extend their internal structure to introduce derivation or metadata links.

Indexing systems exist that can support text searches within such objects (e.g. the Content Indexer in the MS IIS web server, or the full text indexer in Oracle). So there should be no particular problem in the content of such objects being discovered through free-text searches.

Access to such objects may be limited to specific operating systems (specifically Windows). Many types of object can be displayed through plug-ins in a web browser, while others need more powerful helper applications. Databases of this type already exist, such as the nVision[15] service from the Future Foundation.

4.5 Synthetic Information

The importance of synthetic information is fundamental to our view of the database as a general resource. Modelled results are being considered in database terms as data, including results from forecasts.

[15] www.nvisiononline.co.uk

An important objective of the synthesis work is to generate standard, validated, updateable Base matrices for use by modellers and others, but, in turn, modelling is the process by which such synthetic information (base matrices and others) will be produced.

The conceptual and algorithmic aspects of modelling and data synthesis are discussed in a separate report[16]. This section summarises some of the important issues, and is based on material taken from that report.

4.5.1 Motivations for Synthesising LATS Data

The LATS 2001 database is required to be rich in the type of information that it stores in order to support the wide range of transport policies being considered in London. These include encouraging an integrated view of transport across different travel modes, as well as policies that manage the demand for transport.

A particular issue arises from the different component surveys occurring in LATS that observe some travel data more than once. This provides a useful crosscheck on key statistics but also implies difficulties in the case of discrepancies. Avoiding possible serious discrepancies arising is a concern of LATS 2001, since the problem was encountered in the LATS 1991 surveys.

Aided by developments in the field of trip matrix estimation and by research commissioned by Transport *for* London, there is seen to be a realistic prospect of using data synthesis techniques as a means of obtaining LATS data that are rich in content, self-consistent across and between data categories, and statistically sound.

The introduction of data synthesis processes runs the danger of affecting the perceived credibility and assurance that is associated with directly observed information. However, there is a view that relying on direct usage of observed data (as was generally the case for LATS 1991 and previously) can itself result in problems that constrain the exploitation of the data. Hence, if a sufficiently sound data synthesis methodology can be identified then (almost paradoxically) this can be a preferable basis for furnishing a reliable database of travel information for the London region.

The observed data is essential to any modelling or synthesis, but it suffers from various limitations:

- It represents the immediate moment, not the underlying process.
- It suffers from variability (seasonal effects, day of week, time of day, weather, events, right down to the level of people making apparently random decisions).
- It is incomplete and inconsistent, because we cannot observe everywhere at once, and cannot always observe what we really want to know.

By using a model we can bring together information from different sources, and we can include prior information, knowledge and understanding. Of course, we cannot create any information that is not already in these resources, but we can bring it together in a consolidated and coherent view that allows us to perceive more than is possible from the fragmented and variable picture given by individual sets of observations. Altogether we can produce a consolidated view of the underlying processes that balances conflicting individual datasets and explicitly quantifies the amount of information (precision) that we have about different facets of this view. We are still dependent on the quality of the model, but this can be validated, against the data and by understanding its assumptions and implications.

4.5.2 Using Synthetic Information

Once the model has been fitted (or calibrated) against the available data and our assumptions, we can use it to generate synthetic information about any aspect of the modelled system. In particular, this allows us to generate synthetic summary matrices based on our ‘best’ or ‘base’ estimates about the underlying processes, together with information about the precision of the synthetic information.

[16] LATS 2001 Data Synthesis: Specification of Data Synthesis Methodology by Miles Logie

These best estimates of corresponding parameters or measures (for the transport system that the model describes) will often correspond to summary information that can be obtained from observed data (though a model will also allow us to produce estimates for things that cannot be observed directly). The advantage of the synthetic estimates over the direct ones is that the model will have smoothed out and adjusted for factors (both random and predictable) that cannot be controlled during the data collection.

As well as producing 'best' estimates (with nuisance factors smoothed out) we can use the model to produce estimates for more extreme situations, perhaps to investigate the system response to difficult situations (road or station closure, strikes, flooding, for example).

A fitted model contains components for variability as well as parameters of basic rates and relationships. Using the variability components we can generate data (from the model) that corresponds to observations that might have been collected in surveys, representing the experience of (synthetic) individuals. This synthetic population data can then be presented and analysed using the same methods as for real data. Of course, we can never discover anything that is not implicit in the fitted model, but it can be easier to present and demonstrate features this way (particularly those related to variability) than from the 'best' estimates. Synthetic population records are sometimes called Simulated Data.

Note that we can choose which components of the model are fixed and which are variable when synthesising records. So we can produce different synthetic populations that demonstrate different aspects of variability (with or without seasonal effects, or weather variability, for example), and can explore extreme scenarios (behaviour with road or bus route or tube line closure, for example).

4.5.3 Aims

Having established the value of producing synthetic information through model fitting, work is accordingly in progress to specify a methodology that:

1. Establishes a LATS database drawing information from all component surveys that is self-consistent across and within temporal and spatial groups, journey types, travel demand segments, and so on.
2. Provides a mechanism for updating the database in future years as new survey data become available.
3. Provides a view of the database’s strengths and weaknesses in respect of precision and variability.
4. Allows researchers and others to understand the data synthesis process and the consequences arising from it.
5. Defines the methodology in a manner that permits its implementation in one or more stages through one or more contracts.

4.5.4 Problem Formulation

The LATS data synthesising problem is formulated in broad terms, taking a holistic view of the matter. It seeks to make use of all LATS 2001 observed data sets, as well as allowing for a range of anticipated future surveys.

4.5.5 Solution Methods

The solution methods are based a number of well-established techniques, but involve a distinctive process that makes the approach innovative.

The statistical pillars of the methodology are the widely used techniques of Maximum Likelihood and Bayesian methods. The Bayes theorem is well suited to using partial (conditional) information in association with basic information to generate more extended information.

The Maximum Likelihood method can estimate the parameters of a ‘synthesising function’ that calculates synthesised data values that can be shown mathematically to be the statistically most likely values, given the values of data that are observed, and subject to certain explicit assumptions.

Maximum Likelihood allows an objective function to be derived that has the property that its maximum value corresponds to the best (that is statistically most likely) settings of the parameters. Finding this maximum value, and hence the best settings, requires an optimisation method. The objective function considers all the relevant data together, and this provides a means of integrating information from different survey sources.

The implementation of the optimisation algorithms is likely to make use of computationally intensive methods such as the MCMC (Monte Carlo Markov Chain) and EM (Expectation–Maximisation) algorithms. These have been shown to be effective in other disciplines for problems of similar complexity, so we are confident as to their suitability in the transport-modelling domain.

The methodology considers both direct and indirect observations of the data to be synthesised. Indirect observations may require the involvement of transport modelling methods, and the consequences of this have relevance both to the LATS database and, potentially, to modelling practices.

4.5.6 Implementation

The implementation of the data synthesis methodology is a significant undertaking that will be the subject of a separate contract. The facilities of the LATS database will provide the important data handling capabilities, including the ability to access and aggregate data in varied ways and to keep audit trails. This means that work on the synthesis component can focus on methodological and algorithmic matters, and so the marginal cost of the data synthesis is accordingly considerably reduced.

We intend first to prove the methodology with sample data using standard statistical package software. We will then consider how best to implement the more customised software facilities that this demanding application is likely to require. These facilities will support the research and exploration of different methods that should be a supporting activity to the implementation. Some aspects of the implementation, such as precision estimation, can be elaborated once a central capability is established.

It will be important to provide varied documentation supporting the LATS 2001 data synthesis methods so that the techniques are approachable for both non-specialists and specialists.

4.5.7 Database implications

From the database side we need to be able to store the results produced by modelling or data synthesis, together with appropriate metadata. Several factors are important.

- Storing the synthesised information so that it can be used.

  Much flow data (that ignores routing) can be seen as a multi-dimensional structure, with two location dimensions (one each for Origin and Destination) and other dimensions for factors such as mode, time (time of day, day of week and season) and trip purpose.

  Other types of model may need other, more general and flexible structures. For example, trips or tours including segment information require a more complex structure.

  All synthetic information (and much other data) will require associated variability (or distribution) information, so that confidence or precision can be assessed.

- Handling multiple versions of synthesised information, perhaps using different assumptions or augmented with new data.
- Storing the conditions and assumptions used for particular realisations of synthetic data, including sources and parameter estimates (with their precision or distribution information).
- Storing process information about synthesis – at least enough for an adequate audit trail, and ideally sufficient to reproduce the process automatically.

5 Development Components and Phases

5.1 Introduction and Approach

Almost everything we are discussing can be done by building a specific implementation in a suitable programming language running in conjunction with a relational database system (probably either Oracle or MS SQL Server). However, it will be extremely difficult and expensive to produce a generic solution, capable of dynamic evolution, in that way. Object-Oriented programming and database ideas appear to be much more promising as a base solution technology.

5.2 Discussion with Suppliers

In addition to discussions with potential users, an interim version of this report was sent to eight organisations that develop, implement or support software and systems related to statistical databases. Six replies were received, two supportive but brief, and the four longer ones are presented as appendices. The intention of this discussion was to identify any areas possible suppliers saw as particularly difficult to implement. The organisations approached are mostly from the statistical database area, and do not have particular (or any) experience with transport. The general conclusion seems to be that all the ideas presented are consistent with ideas and expectations elsewhere, though the implementation of the whole will be an interesting challenge.

5.3 Implementation Phases

This is a possible organisation into phases, using the various Solution Components listed below. It recognises that the dependencies mentioned are not necessarily absolute or one-directional. Each phase will include some preparatory work for subsequent ones, so that later developments are facilitated. Most components have both basic and more advanced facets, the latter often dependent on other components. In general the advanced components do not need to be implemented until later phases – these sub-components are not shown.

Phases 1 and 2 are required to produce the most basic usable system. Subsequent phases are sequential, and add more layers of functionality. It should be possible to cease development after any phase and be left with a useful and usable system, so each phase should include a review of the benefits of the next one, leading to a decision whether to proceed with detailed planning and tendering.

Table 4 Implementation Phases

Phase	Included Components	Main Deliverables
0		**Preliminary stages**, Specification Study, Tender call, review and selection for Phase 1
1		**Initial functionality** to hold clean data records and summaries. Interface and tools for Transport Specialists, including Basic Metadata. Includes aggregation and manipulation tools for the preparation of summary information, basic presentation facilities, and export functionality for transferring information to other systems.
		Modelling Study (in parallel)
	SCR:1	Clean Data Records
	SCR:2	Summary (Aggregate) information
	SCR:3	Base Matrices
	SCR:7	Basic Metadata (Classifications and Labels)
	SCR:10	Data Loading Tools

Phase	Included Components	Main Deliverables
	SCR:12	User Interface Design Study
	SCR:13	User Interface: Transport Specialists
	SCR:15	Presentation tools for summary information
	SCR:16	Statistical manipulation and derivation for micro and aggregate information
	SCR:24	Export facilities, including metadata
2		**Data Loading Phase**, plus Data Manipulation Initial Population of the database and base demand matrices, initial base (synthetic) estimates. Metadata for the clean data records will be obtained from the input and cleaning stages, and for the summary information from the aggregation processes. This phase will overlap with Phase 1, using the loading, storage and manipulation facilities as they become available. It will start as soon as information is available from Stage 3 of the DP plan. It will also take inputs from the processing stages of the modelling study.
3		**General User facilities** Basic facilities and interface for non-specialist users, including discovery and presentation tools. Includes bookmarks, discovery through catalogues and thesaurus. Content extended to include discussion and commentary, charts and diagrams, plus full-text search facilities. User management and access control facilities will be needed.
	SCR:5	Analysis and Commentary, with full text search and linking (through bookmarks) to and from sources
	SCR:6	Graphs and Charts
	SCR:8	Conceptual Metadata (Catalogues, Thesaurus)
	SCR:11	Bookmarks (general linking between information)
	SCR:14	User Interface: Other Users
	SCR:21	User Management
	SCR:22	Access Control
	SCR:30	Operation and Management interface
4		**Updating and Process management**: History capture and execution. Version Control. Updating for basic information, summaries and synthetic estimates. Derived precision information for all measures.
	SCR:4	General Synthetic Information
	SCR:9	Additional Metadata, (Processes, etc.)
	SCR:17	Version Control
	SCR:18	History recording (Capture of processes)
	SCR:19	History Execution (Rerun captured steps)
	SCR:20	Support for Updating, where new versions of information are added to the database
	SCR:23	Variability attributes for all numeric information: automatic derivation and updating

Phase	Included Components	Main Deliverables
5		**Advanced Features**. More general synthetic estimation, including model representation and generation of synthetic estimates and populations. Linking to external systems.
	SCR:25	Linking to and from other systems: interfaces, protocols, structures, standards, agreements
	SCR:26	Modelling Support, holding model definitions and linking to synthetic information
	SCR:27	Generation of Synthetic Information from models
6		**Full support for modelling**
	SCR:28	Model Fitting within or linked to the database

5.4 Solution Components

The following table identifies some of the components that will be needed in the implemented solutions. It is drawn largely from an analysis of the Design Components elaborated previously.

The Start Phase column indicates the stage at which work on the component needs to begin (as shown above), though many have sub-components and extensions that can happen later. All components must be implemented in a manner that supports the later components that are dependent on them. In many cases this support will not be completely present from the beginning, but will require development or even some re-implementation of the earlier component. The important point is that the implementation of the early components must not ignore or inhibit the later components.

Table 5 Solution Components

Reference	Component	Implementation Issues	Start Phase	Depends on	Implements part of
SCR:01	Clean Data Records	The use of an RDBMS will give the quickest implementation for the storage of the basic (clean) information from the various LATS surveys. There may be advantages in using Data Warehouse systems for this, but these are unlikely to be sufficient to justify significant additional cost or effort. The solution must maintain the relationships between the various data sets.	1	SCR:10	DCR:1
SCR:02	Summary (Aggregate) information	The choice of a suitable tool for this component will be crucial to the success of the implementation. Candidate tools will be available from Data Warehouse, OLAP and Statistical Database suppliers. Adequate statistical support and extensibility will be crucial.	1	SCR:10	DCR:1
SCR:03	Base Matrices	These will be obtained from the processing stages of the modelling project, at least in their preliminary form. Implementation will be as a special case of summary data (SCR:02).	1	SCR:10	DCR:1

Reference	Component	Implementation Issues	Start Phase	Depends on	Implements part of
SCR:04	General Synthetic Information	Once the more advanced modelling functionality has been developed (outside the database) it will provide better and more forms of synthetic information. Some will fit into the structures for data and aggregate information, but others will need new structures. So the solution will be similar to SCR:01 and SCR:03, with extensions.	4	SCR:10 and/or SCR:27	DCR:1
SCR:05	Analysis and Commentary, with full text search and linking (through bookmarks) to and from sources	Storage of (machine readable) documents (probably in the form of Word, Excel and .PDF Blobs, including graphics) in the database. Requires full-text indexing (as in Oracle), and depends on the bookmark service for the construction of catalogues and links.	3	SCR:10, SCR:11	DCR:1, DCR:2
SCR:06	Graphs and Charts	Basic business graphics (cf Excel, and SCR:05) as a presentation tool, but see also SCR:15. More sophisticated facilities (including mapping) will use external functionality via system links.	3	SCR:10	DCR:1, DCR:4
SCR:07	Basic Metadata (Classifications and Labels)	It is relatively easy to build general structures and functionality for this simple metadata in a relational system, though the normal solution proposed by relational database specialists (normalised label tables for every field) is not appropriate. A specialised tool would be much more flexible.	1	SCR:10, SCR:11	DCR:2, DCR:4
SCR:08	Conceptual Metadata (Catalogues, Thesaurus)	This area is well understood by specialists in the field, and not too difficult to implement. A specialised metadata tool, as with SCR:07, would be better. Setting up the information content for these elements is a considerable amount of work.	3	SCR:11	DCR:2
SCR:09	Additional Metadata, (Processes, etc.)	As SCR:07 again. A more general solution that extends to more complex structures and functionality is considerably more difficult.	4	SCR:11	DCR:2, DCR:6
SCR:10	Data Loading Tools	The components chosen or implemented for holding the various types of information will generally include tools for loading their own type of data, but some extension and integration is likely to be needed. The extensions will largely relate to loading and/or generating suitable metadata (including links) as the information is loaded, and the integration will be to provide a more general operational environment for controlling the loading processes.	1	SCR:29	DCR:18, DCR:19

Refer-ence	Component	Implementation Issues	Start Phase	Depends on	Implements part of
SCR:11	Bookmarks (general linking between information)	This is relatively straightforward with an object-based system, since objects have identities and the linking concepts can be built into the base object classes. With a relational or hybrid system it is likely to be much more difficult to implement and maintain functionality that operates across systems.	3		DCR:3, DCR:2
SCR:12	User Interface Design Study	Important to do this properly, and to get it underway early, with most effort on the general user interface.	1		DCR:4, DCR:8
SCR:13	User Interface: Transport Specialists	DBMS interface, Web browser, other packages (through links). Can rely on the knowledge and expertise of the 'in-house' or controlled users.	1	SCR:16, SCR:15, SCR:21, SCR:22	DCR:4, DCR:8
SCR:14	User Interface: Other Users	Web browser the most likely solution, but more specialised browsers, still operating over the Internet (cf Beyond 20/20) may come into consideration.	3	SCR:08, SCR:11, SCR:15, SCR:21, SCR:22	DCR:4, DCR:8
SCR:15	Presentation tools for summary information	OLAP or specialised statistical browser, but needs extensions for links, histories and additional metadata.	1	SCR:07, SCR:11, SCR:22	DCR:4
SCR:16	Statistical manipulation and derivation for micro and aggregate information	As SCR:15, but with further extensions for management of derivations.	1	SCR:07, SCR:11, SCR:22	DCR:5
SCR:17	Version Control	Feasible with Objects, otherwise hard to do in a general way.	4		DCR:9
SCR:18	History recording (Capture of processes)	Needs to be programmed in to all functionality in the system – difficult when using components that have their own parameters and access to information (components that are not encapsulated).	4	SCR:09, SCR:11	DCR:6
SCR:19	History Execution (Rerun captured steps)	Straightforward given SCR:18 – complications from SCR:17 and the need to allow users to vary details.	4	SCR:17, SCR:18	DCR:6, DCR:9
SCR:20	Support for Updating, where new versions of information are added to the database	Should be straightforward, given the components it depends on.	4	SCR:10, SCR:17, SCR:19	DCR:18, DCR:19
SCR:21	User Management	Standard problem.	3	SCR:12	DCR:7, DCR:8, DCR:19
SCR:22	Access Control	Standard problem, but good ideas, including implementation components, coming from the Faster project, perhaps also from OMG initiatives.	3	SCR:21	DCR:7, DCR:19

Refer-ence	Component	Implementation Issues	Start Phase	Depends on	Implements part of
SCR:23	Variability attributes for all numeric information: automatic derivation and updating	Similar problem to SCR:18, plus algorithms for calculation – research proposed in Synthetic Estimation study.	4	SCR:01, SCR:02	DCR:1
SCR:24	Export facilities, including metadata	Standard solutions available for micro data, e.g. Triple-S, DDI. Standards under discussion for summaries and more complex structures, through OLAP and statistical databases.	1	SCR:01, SCR:02, SCR:03, SCR:04, SCR:07	DCR:10
SCR:25	Linking to and from other systems: interfaces, protocols, structures, standards, agreements	Should build on existing standards and discussions (using e.g. ODBC, XML, etc.), but with extensions for statistical, transport and metadata coverage.	5	SCR:07	DCR:11, DCR:14
SCR:26	Modelling Support, holding model definitions and linking to synthetic information	Should not be too difficult, given the components it depends on.	5	SCR:07, SCR:09, SCR:16, SCR:18	DCR:12
SCR:27	Generation of Synthetic Information from models	Should not be too difficult, given the components it depends on.	5	SCR:19, SCR:26	DCR:12, DCR:1
SCR:28	Model Fitting within or linked to the database	Significant additional components for specification and estimation of models.	6	SCR:26	DCR:12
SCR:29	Backup and Disaster Recovery tools	Standard.	1		DCR:17, DCR:19
SCR:30	Operation and Management interface	Standard problem.	3	SCR:10, SCR:21, SCR:22	DCR:19

5.5 Dependency Charts

Appendix 3: attempts to visualise the dependencies between the Solution Components in the form of Gantt charts.

6 Underlying Concepts

This section contains further brief discussion of some of the elements in the database and the technologies that can contribute to it, as background and justification for some of the ideas that have been presented.

6.1 Individual data records

Individual records are readily stored in a Relational Database management system (RDBMS). These are well understood, and provide a wide range of facilities for data access and structure. They tend to be weak in facilities for data discovery, exploration and presentation: generic facilities always exist for searching and for writing reports, but these are designed for building static systems, not for flexibility.

RDBMS designers have generally not made explicit provision for metadata, so that requires additional facilities.

Statistical packages often have considerable flexibility in terms of analysis facilities, but do not provide sufficient flexibility for data storage or access to be used as the primary storage for the data.

Exporting or linking data records to other systems is not a difficult problem, since the ODBC standard is widely used by systems that process or store individual data records. Exchanging basic metadata is rather more problematic, but agreement and standardisation of structures and semantics is fast approaching, so the LATS system can draw on that work.

6.2 Relational Databases

Relational database systems **exist** and work **effectively** for **micro–data**. Many of the front–end **tools** supplied with commercial systems are useful. The principles of **data–independence** and explicit representation of **metadata** have obvious application in statistical systems. The Client-Server approach, in which application systems (including many of the major statistical systems) operate as front–end clients to DB server systems, has become the norm. Relational systems have great **flexibility** for linking information on the basis of data values.

The relational model is **not rich** with features for statistical applications: additional data types (Domains and semantics), data structures and operators could all provide much better support for statistical uses. Earlier proposals for statistical extensions to the relational model or as conceptual models have had little impact.

Entity–Relationship design and database design techniques in general provide an extremely useful approach for analysing data structures and data requirements for complex structures of individual records. The relational model provides a very useful way of thinking about some types of statistical data. Even if you don't use the model for implementation, it's still a good idea to think your data through in these sorts of terms. However, there are certainly limitations in the model, particularly for aggregate data.

Links between statistical packages and relational DBMSs have improved, but the DBs have not provided any major additional features for statistics. Object Oriented systems are much more promising from this perspective.

6.3 OLAP – On-Line Analytical Processing

Following on the success of relational databases, database scientists began to perceive the need for tools to analyse the large amounts of information being gathered in large systems. From this the idea of Data Warehouses and the related OLAP tools developed. Whereas most RDBMS systems are optimised for transaction processing with active data records, data warehouses (DW) are designed for processing large volumes of static information in dynamic ways. A DW may fulfil the main requirements of the relational model, but the implementation will be quite different from a RDBMS.

The main form of analysis offered is the exploration of aggregate data. Recent efforts have focussed on automatic exploration through Data Mining tools, but the early work was on the manipulation of aggregate data summaries, for which the OLAP tools were developed. Aggregate data is viewed as a multidimensional hypercube (usually just called a data cube), in which the dimensions are classifications and the cells contain measurements in the form of counts or sums. The dimensions usually have structure and form a hierarchy of levels (as with geographical classifications), and the OLAP tools contain functionality to increase or decrease the level of detail as the information is being viewed – this functionality is usually referred to as 'drill down' and 'roll up'. Note that rolling up a complete dimension has the effect of reducing the dimensionality of the data cube by one, producing the marginal table over the removed classification. The functionality to select subsets by selection both within and across dimensions is usually referred to as 'slice and dice'.

This structure and functionality is clearly relevant to much survey information, since each summary table is at base an example of a data cube. However, there are some complications. This work is based on traditional computer science views of databases, and does not give much attention to the specific requirements of statistics. The functionality provided concentrates on manipulation and selection (which are essential, but not sufficient). Not much help is provided for derived variables (other than sums) and for their maintenance across the other operations. More abstract issues, such as validity rules or contextual metadata, are not usually addressed.

Many of the major RDBMS suppliers have produced OLAP facilities, and several independent commercial developers have produced specialist DW systems. Oracle have the Oracle Express component, and Microsoft have OLAP facilities built in to SQL Server (from version 7), and the Pivot Table component in MS Excel can be used as the presentation and manipulation interface to a data cube.

Many issues of standardisation for DW and OLAP systems were addressed in the Common Warehouse Metamodel (CWM) initiative, which was approved by the Object Management Group (OMG) in 2000. This is a complex proposal (already being revised) and time is needed before its implications can be fully understood.

6.4 Statistical Databases

OLAP systems suffer from having been designed by Computer Scientists, largely without reference to statistical ideas. This is despite the presence of a strong thread of interest by a number of computer scientists in statistical database issues, as evidenced by the series of SSDBM (Statistical and Scientific Database Management) Workshops, which have been running since the early '80's. The most obvious omissions are:

- any treatment of variability, through automatic calculation of standard errors of any similar measure,
- support for extended labelling, whether at the level of classification elements or data values (footnotes),
- support for derivations other than sums,
- any conceptual underpinning, for example to address whether it is sensible to combine two classification elements or to sum rates.

In general this can be seen as an absence of statistical metadata and statistical processing functionality.

A number of statistical agencies have tried to build specialised systems to support their information processing and dissemination requirements – an early example is the PC-Axis system from Statistics Sweden. This has led to a significant investment in research and development in statistical metadata and statistical processing systems, largely funded through the EC R&D programmes and organised by Eurostat. While no clear solutions have emerged in the form of products, there has been significant progress in the formulation of ideas and concepts, and there are a number of promising 2nd phase projects underway. Examples are the Faster project (www.faster-data.org) led by the Data Archive at Essex University, which builds on the Nesstar project (www.nesstar.org), the Mission project (www.epros.ed.ac.uk/mission) led by Edinburgh University, which builds on the Addsia and Idaresa projects, and the Bridge system being developed by Run Software (www.run-software.com), which is a development from the IMIM project.

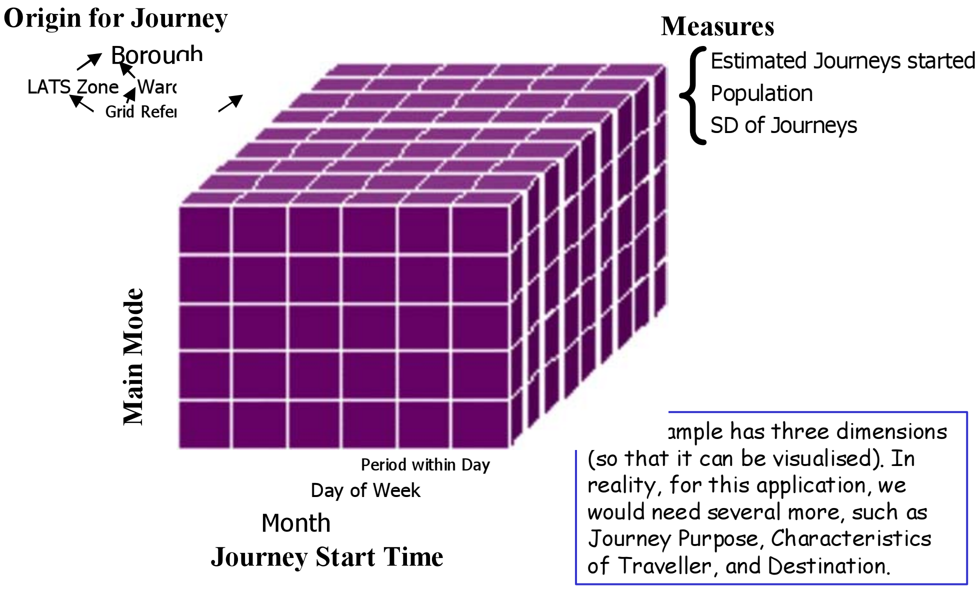
A number or commercial developments are also underway from specialist companies. Space-Time Research (www.str.com.au) in Australia have their Super-Star system, and are cooperating with the Bridge developments. Beyond 20/20 (www.beyond2020.com) have sold systems into many government agencies, for example in USA, Canada and France. The system is a Client/Server application where the client software is a browser with several versions. The lightweight browser has limited functionality (display with manipulation and selection) but is cheap enough to be given away to clients. Those who need more functionality, for example to derive new measures or redefine classification structures, can invest in a more expensive version. The client browser can work with local files or information requested from a remote server.

These specialist tools are valuable, since they pay much more attention to statistical issues than OLAP systems. Some are being used as complete census processing systems (from data capture through to on-line interactive dissemination), but none forms a really complete solution for LATS.

Along with these developments have gone a number of standardisation initiatives, particularly on statistical data structures and metadata. Some coordination and consolidation of these efforts is being attempted through the MetaNet project (www.epros.ed.ac.uk/metanet).

6.5 Multi-way Tables (Data Cubes) for Aggregate Data

Figure 4 A Data Cube for aggregated data – Estimated Trips



ample has three dimensions (so that it can be visualised). In reality, for this application, we would need several more, such as Journey Purpose, Characteristics of Traveller, and Destination.

The logical data structure for summary information is a **multi-way table**, a hyper-cube with multiple dimensions and with cells containing multiple values. Each **dimension** has a classification structure, with consolidation from detail to broad groups through various levels of aggregation in a hierarchical tree. Each cell of the table contains multiple **measures**, the same in all cells, which can be counts, sums, means, or other expressions. It is common parlance to talk about aggregate information being stored in a **data cube**, rather than a multi-dimensional hyper-cuboid, even though there will usually be more than three dimensions, and they will be of differing lengths.

A few academic researchers from computing science have been looking at statistical database issues for some time. A significant contribution is the STORM[17] proposal from Rafanelli and Shoshani, which offers a formal model for aggregated data (in the form of a data cube). The model formalises the concepts of dimension and

[17] M. Rafanelli, Shoshani, A "STORM: a Statistical Object Representation Model". In Michalewicz, Z. (Ed.). Statistical and Scientific Database Management, V SSDBM. Lecture Notes in Computer Science, Vol 420, Springer Verlag, 1990

measure, so that, for example, each measure must have an aggregation process defined. Other details relating to the underlying population and the selection rules are important. Later work by Lenz and Shoshani[18] has looked at different classes of summary measure and the rules for summarising across categories.

6.6 Metadata

The term *metadata* is used widely in the context of database and statistical systems, though there is not general agreement on exactly what it means. We take a very broad view:

> *Metadata is anything that you need to know to make proper and correct use of the real data, in terms of reading, processing, interpreting, analysing and presenting the information. Thus metadata includes file descriptions, codebooks, processing details, sample designs, fieldwork reports, conceptual motivations, etc., in other words, anything that might influence the way in which the core information is used.*

Metadata can be used informally by people who read it (and use it to affect the way they work with or interpret information), and formally by software to guide and control the way information is processed. Processes can also generate metadata.

Much of the metadata will be in formal, machine processable form, so that it can be part of the input to other processes. Much of it will be searchable, so that users can discover whether the database contains information of interest. Some of it will be formally structured, some of it not, some will describe concrete aspects of particular stored information, other will relate to more abstract concepts that underlie the objectives of the information.

Properly structured metadata is essential for

- effective discovery of interesting information, through linking between descriptions and information structures,
- dynamic exploration of summary information, with facilities to reduce or expand detail, and
- automated data exchange that includes appropriate metadata.

Note that this is much broader than the view represented by the Dublin Core[19] standard for metadata, which is essentially a cataloguing approach. It specifies certain items of descriptive information that are needed to accompany a dataset: while valuable, this is restricted to purely textual material.

6.7 Metadata Management

The capture or collection of metadata should ideally be integrated with the creation of the resource to which it relates. Experience has shown that creating the metadata manually as a separate, subsequent process is error-prone and time-consuming. Thus the creation of physical and operational metadata should be built into the design or production processes of the resources, and, as far as possible, the descriptive and conceptual metadata should be created as adjuncts to these processes.

The concept of linking is essential for metadata, so that, for example, it is possible to move from reading the description of something to looking at the thing described, perhaps in the style of hyperlinks. Linking is also needed for statistical information. For example, with aggregate data we must be able to link from the dimensions of a data cube to the variables in the source data that were aggregated, and to the classifications used to define the possible groupings for each dimension.

[18] H-J Lenz, A. Shoshani "Summarisability in OLAP and Statistical Databases". In D. Hansen, Y. Ioannidis, Proceedings of SSDBM 9, IEEE Computer Society, ISBN 0-8186-7952-2, 1997.

[19] Dublin Core Metadata Initiative – www.dublincore.org

With metadata it is important to represent abstract concepts as well as concrete instances of these concepts. For example, the idea of the Purpose of a Journey will need to exist within the database, so that it can be defined, and this concept will need to be linked to classifications that distinguish different purposes, and to variables that record the purpose (according to a classification) of actual journeys. Thus a user will be able to link to all the components of the database related to the concept of Journey Purpose, not simply to find components that include the text 'Journey Purpose' in their description or label.

6.8 Use of XML for Structure and Standards

Metadata tends to have complex structure, so it is complicated to store and use effectively. There have been efforts to agree on structure and definition for some years, with considerable funding input from National Statistical Offices and Eurostat over the last decade. This is beginning to bear fruit, with some agreement on principles emerging. In addition, various groups have been proposing standards in particular specialised areas, an activity much motivated by agreement on the XML[20] (eXtended Markup Language) standard.

The significance of XML arises from its ability to describe data and data structure in a manner that allows varied systems to access that data. A familiar example of this issue is provided by relational databases: these store data in a defined structure (according to a standard structural model) that is then readily accessible using such technologies as ODBC. While ODBC is an important and efficient mechanism for exchange of data that fits into the relational model, it not extensible in any way to structures outside that model. In contrast, XML allows the structure of the information to be specified, and so removes the constraints of relational structures and, significantly, allows information to be viewed in more powerful ways as 'objects'. The object approach (the object-oriented paradigm) allows semantics and operations (methods) to be associated with the definition of the data structure.

XML structures are effectively self-describing, through the use of an associated XML schema definition (or through a DTD – Document Type Definition). It is not, however, possible to embed semantics and methods of the structures in a XML schema definition – this has to be done through a separate standardisation agreement among the users of the structural standard.

XML is supported by various technologies including 'SOAP' (Simple Object Access Protocol), which allows systems to be distributed over different machines and locations, as well as by application programming interfaces like 'DOM' and 'SAX'. This is significant for major systems such as LATS, which are also seeking to be accessible and to develop over time. XML can play an important role in simplifying the implementation of all the external connections shown in Figure 1. For interfacing to systems that require specific data formats, XML has associated style (XSL) and transformation (XSLT) languages that can filter an XML datastream into a different (physical) structure.

6.9 Object-Oriented approaches[21]

6.9.1 Object-based methods

Traditional programming languages and methodologies recognise the need for both algorithms and data structures, but they tend to keep these two things separate. Database design methodologies, such as Entity–Relationship modelling address issues of how data is used, but then concentrate on defining the right data structures to support this use.

The object-oriented (O-O) approach (which started with programming languages, but has much wider application) recognises the central importance of process alongside structure, and keeps the two things closely to-

[20] www.w3.org

[21] Some of the material in this section is based on descriptions of object-oriented methods found on the Internet at www.teleport.com/~bstonier/devhbook/objectori.html and www.cslab.vt.edu/vse/UsersGuide/chapter_1.htm

gether. The ideas were first proposed in the 1960, and have been dominant in programming theory for the last 15 years or so. The most widely used O-O languages are C++ and Java, but even Visual Basic, Delphi and Fortran 90 make use of O-O ideas.

Central to object-based methods is the idea of a **Class**. This is a generic definition of a type of object. A class will have properties, which are the attributes that describe an object, but also behaviour, specified by the types of action that the objects can be asked to perform. The properties (or Attributes) associated with a class are not limited to simple measures, but can be complex structures including other objects and collections of objects. Behaviour is implemented through functions (usually called Methods), which are specific to a class. Access to the attributes and methods of a class is only available through a well-defined Interface, which protects the internal aspects of an object from external interference (or side effects).

Along with O-O programming languages have come O-O design methodologies. In recent years these have coalesced into a standard called the Unified Modelling Language (UML). Although initially focussed on program development, UML is rich enough to assist in the design of any dynamic system that contains objects with attributes and behaviour.

6.9.2 Object Concepts

There are many variations of object-orientation, and therefore many different definitions[22] of what it means to be object-oriented in its purest form. The three most basic elements, however, are Encapsulation, Inheritance, and Polymorphism.

- **Encapsulation** is the packaging of data and methods into a single unit, usually entitled a class.
- **Inheritance** is the ability to use and extend existing logic by deriving a new class from an existing one.
- **Polymorphism** is a way to treat objects of different classes in a generic way, so that you don't need to know the type of the object you are interacting with.

According to Booch (one of the authors of UML), the usage of the object model for the design process of an application offers several advantages over traditional techniques:

- The application of the object model substantially simplifies the development of complex programs.
- The object model facilitates the reuse of both code and designs.
- Systems designed according to object-oriented principles can evolve over time and can be adapted easily to accommodate future demands, without necessitating the abandonment or complete restructuring of the original design.
- Object orientation reduces the risks in software development.
- The object model resembles human cognition more closely than traditional design paradigms.

6.9.3 Unified Modelling Language (UML)

The UML standard was developed within the Object Management Group (OMG) as a way to design and represent object models. It is a collection of diagram types and components for representing various types of object and behaviour. It is a formal specification with semantics and conventions for representation of every element of a model.

UML recognises that complexity is at the heart of most modelling, and it provides specific functionality to support this. For example, the same items (whether classes or objects or some other element) can participate in multiple diagrams, with different emphasis or different level of detail or abstraction. This corresponds to the

[22] Some more extended definitions appear in Appendix 2:

idea of views in relational databases, where the same information can be viewed in different arrangements to meet different needs, or to reveal different aspects of its structure or behaviour.

It also recognises that designs must exist at different levels of detail and need to represent different aspects of the behaviour of a system. This extends from User Requirements (in Use Case diagrams) through Class and Object definitions, down to coding and implementation (Statechart, Activity, Sequence, Component and Deployment diagrams).

The origin and emphasis in most UML descriptions is on software implementation, but there is potential for much wider application for the design of any system that can be conceived in terms of objects. It is rich, complex and extensible.

A number of tools for designing in UML exist, and it is a requirement of the standard that they are able to exchange design information (which is done using XML). Several design methodologies have been developed (generally for software development), consisting of rules and guidelines about how to design good systems. UML thus provides a potential mechanism for a system to be designed in a way that supports interchange between development teams and extension over time.

6.10 Modelling transport

The conceptual and algorithmic aspects of modelling and data synthesis are discussed in a separate report (see 4.5), and some of the important issues have already been discussed. Our objectives in the database system are to provide facilities that support the methods and procedures recommended by that report. In the longer term we hope to be able to support both the expression and fitting of models within the system, but, realistically, this may not be possible in the short term.

A model is a mathematical specification of how (part of) the (London) transport process works and responds to internal and external factors. The model includes parameters, estimated both from data (LATS and other sources) and by other means (including informal ones), and includes statistical variability, both in the estimation of the parameters and in the operation of the processes. A model is dynamic and is likely to need updating. The processes by which the parameter estimates are obtained need to be specified, so that people can review how the model works, and so that the processes can be repeated. In general the parameter estimates will be full posterior distributions, not just point estimates.

The parameter estimates form a summary of (part of) the transport system (seen through the view of the model), so can be used as the basis of ‘best’ or ‘base’ estimates of actual demand for transport services. Models thus underlie the generation of synthetic data, whether through the imputation of missing (or otherwise unobtainable) observations, or through the estimation of measures that are not directly observable. However, the model can also be used actively to explore the way in which responses change as inputs or assumptions change, and thus to develop forecasts or predict the impact of policy or other changes, or to estimate values for unobserved combinations of inputs.

Thus a model in the database needs to include:

- The mathematical form of the model
- The parameter estimates that specify the current state of the model
- The functionality (processes and algorithms) needed to make forecasts or explore the impact of change on the model.

It is also necessary to be able to store datasets generated from a model (synthetic data). Such generated data may exist independently of an explicitly represented model, but where the model is available the dataset should be linked to it. The parameter estimates used for the generation of the data should also be stored with the data. There may be several versions of synthetic data addressing the same subject, using similar models, but with varying parameters or data resources used.

7 Appendix Group A: Supplementary Information and Explanations

Appendix 1: Glossary

The majority of terms in this list are taken from the Synthetic Estimation report, written by Miles Logie (see section 4.5).

Term	Meaning
Base matrix	A trip matrix forming the base from which forecasts are made.
Coefficients	Fitted (or otherwise estimated) point values that characterise a formula or equation. For example, in regression the fitted line has slope and intercept coefficients, these being the actual values estimated using the input data.
Confidence level	Value indicating the relative degree of certainty associated with an observation or prior data value matching an actual (underlying) process. In Statistics the term is used for the chance that an estimated range for the likely values of a parameter actually contain the real value.
Data Warehouse (DW)	A database designed for processing large volumes of static information in dynamic ways. A DW may fulfil the main requirements of the relational model, but the implementation will be quite different from a RDBMS.
Expansion factor	A factor (weight) corresponding to the Inverse of the sampled proportion used in a survey.
GLA	Greater London Authority
HH	Household
Imputation	The estimation of missing data values, using knowledge of underlying probability distribution functions (perhaps estimated from the existing data) and of the mechanism that causes the non-response.
Leg	Part of a trip using a single of mode travel.
LTS	London Transport Study
Metadata	Anything that you need to know to make proper and correct use of the real data, in terms of reading, processing, interpreting, analysing and presenting the information (see section 6.6)
Missing data	Data (single values or whole records) that should have been collected in a survey but were not, for whatever reason. The consequence of non-response (at the question or respondent level).
Modelling	Transport modelling, including ‘Four-stage modelling’, logit choice models.
Objective function	Mathematical equation containing variables, and their relative weights, to be maximised or minimised.
ODBC	Open DataBase Connectivity – a protocol for transferring information between relational database systems.
OLAP	On-Line Analytical Processing – the manipulation of structures of aggregate statistical information.

Term	Meaning
Parameters	Components of a model relationship or equation that represent general aspects of the underlying model, in contrast to the observed values of variables (which characterise particular individuals or objects). A simple example is given by the slope and intercept components of a regression line: in $y = \alpha + \beta x$, x and y are variables and α and β are parameters. An alternative usage is that after fitting a model and obtaining parameter estimates (or coefficients), the parameters are the components of the model that can be varied to produce different estimates (or forecasts) as outputs from the model.
Parameter Estimates	Estimates of the actual values taken by the parameters in a model. Estimates are obtained by some estimation process, which chooses the parameter values that optimise the fit of the model to observed data. The estimates may be single (point) values, but in general, because of uncertainly in the model and variability in observed data, are posterior distributions, showing the support or confidence associated with a range of possible parameter values. Parameter estimates are referred to as Coefficients in some contexts.
Part trip	A part of a trip whose defined start and end points may not correspond to the ultimate origin and destination of a trip.
Partial data	Data from an incomplete set of observations.
Posterior	Relates to situation after synthesis.
Prior	Relates to situation before synthesis.
RDBMS	Relational DataBase Management System
RSI	Roadside interview
Stage	A part of a trip from a set of parts which, when combined in sequence, represent a trip.
Synthesis	Estimating values through parameterised equations that may or may not relate to modelling.
T*f*L	Transport *for* London
Tour	Set of trips returning to original origin point, usually home.
Traveller surveys	Generic term for roadside and on-mode surveys.
Trip	Single travel movement between an origin and a destination zone to achieve a purpose.

Appendix 2: Some O-O Definitions

Object: An object is an element of interest in a system being modelled. Each object possesses some characteristics, performs some services, and exhibits some behaviour. Objects are the building blocks of models.

Class: A class is a grouping or categorization of objects with the same characteristics, services, and behaviours. A class is almost always defined as an extension of some other class, using the mechanism of 'inheritance'. The starting point in an O-O programming environment is the 'root class', which provides all the attributes and methods that must be possessed by every object in the system.

Inheritance: When a new class is based on an existing one it is called a subclass. A subclass inherits all the characteristics, services, and behaviours of the parent class and (through the parent) of any ancestral classes, tracing back to the root class. The parent of a subclass is called the super class. Inheritance means that only the things that are different have to be defined for the new class. It can customise what it inherits and/or provides more characteristics, services, or behaviours. Inheritance significantly facilitates reusability of earlier developed classes and decreases model development time.

For example, in dealing with transport we may define a class for a stage in a journey. This might have origin, destination, duration and mode variables. This stage class might have subclasses for different modes (where these need different characteristics), such as for walking, using own vehicle or using a fare-paying vehicle. The latter might have subclasses for Bus, Tube, Train and Taxi. These subclasses inherit the variables of their parents, so all have origin and destination, but the algorithm to compute the cost of the stage would be different for each one.

Instantiation: Creation of an object belonging to a class is called instantiation. The new object has all the characteristics (instance variables) and behaviours (instance methods) specified in the class from which it is instantiated. Instance variables of a class are created for each object instantiated as a member of that class. Instance methods are inherited, but no method code is replicated.

Variables: Characteristics (attributes) of an object are represented by variables. Instance variables, declared in a class, are used by the instance methods of that class and are created for each object instantiated as belonging to that class or any of its subclasses. Local variables are declared within a method for use only during the execution of that method. On completion of a method, all local variable values are lost.

Methods: Services provided and behaviours exhibited by an object are specified in methods. Two types of methods exist: class methods and instance methods. Class methods are used to provide services specific to a class. For example, each class, by inheriting from the root class, provides the method 'new' which creates an instance of a class. Instance methods, given in a class, are used to specify the services provided and behaviour exhibited for each object instantiated from that class. Each instance created as belonging to a class provides the services and behaviour specified in the instance methods of that class.

The method code is specified only once in the class and is not replicated on each instantiation of an object from that class. Therefore, model maintainability is significantly facilitated since a potential change is localized to only one method when hundreds of objects may exhibit the method behaviour.

Message Passing: All instantiated objects communicate with each other via message passing. Sending a message to an object implies the invocation of one of the receiving object's methods. Generally, it is said that "send message M to object A" as opposed to "send a message to object A to invoke its method M." Objects are identified by their unique addresses internally maintained by the system. They are called the object references. An object reference is used to specify the object that receives the message.

Polymorphism: Polymorphism refers to the ability of an object to assume more than one form, or for the same method to be invoked for different objects. For example, if an object that represents a journey stage is asked to return its cost, the asking object does not need to know mode was used for the stage, even though different modes calculate their cost in different ways.

A person object reference may refer to a patient, passenger, or customer depending on the logic at run time. Then sending a message, such as computeServiceTime, to the object pointed to by the person object reference would invoke a different algorithm (procedure) depending on the form of that object (patient, passenger, or customer) at run time. This means that an algorithm that operates on a heterogeneous set of objects does not need to consider the classes to which the objects belong – it simply sends a message to perform the desired action, and each object handles it as appropriate depending on its class.

Encapsulation: The methods of an object, specified in the object's class, describe the services provided and behaviours exhibited by that object. Any object belonging to a subclass of the object's class can access the attributes of the object directly. All other objects must request the object's service or trigger its behaviour by sending a message. How an object provides a service or exhibits a particular behaviour is completely hidden from the rest of the world. Only through message passing, can an object's service be requested.

Encapsulation means that the object hides its implementation from the caller objects requesting its services via message passing. Suppose that an object A wants to access the value of an instance variable V of object B. Due to encapsulation, object B's instance variable V, is hidden and cannot be directly accessed. Access is permitted only by having object B provide a method in which the value of instance variable V is returned. Thereafter, any object can send a message to object B and invoke the method that returns the value of its instance variable V.

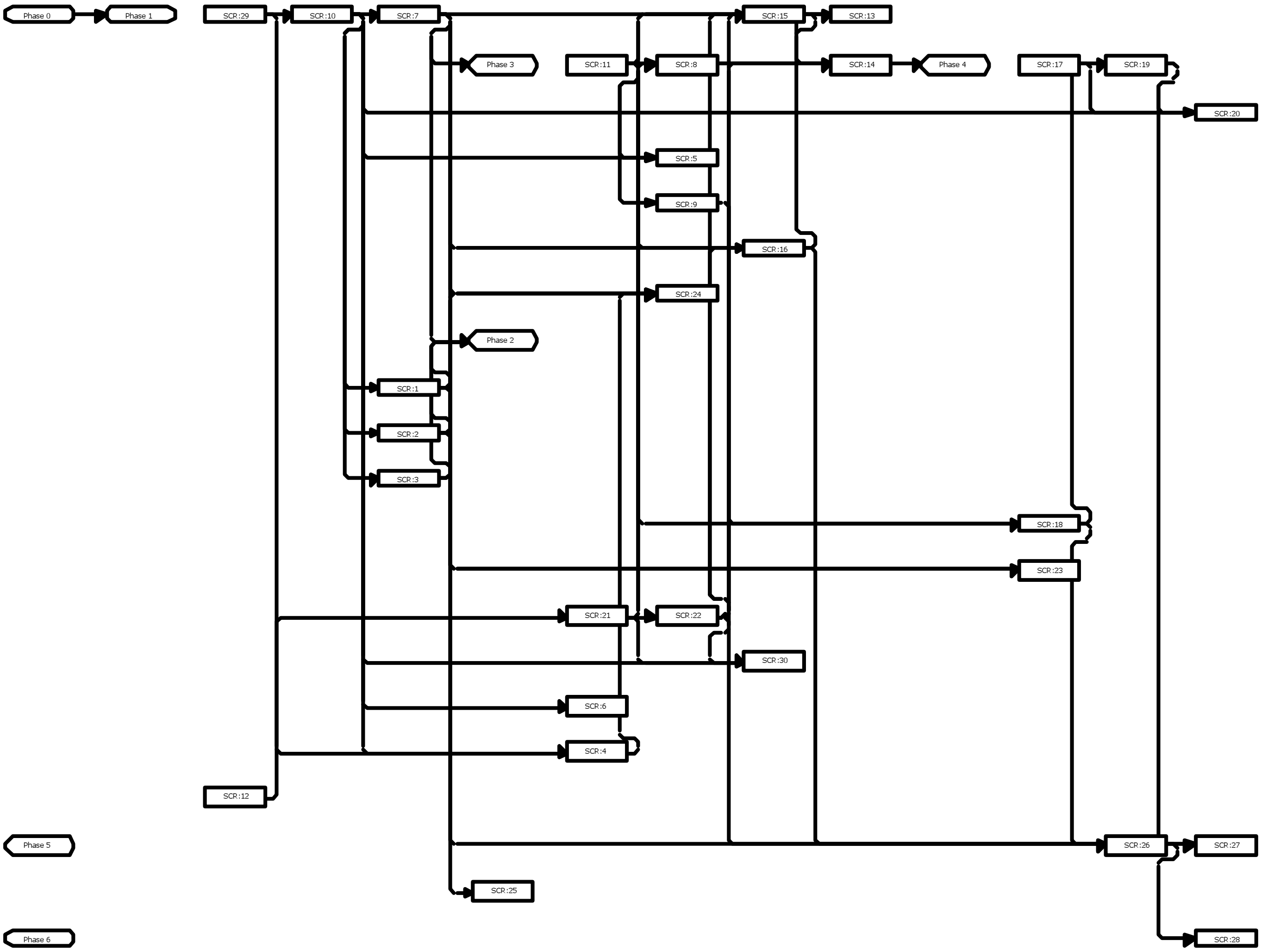
By insisting on (and enforcing) the rule that communication can only take place over an explicit interface, we generally avoid the problem of side effects (in which changing a data value has unforeseen effects in some other place). Interfaces still need to be well defined, but this definition has to be agreed and respected by both called and calling objects – there is no other way.

From encapsulation we can develop the idea of ‘components’, which are independent pieces of software that deliver services over a well defined interface using a standardised communication protocol. A component can be replaced with a new version that supports the same interface (but is perhaps more efficient, or offers more functionality), without affecting any of the users of the component.

Appendix 3: Pert Chart for Dependencies between Solution Components

The following diagram (Pert chart) attempts to show the dependencies between the Solution Components and Phases. All tasks have been given a standard duration, so the chart does not attempt to show any information about likely duration, only an outline of the sequencing that is needed, and this only at a rather general level.

Figure 5 Pert Chart for Solution Components



8 Appendix Group B: Comments from Software and Service Organisations

Various suppliers were invited to comment on the preliminary Database Outline version of the LATS Database Design Study report. Their comments are presented here. These comments have been taken into account when preparing this final version of the document.

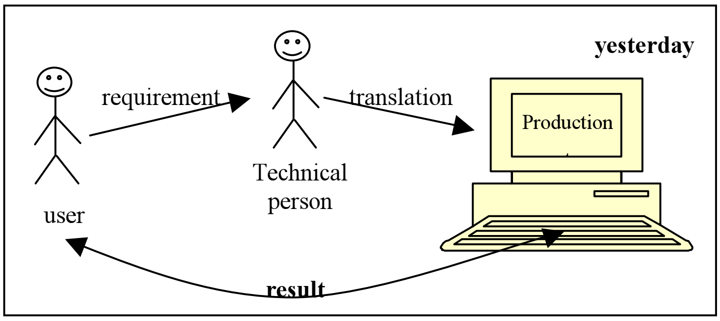
Appendix 4: Reinhard Karge, Run Software-Werkstatt GmbH

Koepenicker Strasse 325, 12555 Berlin, Germany, reinhard.karge@run-software.com, 30 April 2001

A4.1 Metadata perspective

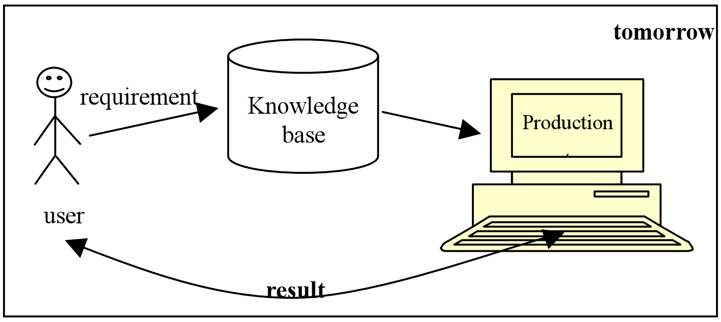
We have read the LATS Database Design Study with much interest. It sounds really ambitious but not unrealistic at all. Since our area is especially statistical metadata and knowledge bases we will make some comments from this perspective.

When providing a statistic with a well-defined number of output tables the simplest way is to create a team of experts that does the work. In this case metadata might be used to provide some background information. For creating a more flexible system you need either a team of experts that provide the required results or you store the expert knowledge in a knowledge database (enhanced metadata repository), which transforms the user's requests into production control information.



Statistical knowledgebase

In traditional statistics technical persons have translated statistical concepts for surveys or products into software specifications (programs or control information) that could be used for producing the required result. The same happened for specific external user requests, which are defined on a "conceptual" level and need to be translated into technical terms.



Newer statistical systems are going to replace the technical persons more and more by automatic procedures that translate standard requirements but also more and more specific requirements directly into process specifications. Such systems are more flexible and do not depend on specialists and experts in special software packages. This strategy requires, however, a statistical knowledge base or an enhanced metadata repository.

Hardware and software has been produced during the last years supporting this tendency. At the moment we cannot completely replace technical experts by knowledge-based systems but such systems can take over a number of tasks in a statistical environment.

Building a knowledge base means to structure statistical knowledge but also to enter the knowledge of statistical and technical experts in an enhanced metadata repository. Good experiences have been made in structuring statistical knowledge bases during the last years using terminology models.

Terminology Model

The conceptual metadata structure, which is the base for an enhanced metadata repository or knowledge base, is quite similar in statistical environments, but not the same. Concepts differ slightly and special features are required in different environments. We have made good experiences to reflect conceptual metadata models in form of terminology models, which are easy to handle by statistical experts and very useful for software developers. Hence, we would suggest creating a group of subject matter experts and metadata experts at the beginning of the project to define a terminology model for the LATS project. This is a good base for communication between the project partners but also for technical implementation, since the terminology model defines the communication items, the semantics for information exchange between users, users and processes and from process to process.

A4.2 Knowledge base solutions for statistical production

Statistics Switzerland is going to build a statistical knowledge base for the Census 2000, which could be considered as a step forward to building a knowledge-based statistical production system. More ambitious is the Swiss CODAM project, a metadata driven datawarehouse solution for statistics Switzerland. Both will show first results at the end of 2001.

We are involved in both projects in Statistics Switzerland. Moreover, we are involved in the METAWARE project from EuroStat. All these projects, but especially the Swiss ones (because they are intended for real statistical production), will provide a number of elements, which might be useful for the LATS project. All those projects are trying to put an essential amount of technical expert knowledge to the knowledge base or statistical metadata repository.

The LATS project could benefit from this development on the one side but it can also add some new aspects to current developments.

A4.3 Census 2000 and CODAM project in Switzerland

This is an excerpt from the datawarehouse concept of Statistics Switzerland. It gives a rough overview about the components planned for a metadata driven datawarehouse system, which will be implemented based on SuperCROSS for the tabulation part. The datawarehouse is based on an ORACLE database.

The structure is mainly designed for the Census 2000 in Switzerland. But it is more general and meets also the requirements for the Common Datawarehouse solution CODAM in Statistics Switzerland which will be build based on Bridge^{NA} and SuperSTAR,

The concept is based on the assumption that an information system for the end user is required as well as a system for production control, which are both based on a central metadata system (CMS), an enhanced metadata repository. Only when the CMS fulfils both requirements the consistency between the information system and the statistical output can be guaranteed.

The concept describes the system as a collection of required functions (components), which provide the requested functionality by interacting with each other. The components are not based on specific software products but describing the structure we had features of Bridge^{NA} and SuperSTAR in mind.

All request are sent to the central metadata system. This will guarantee:

1. A unique user interface
2. The consistency between data and metadata
3. A maximum of re-usability of metadata
4. Providing production system independent tools

Request specific user interfaces as WEB retrieval interfaces, production interfaces etc. will be provided, which will meet exactly the users requirements.

This is an overview about the position of the CMS in the statistical production and information system and its role for administrating and storing data. We suppose that data are stored in a datawarehouse or in a comparable system.

In such an environment a CMS would have to provide the following functions:

Documentation: The CMS contains, beside operational metadata, descriptive metadata for data on the level of classifications and its items, variables, tables, but also on cell level (e.g. footnotes). Descriptive metadata for different metadata objects can be combined to user-oriented documentation (e.g. as background information for a WEB based or online table or as explanatory text in paper presentations).

Production control: Production control is based on operational and physical metadata. Production control tools can "translates" conceptual definition into operational metadata. Thus, the production rules for a request like "provide a table with personal income by sex and age groups in five years from 1995 to 2000" could be completely generated. The specification of more difficult processes might be extended by specific information.

Production control is based on the principle that all existing data cubes in the datawarehouse are described in the CMS as well as all possible aggregations (virtual cubes). This allows combining data from virtual and real cubes as required by the user. Moreover, all data is linked to conceptual metadata information allowing correct interpretation of provided data.

Information retrieval: Conceptual metadata in the CMS allows supporting different retrieval functions using free text search and thesaurus-based keyword search as well as linked object techniques. Since retrieval requests are expressed mainly in terms of metadata the quality or conceptual metadata determines the quality of retrieval functions. User specific retrieval functions have to be developed in any case, but this is a minor task when having the metainformation in the CMS.

Customer administration: Requirements from customers (users) should be registered for being able to analyse user's requirements and to improve the services for different types of users. Moreover, it can be used as an input for accounting systems and security check.

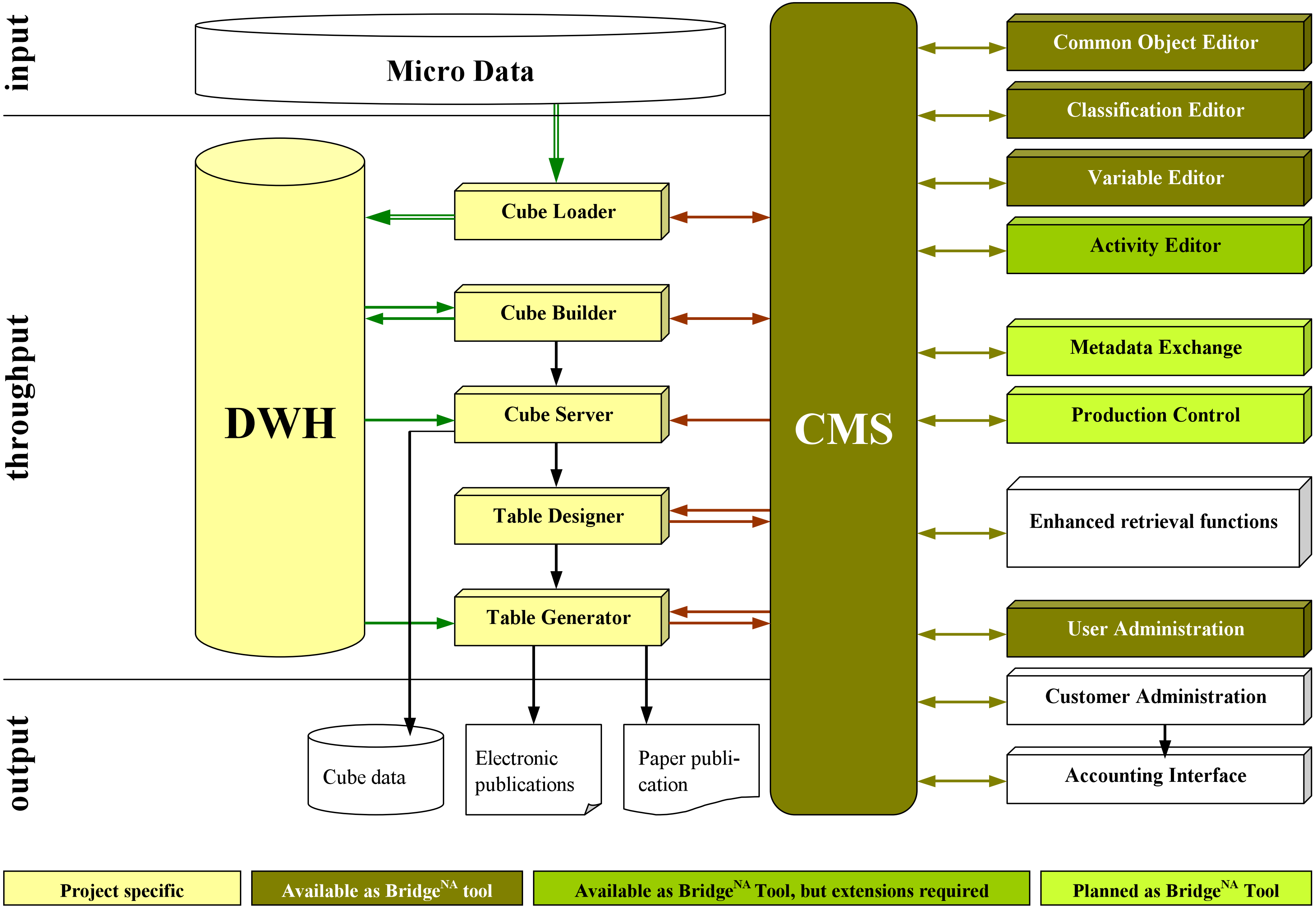
The following picture gives a rough overview about the required components in a metadata driven statistical production system and the interface between the datawarehouse (DWH) and the CMS. This overview is not complete but it shows what is available and what will be developed in the near future. Other standard components as well as project specific features in a certain environment will extend the system.

Since we are dealing with the metadata part the picture is focused to the metadata system (CMS).

A4.4 The METAWARE project

The METAWARE project is studying and implementing requirements on metadata driven statistical datawarehouse systems. The METAWARE specifications are more general than the Swiss datawarehouse solution. Nevertheless, there are a lot of similarities and both approached differ not so much in principle but more in features, which are supported. One interesting part in the METAWARE project is flexible query mechanisms on the datawarehouse via the WEB.

Another aspect it the METAWARE project is not considering a special datawarehouse solution but providing more general solutions that will be demonstrated by datawarehouse systems based on ORACLE and Microsoft SQL server.


input
throughput
output
Micro Data
DWH
Cube Loader
Cube Builder
Cube Server
Table Designer
Table Generator
CMS
Common Object Editor
Classification Editor
Variable Editor
Activity Editor
Metadata Exchange
Production Control
Enhanced retrieval functions
User Administration
Customer Administration
Accounting Interface
Cube data
Electronic publications
Paper publi-cation
Project specific
Available as Bridge^{NA} tool
Available as Bridge^{NA} Tool, but extensions required
Planned as Bridge^{NA} Tool

A4.5 Knowledge based solution for LATS

For the LATS project a dynamic solution seems to be a preferable solution, since the LATS project will run for many years (as long as public transport has to be organized). Several requirements will change over the years and new ones might be defined, which means the project specifications have to be updated as well.

A knowledge base solution for LATS has the advantage that rules for storing data, building synthetic data and query mechanisms can be defined on a conceptual level. This is a dynamic approach and can be developed with changing requirements. Production systems may change or can be replaced by more appropriate ones without affecting the concepts and rules the project is based on. This guarantees flexibility for further conceptual development as well as flexibility for choosing the most appropriate production systems for different purposes.

Appendix 5: Lawrence A Hopkins, Branch Manager, Tessella Support Services plc

4 Wellington Court, Wellington Street, Cambridge CB1 1HZ, www.tessella.com

A5.1 Comments on the LATS Database Design Study

1. The document appears to be quite thorough and correct in what it says. It is, as the author quite clearly states, neither a requirements specification nor a design document. The document is quite broad in its scope, touching on some user requirements, some system components (design), and available technologies (implementation). The document could almost be described as a repository for ideas that the authors wish to ‘keep warm’.
2. The discussion of the user requirements of the system is fine, bearing in mind that it is not intended to be anything as formal as a User Requirements Document. There is nothing in the requirements that gives rise to any concern.
3. Relational databases will almost certainly play some role in the core of the system. The need for diverse data (e.g. data from questionnaires of different formats) can be stored by having separate tables for each questionnaire, and ‘views’ and stored programs can be used to act on the information common to different questionnaires in a unified fashion.
4. Object databases are mentioned in the report, although it is not clear to us how they would be of great use to the system. It is quite possible that the authors have envisaged how they would fit in, but their applicability has not been described in the document.
5. The term ‘version control’ is used in the document to describe both audit trails and logs of a user session’s analyses performed. ‘Version control’ may not be the most accurate name for it, but it is certainly an established technology and an appropriate choice.
6. The Workbench Paradigm is a very sensible approach, which boils down to storing many pieces of data to do with users’ sessions.
7. We consider the Thesaurus-based search techniques to be an excellent way of taming the unstructured metadata that can occur, and transcending differences in terminology.
8. The discussion of XML is a bit out of place in this document – it is more of a design and implementation issue, and the choice of technology will be subordinate to system requirements shown up at the end of the requirements phase.

Appendix 6: Warren Richter, Chief Technology Officer, Space-Time Research

1102 Toorak Rd, Hartwell, Melbourne, Victoria 3124, Australia

A6.1 Some 'non-technical' considerations

LATS is an ambitious project which will break new ground 'mixing and matching' technologies and methodologies to suit information/analytical requirements. Although there are going to be some significant technical challenges, I agree with you that solutions either exist or are being developed. In any case, it has been my experience that most technical problems can be solved, or workarounds found. In my view, one single critical success factor dominates all others in a complex, long term project - managing the perceptions of end users and the 'owners' of the project. I appreciate that the LATS Database Outline document is not the place to cover project and client management issues in any detail, and you may already have identified and considered these things separately, but can I make a couple of suggestions for your consideration that address the issue of perception management in the context of this particular project.

1.1 Managing expectations

I believe you have been realistic in establishing the broad timetables, in setting the broad objectives, and in identifying the sequence of development of the methodologies and systems. However, the Design Report will need to be quite clear about the availability of capability and I suggest it include a section for 'users and owners' that describes 'what you will be able to do and when'. This would be couched in terms of 'by XXXX users will be able to define their own recodes and variables to be stored and applied to the underlying data on request or incorporated in the data if they have the access privileges to do so'. 'By XXXX, users will be able to define the mathematical form of a model, store and describe the model in the metadata management system (in fact they will be required to provide appropriate metadata) apply the model, store the results, and make them available to nominated users'. Needless to say, the descriptions of what capability will be available should be exciting, but the timetables should be very conservative!

1.2 Love or hate at first sight

It has been my unpleasant experience that despite everything the project team says about prototypes or early releases, user's perceptions and their support for the project seem to be permanently shaped by their first experience with the system. This simply reinforces your view of the importance of the interface but I would add a couple of suggestions. They will be forgiving if the system does not deliver all the capability they require (particularly if you have described what will be able to be done and when) but there are some basic things that they will reasonably expect to see in the LATS environment such as basic cross tabulation and recoding, data retrieval and saving in various formats. Obviously, the basic interface must be excellent. The point is that they must be impressed with the potential of LATS the first time they see it, and I suggest there ought to be a rigorous set of criteria applied to trigger the first release, or even a pre-release to users.

1.3 Contractor management

I agree that the project is going to involve several contractors because no single organisation will be able to deliver the required functionality or expertise (see below for comments on architecture and functionality). In these circumstances it is always tempting for the project authority to seek the services of a prime contractor, for obvious reasons. I think this can be made to work but it is fraught with difficulties and I have never been keen on this approach in complex development projects where sharing intellectual property is required. With LATS in particular, I believe the key to success will be interaction and idea sharing among the best in the business, building on very sound base infrastructure which provides key capability from day one. I think the way to make this happen is to have these contractors working in a team environment (signing non-disclosure agreements would be a pre-condition for team membership), and working to a 'LATS Team Leader' who is either not a contractor or at least a contractor from outside industry (such as your-self). This way, you have everyone working to a higher purpose (ie the success of LATS rather than the success of the prime contractor) and their and their companies' contribution being highly visible to the project authority. It also allows for particular management responsibilities for joint functions to be allocated to an individual contractor. For example, with the CODAM project in Switzerland (more about that later), my company has been given responsibility for, inter alia, managing the development of the interfaces between Bridge and our software (SuperSTAR) and for managing and specifying the enhancement path for Bridge. However, the developers of Bridge (Run Software) are contracted to the Swiss Office of Federal Statistics (OFS). In this way, the OFS' management task is simplified but the benefits of a team approach and direct contractual arrangements is maintained.

A6.2 Technological, informational, and statistical considerations

LATS is ambitious, but I do not believe it to be risky provided that sound, base infrastructure is identified and selected as early as possible. As you have correctly stated in the Database Outline document, the solution will have to be a hybrid and I would like to spend a little time on what might be the key elements and considerations which dictate what the hybrid components should be.

There is no doubt in my mind that a RDBMS will be required for basic data manipulation and storage and to take advantage of the 'generic' features of RDBMSs such as allowing a variety of data structures, powerful programming tools and utilities etc. Although everything can be done in a RDBMS if you have time, lots of money for CPU cycles and extremely good and expensive programming skills, it is also likely that LATS will want to take advantage of the advanced statistical functionality available in packages such as SAS and SPSS. As you have observed, advanced statistical agencies have all found it necessary to have a mix of software packages on a 'horses for courses' basis ie a RDBMS, SAS, SPSS or other statistical packages, and an advanced online analytical/cross tabulation package such as SuperSTAR. Many of them are now also implementing advanced metadata management systems. The Australian Bureau of Statistics and Statistics New Zealand have very good (in fact I think they are the best in the world from a usability/practicality perspective - but I am biased as I was responsible for the ABS systems) Lotus Notes-based metadata management systems although with some technical limitations that would make them difficult to apply to LATS. However, as you know, the foremost 'thinkers' in the metadata management area are the Netherlands, the Scandinavian agencies and the Swiss, and these agencies are driving the development of Bridge. So, we have the best agencies in the world voting with their feet in terms of statistical infrastructure. The infrastructure they have voted for and the reasons for it are:

2.1 Central, unifying, active, metadata management system

The aim is to make all statistics visible, accessible, understandable and relatable. This in turn will help achieve the holy grail - statistical integration (ie optimising the concepts, sources and methods such that the most efficient set of statistics and statistical collections are in place). However, there are two other very practical considerations driving the adoption of central metadata management systems. First, it’s the only way agencies can fulfil their duty of care to deliver knowledge and understanding to their clients in a 'self-service' Internet dissemination regime (and they are all going down that path - as it seems will LATS). Second, dynamic trade, faster economic cycles, and the availability of new sources of data such as retailers point-of-sale data and administrative by-product data are forcing new, complex methodologies (virtual collections) on agencies and they must have central metadata repositories to describe and manage these new processes. This will include active metadata to define and control the application of models a la LATS, although I don't know of any agency that has implemented this as yet. Nevertheless, I suggest the obvious choice for the LATS metadata system is Bridge. Having said that, you should be aware that Space-Time Research has a very close relationship with Run Software.

2.2 Online analytical processing and Internet dissemination (data and metadata access) software.

This is where I run into the temptation to plug SuperSTAR but I will try to avoid that. Let me just say that all the advanced agencies have recognised that RDBMS and statistical packages such as SPSS and SAS cannot provide the easy-to-use interface, cross tabulation and multi-dimensional analytical capabilities of products like SuperSTAR. Nor can they easily support extremely complex relationships among variables and entities that need to be accommodated in advanced reporting and analytical applications. The classic examples of these are (1) the need to avoid double-counting (or miss-counting) when there are multiple dependencies among persons, households and 'episodes' associated with these entities; (2) the sheer cross-tabulation and multi-dimensional 'grunt' needed to support online analysis and/or Internet-based self-service reporting and manipulation of complex datasets; and (3) requirements for 'linked analyses' where records with certain attributes need to be linked dynamically and easily (without programming) and then analysed or cross tabulated (eg find all people who have had a heart attack in the last 10 years AND who have had the following kinds of treatment or pathology in the two years prior to the heart attack, AND/OR the two years after the heart attack, report on treatment X type X date X hospital X kind of outcome, list unit records).

2.3 Statistical packages for processing, modelling and analysis

Just to complete the infrastructure picture; agencies also need SAS etc for these applications although I have noticed that there is a distinct move towards reducing the use of SAS etc, towards RDBMS combined with packages like SuperSTAR. I believe this has a lot to do with downsizing, skills shortages and the increasing complexity of statistical methodologies. This latter point is I think, forcing agencies towards 'centres of excellence' approaches for quality assurance purposes rather than the 'everyone can write (and make mistakes in) SAS and SQL' approach.

2.4 'Informational' considerations

This overlaps to some extent with my comments under 2.1 but is worth covering as a separate point. Although there will be a limited number of datasets in LATS initially, the numbers will grow over time. Moreover, I expect the LATS project will want to influence the design of the surveys and administrative sources generating these datasets with the aim of optimising the total set of information for decision-making and analyses. If this is the case, it reinforces the needs for an advanced metadata management system. It also suggests that capturing the metadata describing current data, and establishing some kind of information management regime across the organisations generating data for LATS is a matter of urgency. The reason is that design decisions affecting the flow of information for a long time into the future are being made now and ideally, the decisions should be informed about what is needed to maximise relatability. In an ideal world, the designers of future information sources relevant to LATS would be asked to say why they could not use existing information concepts (ie variables and classifications) and appropriate methodologies before they commit to a design, and they need a metadata management system to be able to do this. I appreciate that this is not easy, but a London transport information 'Czar' could be worthwhile creating.

2.5 Standards and information creation

I think you have identified the key standards that might be relevant to LATS. Unfortunately, I expect you will have to pick some winners, as there is still a great deal of Microsoft vs the rest in the standards game. We are keeping a close eye on the following: Microsoft's XML for analysis standard for definition of structural metadata (but I don't know if it is a winner); SOAP, Microsoft's OLE DB for OLAP seems to have some market support but I would not suggest that it is worth making adherence to this a pre-condition for LATS at this stage. Similarly Sun's JOLAP (Jave-based solution for a standard OLAP interface) is relevant but I wouldn't put a lot of money on it.

On balance however, I don't believe these particular standards will address the key issues for LATS, which really revolve around managing the flow of data, and managing the information (dataset) creation process rather than mixing and matching interfaces and databases. The reason for this is that products like SuperSTAR, SAS and SPSS rely on their underlying data structures to deliver the capability (functionality and performance) users require. In other words, we could run queries on a RDBMS using the SuperSTAR interface but the underlying limitations of the RDBMS' data structure means that performance would be dreadful unless hardware and CPU cycles were thrown at it. Similarly, some of the things RDBMS' are good at are hard to do on a SuperSTAR database. In a nutshell, its horses for courses at this stage, but I do believe there are some prospects in the future of being able to formulate a query and having it passed to the most appropriate database and database engine. However, I think LATS will be complicated enough without requiring this kind of bleeding edge technology.

In regard to XML, the problem seems to me to be that it while it is self-describing it is unfortunately not self-disciplined. We use it internally in SuperSTAR and we, and other vendors would have little difficulty in conforming to an XML standard but I can't be more precise than that at this stage. I believe that there are moves afoot to have a XML for GESMES (the EDIFACT generalised statistical message standard) but I believe this is a way off. I have never been keen on GESMES anyway - it is extremely complicated, and I don't see it being taken up in any significant way.

2.6 Specifics

As I have previously indicated, you have covered the issues very well. The following are a grab bag of ideas and comments.

2.6.1 Operations Management

I indicated in 2.5 that a key issue will be the managing the flow of data and metadata and the information creation (and description) process. This suggests that you should specify that vendors should be able to provide good production management processes and perhaps, should be open to a workflow management regime. However, I do recommend that things be kept as simple as possible for as long as possible and this would probably preclude adopting a workflow management system initially.

2.6.2 Architecture

I recommend a central, unifying, active metadata management system which traps all variables and classifications as they are created, which 'forces' metadata confrontation (ie why don't you use this classification?) and which feeds metadata to the analytical interface(s). Ultimately, the metadata management system should evolve towards a knowledge server which acts as both a ‘navigator’ and an ‘explainer’. The metadata management system would also be the repository and catalogue of relevant objects including models. That said, I don't believe it would be worth emulating binary object cataloguing in the metadata management system and if this is important you might consider linking the metadata system to a capable document management system. Note also my comments on the very important 'informational' aspects which are more managerial than architectural but which could have a profound long-term effect.

Any online analytical processing/cross tabulation package must have the capability of being tightly coupled with data systems such as RDBMS so that data can be mirrored automatically, and passed back quickly. This also applies to any statistical packages.

You have mentioned a 'history' mechanism and this could probably be achieved in the RDBMS by using rollback facilities, but this could be very tricky and wasteful of disk space and CPU cycles if it is applied at the micro-data level. An alternative could be to use the OLMAP (online micro analytical processing) features of SuperSTAR, which can quickly mirror RDBMSs and compress data 'naturally' (ie it is usable in its compressed form), to form an 'active archive' of data - particularly the larger sets of micro-data. This means data can be transferred directly from SuperSTAR as required.

2.6.3 Single unifying interface

Yes, I think this is a terrific idea (and we have a great candidate for it!) but there are limitations imposed if it is to be browser based. We are just completing the development of a product called SuperWEB which I believe to be the best of its kind, but we have been forced to compromise on some functionality in the short term because our clients wanted it to be standard-browser based (ie no plug-ins). If you want excellence in this area, it may be necessary to accept that your users will need plug-ins so that the interface developers can have a free hand. It will be important to establish this up front.

2.6.4 Precision

Is this to be applied at the cell or higher levels? Obviously some measures of precision will be needed at the dataset level and at for some lower levels but cell-level precision indicators imply 'shadow tables' or 'shadow database fields' or dynamic calculation or a mix of all three. We (and I expect others) can support this but it needs to be precisely defined if the capability is needed in the first release.

Finally, I mentioned the Swiss CODAM project above. Although the orientation of CODAM is different to LATS (it is a prototype of a corporate metadata management system with a closely coupled output database/data warehouse) there are many similarities, and it will be breaking new ground for a statistical agency. The project has a hybrid architecture with SuperSTAR being installed first (with tight interfaces between it and Bridge) and then an Oracle database, coupled to SuperSTAR. In this way the Swiss will have all the advantages of a very easy to use interface with well accepted statistical/reporting functionality and performance characteristics via SuperSTAR and they can then mix and match functionality between Oracle and SuperSTAR using the SuperSTAR as the model for interfaces and dissemination/reporting. What is particularly nice about this is that they know it works well from day 1 and they can comfortably add functionality to either Oracle or SuperSTAR with minimum risk. SuperSTAR will automatically write data to Oracle (when we have designed the Oracle db) so Oracle also acts as a safety first mechanism (ie I'm confident STR will be around for a long time but I suggested to the Swiss that it would not be wise to have the national statistical treasure locked into a proprietary database).

Appendix 7: Simon Musgrave, Project Leader, Faster, UK Data Archive

University of Essex, Colchester CO4 3SQ, www.faster-data.org

I certainly enjoyed reading your study paper. I am sorry not be able to do more than provide a few unstructured and hurried comments as they occurred to me. I like much of the thinking and the philosophical concepts (as I am sure you appreciate). In particular the workbench concept is important and the implication that you use some consistent infrastructure protocols (SOAP or the like) to bring resources together in an operational way is interesting. I note the point about the history mechanism. Some interesting challenges for DDI type structures.

I enjoyed the section on synthetic databases, I guess that the Dutch have done most in a practical sense to use these techniques, certainly I am not aware of ONS work in this area and so you could be a UK leader in this field. I need to brush up my understanding of the statistical issues. This makes the point the your paper does a nice job of bringing together the database and statistical methodologies. I am sure you are right to see this as a separate contract. There are not many people who bridge this gap, particularly in an organisational sense.

It might be worth clarifying the level of dynamic disclosure control. Are you talking of simple suppression of low cell counts or techniques to randomise etc. Is this likely to be important given access restrictions to 'trusted' users. There may be some pay-offs here.

It does sound like an ambitious but feasible project, especially if the underlying infrastructure is correct. The use of standards is key and can partly be sold as providing access to external sources as well as internal one. I agree about the need to rely on more than one technology.