OPUS

Optimising the use of Partial information in Urban and regional Systems

Project IST-2001-32471

WP3: Development of Meta Data Methods

Title : **Proposals for Metadata for Generic Support of Statistical Modelling in Statistical Databases**

Creator (Author): Andrew Westlake Survey & Statistical Computing

Contributor :

Identifier :	Deliverable D3.1 of project IST-2001-32471
Status :	Draft version of programme deliverable
Type :	Report (peer-reviewed deliverable)
Version :	1.4 – Complete draft with revisions after discussion
Date.Created :	20 May 2004
Date.Modified :	20 April 2005
Date.Next version due :	June 2005
Submission Date :	
Subject.Category	
Subject.Keyword	
Source	
Relation	This header section draws on the e-GMS structure for document metadata developed by the UK e-Gov initiative.
Rights.Copyright	The Opus Project
Contract Date :	April 2003
Publisher (Project Coordinator) :	Imperial College London

Contact Person :	John Polak
Address :	Centre for Transport Studies Department of Civil and Environmental Engineering Imperial College London South Kensington campus London SW7 2AZ United Kingdom
Telephone :	+44-(0)20-7594.6089
Fax :	+44-(0)20-7594.6102
e-mail :	j.polak@imperial.ac.uk
Consortium :	CTS, TFL, KATALYSIS, ETHZ, FUNDP, PTV, SYSTEMATICA, WHO. MINNERVA, SURVEY & STATISTICAL COMPUTING, OXFORD SYSTEMATICS

TABLE OF CONTENTS

TECHNICAL ABSTRACT

This deliverable D3.1 is a result of Work Package WP03 of the OPUS project. Work Package WP03 has as title: "Development of Meta Data Methods". The specification of this Work Package from the project proposal is as follows.

Objectives

To extend current initiatives on statistical metadata (particularly metadata for statistical processes) to include the representation of statistical models, both in their underlying structure and as used for particular analyses or data syntheses.

In particular, to provide a representation of algebraic model forms as an extension of existing expression representations, records of input data sources (including their versions), parametric assumptions used in model invocations (including distribution (Bayesian) assumptions for parameters) and links to synthesised information from model invocations. It will also draw on work for generic version control and audit trails.

Description of work

The work package will draw on existing (and ongoing) work by others on statistical metadata (particularly on process metadata), and on the related implementation activities for metadata within the LATS transport database project being undertaken in London. It will also draw on concurrent work in WP2 to identify the form of models to be considered. The work will comprise three interrelated activities.

1. *The development of a representation of the underlying form of statistical models as metadata, in a way that is accessible for review by people and execution by software.*
   This will include algebraic expressions and relationships, distributions, variables and parameters, and will not assume that the model can be represented as a single component.
2. *The development of a representation of the way in which a model is used in the context of a statistical database.*
   This will include recording the input information sources used, together with prior settings or assumptions about parameters (including vague assumptions in the form of distributions and dependencies). The representation should allow the use of a model to be reviewed, revised and re-executed. Issues of version control will need to be addressed, since in general the input sources will be dynamically updated.
3. *The development of a representation of the results of using a model.*
   This will include estimated parameter values (with suitable posterior support or precision information), and also synthesised information generated from the fitted model. The main body of the latter will be stored in standard structures (as for real data), but it is essential to retain the link to the model and the generation process. Issues of version control will arise again.

In this report we discuss general issues about the nature of statistical modelling and about statistical metadata. This leads to an analysis of the conceptual structure of statistical analysis (of the form covered by the Opus project).

We conclude that there are five essential elements for a statistical model. These are

1. Variables, which may or may not be observable, and for which we may or may not have any data.
2. Parameters (characteristics of the underlying system), chosen because we are interested in information about them, or because they are needed for our formulation of the system.
3. Mathematical relationships (of various forms) between the underlying constructs (variables and parameters) of the model, with parameters for the detailed specification of the relationships.
4. Probability distributions (from various families) for variables and parameters, with parameters and interdependencies.
5. Information (prior knowledge) about the parameters in the model (for both the relationships and the distributions).

From this analysis we have developed a proposal for a (metadata) structure for the representation of statistical modelling (including the model calibration process). This conceptual structure is presented as a model using UML.

EXECUTIVE SUMMARY

This document is Deliverable D3.1 of the Fifth-Framework project (FP5) OPUS. The OPUS project aims to develop and demonstrate statistically sound methods of combining datasets that each provide partial information on a single complex of underlying variables.

The expected practical result of application of the OPUS methodology is a calibrated probabilistic model of the problem domain at hand, with which it is possible to calculate the most likely values of missing, unobserved, or unobservable quantities of the object system under study, with potentially important savings of time and resources.

In this report we discuss general issues about the nature of statistical modelling and about statistical metadata. This leads to an analysis of the conceptual structure of statistical analysis (of the form covered by the Opus project), and hence to a proposal for a (metadata) structure for the representation of statistical models (including the model calibration process). This structure is presented as a conceptual model using UML.

1. INTRODUCTION AND FRAMEWORK

1.1 About OPUS

1.1.1 Background

OPUS is a large information management research project, supported by Eurostat as part of the European Commission's Information Society Technologies (IST) Programme. The overall aim of the OPUS project is to enable the coherent combination and use of data from disparate, cross-sectoral sources, and so contribute to improved decision making in the public and private sector within Europe. The research is focused on developing an innovative methodology, incorporating statistical and database systems. Transport planning is a prominent example of a topic that uses multiple sources of data, and will be the main test case for OPUS, but the cross-sectoral nature of the research will be demonstrated through the inclusion of an application in the field of health information as another example.

To meet the needs for comprehensive information on socio-economic systems such as urban and regional transport planning, and in the health services sector, data from diverse sources (e.g. conventional sample surveys, census records, operational data streams and data generated by IST systems themselves) must be combined. There is currently no appropriate developed methodology that enables the combination of complex spatial, temporal and real time data in a statistically coherent fashion. The aim of the project is to develop, apply and evaluate such a methodology. OPUS will develop a general statistical framework for combining diverse data sources and specialise this framework to estimate indicators of mobility such as travel patterns over space and time for different groups of people. The project will undertake pilot and feasibility study applications in London, Zurich, Milan, and on a national level in Belgium. Methods for extending the framework to information aspects of the health domain will also be investigated.

The benefits of OPUS will be:

- Improved estimation of detailed travel demand, using all available information;
- Avoidance of simplified combination of data that can give erroneous estimates;
- Indicators of data quality, to provide guidance for new data collection;
- A framework for managing data from rolling survey programmes;
- Better understanding of the role of variability and uncertainty in results and models;
- Avoidance of confusion from different, apparently conflicting, estimates of the same quantity;
- A generalised methodology for other domains of interest.

The participants in the OPUS project are as follows:

Research Organisations

- CTS (Centre for Transport Studies, Department of Civil and Environmental Engineering, Imperial College London), United Kingdom – Lead Partner
- DEPH (Department of Epidemiology and Public Health, Imperial College London), United Kingdom
- ETHZ (Institut für Verkehrsplanung, Transporttechnik, Strassen- und Eisenbahnbau), Switzerland
- FUNDP, Transport Research Group (Facultés Universitaires Notre-Dame de la Paix), Belgium

Practitioners

- Minnerva Ltd., United Kingdom.
- Survey and Statistical Computing, United Kingdom.
- Katalysis Ltd., United Kingdom.
- PTV AG, Germany
- Systematica, Italy.
- Oxford Systematics, Australia: Peer Reviewer

Public Bodies

- Transport for London (TfL), United Kingdom.
- World Health Organisation (WHO), Italy.

1.1.2 Objectives of the OPUS project

To meet the needs for comprehensive information on socio-economic systems such as urban and regional transport planning, and in the health services sector, data from diverse sources (e.g. conventional sample surveys, census records, operational data streams and data generated by IST systems themselves) must be *combined*. There is currently no appropriate developed methodology that enables the combination of complex spatial, temporal and real time data in a statistically coherent fashion.

The overall aim of the proposed project is to develop, apply and evaluate such methodologies, taking as a specific case study the transport planning sector. The specific objectives of the study are:

- To develop a generic statistical framework to enable the optimal combination of complex spatial and temporal data from survey and non-survey sources. This framework will specify how to optimally estimate the underlying population parameters of interest taking into account the structural relationships between the different measured data quantities and the sampling and non-sampling errors associated with the respective data collection processes. It is envisaged that the framework will be broadly Bayesian in nature. The framework will make no spe-

cific assumptions regarding the particular structural and sampling/non-sampling errors and will thus be relevant to a wide range of application domains.

- To apply the generic framework within the field of urban and regional transport planning. This will involve the definition of specific structural relationships amongst measured quantities and the characterisation of sampling/non-sampling errors, based on domain knowledge from the field of transport planning.
- To develop the necessary database and estimation software to enable the application of the statistical framework in a number of case study areas.
- To undertake a major pilot application study in London, focusing on the derivation of indicators of the mobility and the performance of transport policy measures.
- In parallel, to investigate the feasibility of applying the framework and methodologies developed both in other transport planning contexts and in other proximate domains, specifically environmental management and social statistics.
- Based on the experience gained in the pilot application and the feasibility studies, to evaluate the performance of the proposed methods and to define the scope and approach for wider applications in relevant domains including environmental management and health care.
- To disseminate the results to the relevant academic and practitioner communities.

1.1.3 Motivation

OPUS addresses the situation in which the analyst must combine data from a variety of different data sources to obtain a best estimate, or a fuller understanding, of a system. Such a situation can arise for a number of reasons including:

- No single source contains sufficient information by itself; or
- Multiple sources naturally arise (e.g. through observations at different levels of spatial or temporal aggregation or by means of different survey methods), resulting in a need to reconcile potentially conflicting estimations; or
- The need to update or transfer an existing set of data and parameter estimates when additional information becomes available.

Problems of combining data from different sources to produce consistent estimates of underlying population parameters arise in many fields of study including environmental monitoring, epidemiology and public health, earth observation, geographic information and navigation systems, transport and logistics, and economic and social statistics. Although the risks of using *ad hoc* combination rules and procedures are well understood, there are nevertheless many examples from practice in which just such approaches are still used. This reflects the fact that, although relatively straightforward methods exist for simple cases, there does not exist a coherent and well developed set of applicable methods capable of dealing with the full range of data combination problems, including factors such as:

- Data sources that provide both direct and indirect information on the relevant population parameters

- Data that are presented at different levels of aggregation
- Data sources with differing levels of statistical precision or user confidence
- Data that overlap, but that may provide different or conflicting information
- Gaps in the data observations
- The issues raised by the aging of sample survey data and the consequent need for updating
- Accommodating the updating sources
- The effect of sampling and non-sampling errors (including survey non-response and other sources of missing data)
- The opportunities presented by new data streams from IST systems

The key scientific objective of the project is to develop a generic statistical framework for the optimal combination of complex spatial and temporal data from survey and non-survey sources. The framework will be sufficiently abstract to be applicable to a wide range of potential domains.

Associated with this overall objective is the need for a suitable representation of the statistical metadata that is used for the specification and application of such a framework. That is the immediate objective of this report.

1.1.4 Subject areas

OPUS provides a generic approach but, in each case, it is necessary to make this approach specific to the particular area of interest (whether the area is geographical or topical in nature). A particular test-bed is transport in London, but studies will be made for transport in Belgium, Switzerland, and Italy, as well as health studies.

1.2 OPUS Project Work Package WP3

This section summarises the specifications for this work package.

1.2.1 Objectives

To extend current initiatives on statistical metadata (particularly metadata for statistical processes) to include the representation of statistical models, both in their underlying structure and as used for particular analyses or data syntheses.

In particular, to provide

- a representation of algebraic model forms as an extension of existing expression representations,
- records of input data sources (including their versions),
- parametric assumptions used in model invocations (including distribution (Bayesian) assumptions for parameters) and
- links to synthesised information from model invocations.

It will also draw on work for generic version control and audit trails.

1.2.2 Description of work

The work package will draw on existing (and ongoing) work by others on statistical metadata (particularly on process metadata), and on the related implementation activities for metadata within the LATS transport database project being undertaken in London. It will also draw on concurrent work in WP2 to identify the form of models to be considered. The work will comprise three interrelated activities.

1. The development of a representation of the underlying form of statistical models as metadata, in a way that is accessible for review by people and execution by software.
   This will include algebraic expressions and relationships, distributions, variables and parameters, and will not assume that the model can be represented as a single component.
2. The development of a representation of the way in which a model is used in the context of a statistical database.
   This will include recording the input information sources used, together with prior settings or assumptions about parameters (including vague assumptions in the form of distributions and dependencies). The representation should allow the use of a model to be reviewed, revised and re-executed. Issues of version control will need to be addressed, since in general the input sources will be dynamically updated.
3. The development of a representation of the results of using a model.
   This will include estimated parameter values (with suitable posterior support or precision information), and also synthesised information generated from the fitted model. The main body of the latter will be stored in standard structures (as for real data), but it is essential to retain the link to the model and the generation process. Issues of version control will arise again.

1.2.3 Deliverables

D3.1 Proposals for Metadata for Generic Support of Statistical Modelling in Statistical Databases

D3.2 Specifications for the Extension of the LATS Database System for the Transport Domain

1.3 Objectives of Deliverable D3.1

The key scientific objective of the Opus project is to develop a generic statistical framework for the optimal combination of complex spatial and temporal data from survey and non-survey sources. The framework will be sufficiently abstract to be applicable to a wide range of potential domains.

Associated with this objective is the need for a suitable representation of the statistical metadata that is used for the specification and application of such a framework. That is the immediate objective of this document.

The specific objectives of this deliverable are to:

1. Analyse the generic requirements for the representation of statistical models (and associated specification and fitting processes) of the general type applicable to the Opus project.
2. Investigate existing systems and proposals for the representation of statistical structures and associated metadata to identify elements that support and can be used by modelling processes.
3. Make proposals about the generic structures and functionality needed for the representation of models and processes, so as to support both the implementation of modelling systems and the use of the results of these systems.

1.4 Results presented in Deliverable D3.1

The following results are presented:

1. Background information about the approaches to statistical metadata and statistical methodology that motivate the approach of the report.
2. An analysis of the nature of statistical modelling and the associated components of model specifications.
3. Metadata structures (shown as UML diagrams) that might be suitable for the management of models and model fitting processes.

1.5 Relation with the OPUS Life-cycle

The objective of the opus project is to develop a methodology for the integration of multiple datasets in complex situations. Work package three supports this, through the development of models for the data structures and processes that are needed to support the implementation of the generic methodology of the project. This work package is dependent on the project is a whole, and particularly on work package two, for specifying the types of methodology and statistical method that need to be covered as meta data. The development of methodology is not of itself dependent on work package three, though it is expected that consideration of data structures at an abstract level will contribute to the thinking about the generic methodology. However, the impact of this work package is expected to be felt most in the implementation efforts that will be part of the trials that follow development of the generic methodology.

1.6 Structure of the Deliverable

The following chapter contains a discussion of the general nature of statistical modelling as approached by the Opus project, and finishes with some basic ideas about statistical metadata. In this the distinction between different forms of model (for example statistical models and system models) is brought out. Note is also taken of the different levels of abstraction that need to be addressed at different times.

The following chapter is background information about the development of important ideas about statistical metadata over the past decade, with a bit of earlier history.

This is followed by an extended discussion of the components of a statistical model and the modelling process, culminating in a proposal for a conceptual structure for the representation of statistical models (in the form of a UML model).

The next chapter extends the discussion and structure to cover the model fitting (calibration) process, and touches on the representation of the links between results generated from a model and the model that is their source.

Final chapters summarise previous sections.

It should be noted that this report concerns itself with a very abstract view of statistical modelling. Practical, implementation and operationalisation issues are very important, but are not (in general) addressed here. These matters will be addressed in future documents from the project, and some may feed back to leaven a later version of this report.

2. THE OPUS APPROACH TO MODELLING

2.1 Introduction

2.1.1 Types of Model

Models are abstractions from real-world situations, designed to support some particular context. In this report we are concerned with two different but related versions of this concept.

1. Statistical Modelling. This is the process of determining and calibrating a suitable representation (model) for the underlying system for which statistical data has been collected. Different systems and data require different models, and models should be updated when new data arises or new understanding is recognised. The Opus methodology focuses on the construction and calibration of such models.
2. Metadata Modelling. Statistical Metadata is used to describe, document and control statistical systems. A model for statistical metadata is a definition of the structures and functionality that are needed to describe statistical systems, including statistical models. This type of modelling is close to the computer science ideas of system and data modelling, and draws on those ideas. This report is an exploration of the (single) metadata model that is needed to be able to describe and document statistical models (in general).

2.1.2 The role of Modelling

Models are designed to meet a particular need in a particular context. Thus the form and roles of models can be very different. Some examples may help to show some of the range.

Conceptual Models are an attempt to form a frame of reference for some domain or collection of constructs or concepts. Where concerned with terminology or names (and so sometimes called *Ontological Models*) they are often similar to classification structures, and such structures (for example the International Classification of Diseases – ICD – or NACE, the General Industrial Classification of Economic Activities within the European Communities) can be seen as conceptual or ontological models. Other conceptual models may be concerned with suitable structures for organising ways of thinking about a domain, and the models presented in this report are generally of that nature.

The *Relational Database Model* is a formal specification of the structures and behaviour for databases formed from sets of rectangular tables. This provides a conceptual framework for thinking about databases (one that is widely used) but is also sufficiently detailed and precise to be the basis for the implementation of many database software systems.

The *Object Oriented* approach[1] is an alternative (more general) way of thinking about databases and program structures (an alternative paradigm), built using a different set of primitive constructs, assumptions and conventions.

The statistical *Generalised Linear Model* is a mathematical specification of the way in which a set of predictor variables influence a dependent variable, together with the form of the variability about that relationship. This model is very flexible and is widely used for estimating statistical relationships (using suitable software to calibrate the model to a particular data set), and for discussing the potential form of such relationships. Of course, there are many situations where the GML is not an appropriate form of model.

Structural Models concentrate on the objects and attributes that are used to represent information structures. This is necessary for the exchange of information between systems, but needs to be accompanied by clear specifications of the intended purpose and use of the various elements. Inconsistent interpretation by independent users or implementers working with such a structure is a continuing concern, unless some enforcement mechanism can be specified and implemented. Structural models can be conceptual, in that they provide a way of thinking about the appropriate structures for some context, or they can be physical, and so present the actual structures needed for some particular system.

With statistical metadata we are looking for models that allow us to interchange information between processes and systems and that provide a stable conceptual framework for users to work with complex information structures across processes and systems. We want to support users of statistical systems, support the automation of statistical processes, and exchange information between systems and processes.

We can have more than one model, focussing on different parts of the statistical process, but they should dovetail together when a wider picture is needed. And we should aim to get suitable models accepted as standards, agreed and used across the target domain.

2.1.3 Elements of Modelling

To be functional and useful for people and software, our models must provide formal specifications of components and relationships, in a way that avoids misinterpretation. They must address:

- Structure: how are the elements organised, how are elements grouped and related, what attributes are needed for each type of element.
- Semantics: what do the elements represent, what rules and constraints apply to their attribute values, to their states and to the way in which they are used.
- Methods: specifications of algorithms and processes that apply to the elements and the data they refer to.

[1] A brief description of the essential elements of the Object Oriented Paradigm is reproduced in an appendix.

- Concepts: complete and detailed definitions of the terms and concepts that are the subjects and objects covered by the model and of the relationships between them. In some situations this may correspond to the idea of a thesaurus.

To construct models quickly and accurately we also need a modelling framework or workbench, which provides generic building blocks for model components and tools to support the design process.

For this project we have chosen to use the Universal Modelling Language (UML) to construct and express the structural models that we develop. The UML diagrams presented here were developed using the UML component of the Microsoft Visio 2003 drawing package.

2.1.4 Levels of Model

Models exist at various levels of abstraction, and confusion can arise from not recognising the level to which a particular construct contributes, or at which a discussion about the model is taking place.

For example, the metadata about a particular survey forms an instance of the more general model that can be used to describe other surveys (of the same general type). This in turn will draw on both a conceptual model of the application domains for which the model is appropriate, and on a more abstract model of statistical processes and surveys in general. These abstract models of statistics are sometimes called meta-models, and are themselves constructed as instances of an even more abstract model for the process of defining models.

Within Opus we are concerned with different levels of abstraction for the statistical models we will use. These extend from the very general description of an application domain (the GAPM, or Generalised *a Priori* Model) that sets out the underlying relationships and system knowledge, through to the much more specific and specialised models that are to be calibrated against available data to investigate a particular characteristic of the system.

Once grasped, the fact that there are different levels of model does not need to cause confusion, but the failure to recognise the levels can be very confusing.

2.2 Modelling for Metadata

2.2.1 Statistical Metadata

With Statistical Metadata we are mostly concerned with software to support the processing and analysis of statistical information. Models provide the opportunity to specify how information can be shared between stages of processes (so that later stages can make use of information entered in earlier ones) and how information and specifications can be moved between independent applications. Because we are supporting the development and use of software, our models need to be detailed and precise in their specification of the structures and semantics of the information. However, the model also determines a conceptual framework for process designers and software users, so they must be able to view elements of or generalisations from a

model, with less detail than is needed by software developers. Furthermore, when developing a model we need to work with domain and subject specialists to discover their needs and to help them to agree on model components and structures. These people will probably need assistance to express this knowledge in ways and with sufficient precision for use in the model, and will need help in understanding the model representation of their knowledge, so that they can confirm that the model represents this knowledge correctly.

We use the following definition of statistical metadata.

> *Statistical Metadata is any information that is needed by people or systems to make proper and correct **use** of the real statistical data, in terms of capturing, reading, processing, interpreting, analysing and presenting the information (or any other use). In other words, statistical metadata is anything that might influence or control the way in which the core information is used by people or software.*

It extends from very specific technical information, used, for example, to ensure that a data file is read correctly, or that a category in a summary table has the right label, or that the design of a sample is correctly taken into account when computing a statistical summary, right through to descriptive (intentional) information, for example about why a question was worded in a particular way or why a particular selection criterion was used for a sample.

Thus, metadata includes (but is not limited to) population definitions, sample designs, file descriptions and database schemas, codebooks and classification structures, processing details, checks, transformation, weighting, fieldwork reports and notes, conceptual motivations, table designs and layouts.

The implications of this definition are explored in the next chapter.

2.2.2 Acknowledgements

Many of the metadata ideas presented in this chapter have been expounded and discussed during the MetaNet project (see [MetaNet]), particularly in work groups 1 and 2 and at the final conference. Of particular importance is work with Chris Nelson, of Dimension EDI, but the current author takes full responsibility for all the ideas and opinions expressed here.

2.3 A generic approach to Statistical Modelling

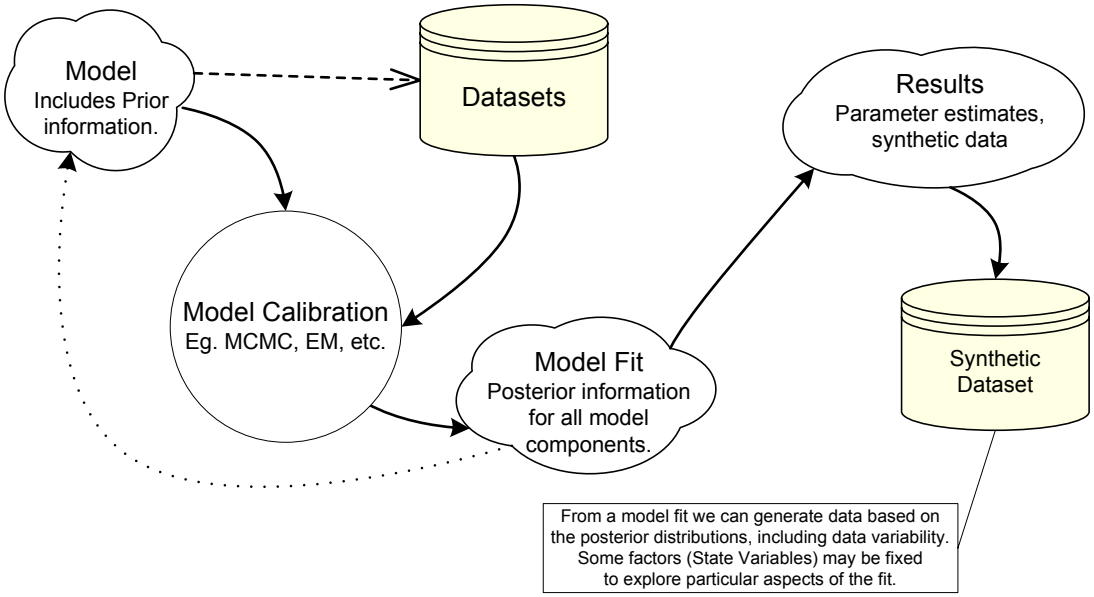
This section sets out my personal view about how a statistical modelling framework might look – at a very generic level. It omits all details (and these are very important in practice, particularly for selling the approach to domain specialists), and some other generic approach might be equally appropriate.

2.3.1 Statistical frame of reference

The theoretical approach of OPUS is Bayesian in nature, implying:

- An a priori starting point (model) is constructed, including implicit representations of confidence in data sources (through prior distributions) and modelling assumptions;

- Additional information is supplied and used to update the model;
- The updated model can be used to provide coherent estimators (with estimates of reliability) for any area that it covers, including combinations of factors for which no data were actually observed. For example it could provide estimates for passengers leaving a particular railway station in a period when no survey information was collected, but overall passenger loading is known;
- As well as parameter estimates, it is possible to use to model to synthesize simulated data sets that demonstrate behaviour of the system, including its variability.



There is scope within the project for the reliance on Bayesian methods to be supplemented with other techniques without altering the general vision. For the present, it is assumed that OPUS will implement its approach using MCMC (Markov Chain Monte Carlo) simulation techniques already widely used in statistical studies, but this is subject to the theoretical phase of work that starts the project.

2.3.2 General Approach

The general algorithmic structure of the approach is an application of the EM (Expectation – Maximisation) algorithm. This can be used in complex models where the model is too complicated to fit all at once, but it can be partitioned into components, each of which can be optimised. The problem is that all the components depend on all the other ones. The solution is to optimise over one component at a time, while holding all the others fixed at (hopefully good) guesses. Then, using the optimised values from the current component, we move on to optimising the next one. This continues round all the components, and then iterates until stability is reached. This can be computationally intensive, but is within the capabilities of modern processors.

As to the model itself, we should investigate whether a very generic approach is possible. I acknowledge that there may be problems in presenting a more generic form, but at least we could think that way. For example, the Normal and Poisson distributions are often used, but these are just special cases of much more general classes of distributions, so perhaps they could be given as examples, rather than being built in as assumptions.

2.4 A Model for Statistical Modelling

2.4.1 Introduction

In transport (for example) we have specific needs to integrate multiple, partial data-sets, but the motivation is to approach the whole area in a much more general way, since similar problems arise in many domains. Applications always require domain knowledge, so we treat these as case or feasibility studies in which we explore the problems that arise when the methodology is applied.

2.4.2 Model Structure

The heart of the model is a specification in mathematical terms (i.e. largely algebra) of the factors that influence traffic flows (or some other system being studied) and the way in which they interact in their influence. Of course, the particular factors and form of relationships are specific to the problem we are addressing.

All the factors will have distributions associated with them (i.e. they are not necessarily assumed to be fixed), and all the distributions and relationships will have parameters.

All the parameters have prior distributions (representing prior knowledge or uncertainty), which will be more or less informative depending on what experience we can bring to the context and the understanding of the model.

In addition the model can have constraints, which in general will be distributional (giving the likelihood of a particular arrangement), though they can be explicit (only particular arrangements are valid).

2.4.3 Bayesian Approach

In simple statistical analysis we represent the uncertainty associated with an estimate of a parameter by calculating a confidence interval. For different levels of confidence we obtain different intervals (or limits) and we can represent the set all limits as a distribution over the possible parameter values. In many cases this will take the shape of a Normal distribution, because the Normal distribution is assumed for the data.

Although we can represent our uncertainty about a parameter as a distribution, this does not mean that the parameter is a random variable. Rather, it is a fixed property of the reality about which we have collected data, and it is our uncertainty that is represented by the distribution.

We can take the idea further, and represent any uncertainty with a distribution. Thus we do not require that the distribution is derived from data, we can simply invent it. Of course, it is not sensible to do this without some prior knowledge, or justification, to support the particular choices that we make. Where we do have knowledge about the parameter we tend to talk about knowledge rather them uncertainty distributions.

With uncertainty represented in the form of distributions, we can draw on what is known as Bayesian Methodology for working with our models.

Bayes' theorem is a simple statement about conditional probability. It comes from the recognition that a joint probability can be written as the product of conditional and marginal probabilities.

$$P(A \wedge B) = P(A|B) \times P(B)$$

Bayes' original use of this was to show how to calculate conditional probabilities, as:

$$P(A|B) = \frac{P(A \wedge B)}{P(B)}$$

In contrast, Bayesian Methodology uses the first formula twice to show how to reverse the ordering of the conditioning.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

If we now substitute a parameter θ for A and consider B to be our data X, we have:

$$P(\theta|X) = \frac{P(X|\theta) \times P(\theta)}{P(X)}$$

This formula says that where we have prior knowledge about the parameter θ, we can update this knowledge if we also know the data distribution depends on the parameter, obtaining the posterior knowledge distribution.

Usually the mathematical form of the distributions is too complex to allow explicit determination of the posterior distribution, so we resort to simulation methods such as MCMC, which provide an empirical estimate of the distribution.

2.4.4 Model Uncertainty

As well as uncertainty about the values of parameters in the model, we may be uncertain about the appropriate form for the model. We can cope with this by introducing additional parameters to control the functional form of the mood, in addition to those that relate directly to the system of interest.

We thus represent uncertainty in the model form or structure by choosing generalised forms of distributions and relationships, and associating prior distributions with those parameters that determine the specific forms of the distributions and relationships. Thus (for example) instead of specifying the Normal distribution for a particular component of the model we might specify the Exponential Class of distributions, and define a prior distribution (perhaps favouring the normal form) over the parameter(s) that determine the particular form of the distribution that is appropriate.

2.4.5 Model Fitting

We then calibrate the model using whatever data is available to us.

For any set of observed data the model will imply a distribution of such observations (based on the parameters and their prior distributions). The calibration process consists of updating the distributions associated with the parameters (obtaining posterior distributions) so as to optimise the fit between the predicted distribution of observa-

tions and the actual one (this is where the MCMC approach may be needed, for intractable model components).

This becomes one step in the EM algorithm, and we continue with other datasets, and iterate to stability.

Note that there is no problem about having more than one set of data about the same subject – they are treated equally, as independent steps in the algorithm. Also, there is no problem when new data arrives. You can do just one more EM step using the new data, which will update the current best estimates in the light of the new data. Better still, rerun the whole EM process, incorporating the new data – because we will start with the current best fit the iteration should be fast, unless the new data is significantly in conflict with previous information.

2.4.6 Model Evaluation

The posterior distribution associated with each parameter (after calibrating the model) encapsulates the information available about the parameter, so we can always make statements about the degree of confidence we have in any parameter value (though this is dependent to some extent on the initial (prior) assumptions). Where nothing in the data tells us anything about a particular parameter, that parameter will retain its prior distribution unaltered.

2.4.7 Weights

Each dataset will have weights, which internally tell us about the relative importance of each observation (with respect to the underlying population distribution) and externally (in their overall total) about the importance or confidence associated with the dataset. Note that adjustment of the external weights gives us a mechanism for 'ageing' the contribution of a dataset to the model – alternatively we can introduce explicit time-varying components in the model and estimate these.

2.4.8 Using the Fitted Model

Once a model has been fitted (calibrated) it needs to be used to address problems of practical importance. So far there has been no mention of synthesised or generated data. That is because we do not need any for the calibration process. We transform the model to the data for calibration, rather than the other way round. (It's probably not quite as simple as that sounds!) The model contains everything we need to know to represent the processes of interest.

However, the information in the model is not necessarily in the form we are interested in. So we will need to project information out of the model into the form of direct interest to users.

For example, a flow on a segment by a mode under some set of conditions (time, weather, etc) will be derivable as some function of relevant parameters. The model will yield a distribution for the information of interest, which encapsulates the variability inherent in the model plus the uncertainty (distributions) associated with the parameters. From this we can estimate particular values or relationships, accompanied by estimates of their variability, or we can simulate the behaviour of some sys-

tem (component of the model) under suitable assumptions about fixed or varying factors.

Note that the distributions of parameters are not (in general) independent and the functions can be complex, so the output distribution will probably have to be generated by simulation (and different items of output information will not be independent either).

2.4.9 Data and Models

Models are not dependent on the availability of related data. This may seem odd, and it is certainly true that a model is unlikely to tell us anything new unless there is some related data.

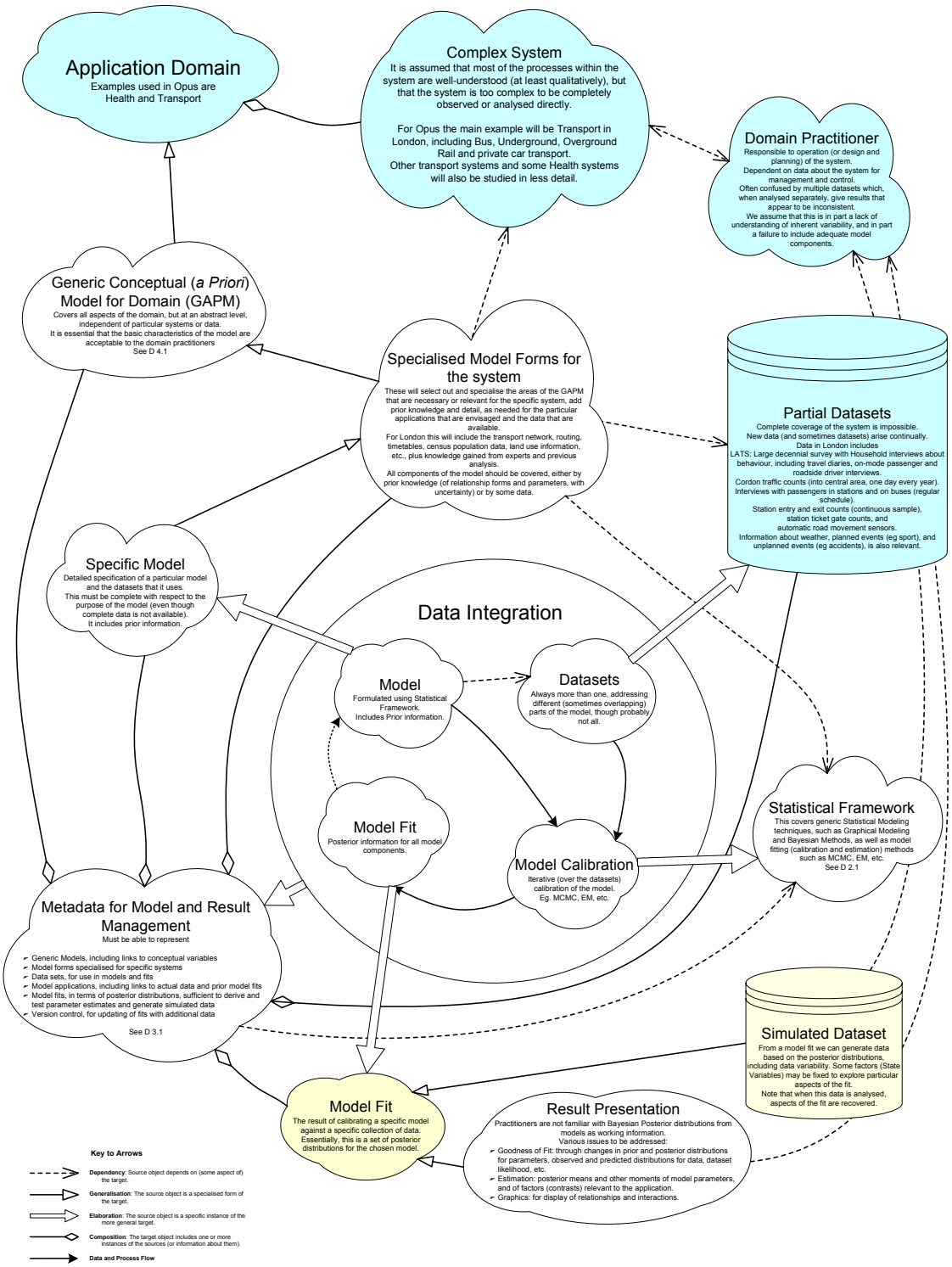
However, the importance of this idea is that the formulation of a model does not need to be constrained to the concepts or variables that are widely available in datasets. If we have no data about some component of a model, then we rely entirely on our prior knowledge about it. But we have not lost anything by including the component in the model.

In a related way, it does not matter that some datasets have less detail than others. We do not need to reduce all datasets to the lowest common denominator, throwing away detail that is not available in all datasets. Rather, each dataset contributes what it can to the model, and the model copes with the different detail available in different datasets.

2.5 The Opus Methodology

2.5.1 Components

The following diagram was originally prepared as a poster to present a summary of the Opus methodology. It draws on the general approach to modelling described above. However, while still very generalised, it is more specific than this general modelling approach, because it focuses on the situation of Data Integration, and the methodology needed to support and implement this.



2.5.2 Metadata for the Methodology

The current report is mainly concerned with finding a representation for the processes in the central circle (the Data Integration processes), and with the representation of the metadata that controls and records these processes.

3. REVIEW OF BACKGROUND AND CONCEPTS

3.1 What is Statistical Metadata?

3.1.1 Beginnings

Statistical Metadata started with the first general purpose statistical packages in the 1960's, software such as BMD, XTab and SPSS. These all needed information about 'Variables': which punch card columns (or paper tape fields) contained information about which statistical measurement, had the values been scaled, what range of values was allowed, had any special codes been used (for example to indicate missing data), what is a suitable label for the measurement, what were the meanings of the codes used for classification variables, and so on?

Quickly the idea of the Data Dictionary (or, sometimes, the Codebook) gained acceptance. This was intended to contain all the information about the data that was needed to perform statistical analysis, but which was not the actual data.

The term Metadata was coined in the early 1970's. The first use was probably in the 1973 PhD thesis of Bo Sundgren (subsequently with Statistics Sweden for many years). The construction draws on the Greek word *meta*, meaning 'beside' or 'with', so that 'metadata' is 'data beside data', or, more usually, 'data about data'. This is clearly linked to older constructions, such as 'metamorphic', 'metaphor' and 'metaphysics'. It has also been used for later constructions, such as Statistical Meta-analysis, 'the analysis of analyses'.

3.1.2 Definition

Statistical Metadata is concerned with support for the processing and analysis of statistical information.

We use the following definition.

> *Statistical Metadata is any information that is needed by people or systems to make proper and correct **use** of the real statistical data, in terms of capturing, reading, processing, interpreting, analysing and presenting the information (or any other use). In other words, statistical metadata is anything that might influence or control the way in which the core information is used by people or software.*

This definition extends from very specific technical information, used, for example, to ensure that a data file is read correctly, or that a category in a summary table has the right label, or that the design of a sample is correctly taken into account when computing a statistical summary, right through to descriptive (intentional) information, for example about why a question was worded in a particular way or why a particular selection criterion was used for a sample.

Thus, metadata includes (but is not limited to) population definitions, sample designs, file descriptions and database schemas, codebooks and classification structures, processing details, checks, transformation, weighting, fieldwork reports and notes,

conceptual motivations, table designs and layouts. In this particular report we are concerned with metadata for the specification, fitting and use of statistical modelling.

Metadata is never new information, but must always already exist in some form. Whenever a questionnaire is designed, the motivation behind questions needed to be thought through, the coding needs to be determined, and the sample design must be elaborated. What is new about the metadata idea is that the information must be formalised, organised, and made accessible. Metadata supports the task of making information available, so that it can be communicated from the place where it is first created to those places where it needs to be used.

An explicit concept of metadata can improve the quality of information, because the need for formality places an obligation on the creators of information to think clearly about purpose and content. The emphasis on use means that metadata systems need to be designed with usefulness and usability as important criteria. Because the potential uses of metadata are very wide, the need for usefulness should be seen as a motivation towards rich design, not as a constraint leading to a minimal, lowest common denominator approach for a single usage context.

Discussion of whether certain information is or is not metadata is generally pointless. I take the view that if information is potentially useful, and it is not data for analysis, then it is metadata. However, even this can be too restrictive, because information that is metadata in one context can be seen as statistical data in another. Because all metadata is data (and so is accessible for analysis if required), this presents no problems.

3.2 Some History of Statistical Metadata

3.2.1 Codebooks, Data Documentation and Relational Databases

Packages such as SPSS introduced commands for handling metadata (mainly the data layout and labelling commands), but treated these as part of the job command structure. The idea of the internal Save File included the metadata associated with the extracted data, but this was not accessible or reusable in other related contexts.

The idea of treating the metadata about a dataset as an independent block of information (a Codebook) arose in the 1970's, with the Osiris package from the ISR in Michigan being important in this development. These ideas were carried forward to the World Fertility Survey (WFS, from 1975 to 1984), where an independent Data Dictionary System was developed. This consisted of a separate file (of fixed format card images) for each data file, giving the layout and coding of all the variables, plus information about special (missing and not applicable) values and some background context.

An important aspect of the WFS system was that it was supported by functionality. A library of Fortran routines was developed, so that programs could be written that accessed the metadata directly. Equally important, an interface program for SPSS was developed. This allowed the user to select the dataset to be analysed and to select (by name) the variables to be analysed from that dataset. The program then generated the

commands needed by SPSS to access the data and label the variables, and integrated these commands with the analysis commands written by the user.

At around this time, the first commercial Relational Database Management Systems (RDBMS) were appearing, based on the model proposed a decade earlier by Ted Codd. This model provides an integrated way of dealing with related datasets, and this model is far richer than the single card-image file that is the basis of almost all statistical systems. Codd's original proposals explicitly address issues that we now see as metadata, by requiring that all specification information about a RDB should be accessible as data, and by including the concept of Domains, corresponding to code lists. Unfortunately, these aspects of the model have not been developed in most commercial systems (or, at least, not to the point that they support statistical metadata requirements well).

Relational database systems have reached an advanced stage of refinement, and are now an important component for any statistical data storage and processing system. This includes the storage of statistical metadata. The problem is that the special functionality needed for handling statistical metadata is not available as native functionality within any RDBMS, but still has to be programmed separately.

3.2.2 Other Metadata

Statistics does not have a monopoly over the metadata concept. An important other strand, relevant to statistical uses, comes from the Information Science domain. In the 1980's, the Standardised Generalised Markup Language (SGML) was defined, under initiatives from the Librarian community, for use in describing documents. This is using the idea of an extended, structured abstract from a document as a means of cataloguing and discovery, without reference to the detailed content of the document.

This idea has been developed and broadened, and led to the Dublin Core standard for document metadata. The UK government (along with many others) has elaborated the Dublin Core for its own e-GMS metadata standard for electronic documents – the identifying information at the beginning of this report is an example of metadata based on the e-GMS design. This standard is part of the UK e-government initiative, and use of e-GMS is now a requirement for all UK government web sites. The purpose of this is to support automated search and discovery processes which operate across multiple sites without needing to know the nature or structure of the sites or the information that they contain.

One side-effect of the e-government initiative is that it is no longer necessary to explain the term metadata every time it is used. However, we do still have to explain that statistical metadata has much wider uses than for document discovery.

3.2.3 Recent developments

A number of projects funded under the EU 4th and 5th Research and Development Framework programmes managed by Eurostat (DOSIS and EPROS) included the development of statistical processing software. Several of these developed proposals for models for statistical metadata.

A lack of coordination amongst the metadata aspects of these projects was a motivation behind the MetaNet project (a Network of Excellence for Statistical Metadata), also funded under EPROS. This brought together many people working on statistical metadata in Europe and more widely, with the objective of identifying the best current work and carrying it forward. A number of important conclusions came from this project. Some are discussed on the next section, and important reports can be found on the MetaNet web site.

At about the same time, an important initiative was developing in the USA. This came from the data librarians, who have links in both the statistical and the information science fields. This led to the Data Documentation Initiative (DDI), which has produced a significant and ongoing metadata model for describing statistical data resources, called the DDI Codebook. While this has a number of limitations in terms of generality, it has considerable depth and applicability, and has been adopted by a number of statistical processing systems. Importantly for Opus, it is the metadata standard behind the Nesstar package which is used by both Transport *for* London and ETH in Zurich.

A related development has been the definition of the eXtended Markup Language (XML), which is based on SGML. This is a standard for the construction of languages for the interchange of complex information structures. It is text-based, and so is easy to transfer over the Internet. It is the basis for a wide range of exchange standards for various processes, and metadata exchange is obviously covered by this. Some authors see XML as the solution to *all* problems associated with the design and use of statistical metadata systems. While XML is extremely important as a tool for exchange, this wider enthusiasm is misplaced. See [West03] for further discussion.

3.3 Important Concepts for Statistical Metadata

3.3.1 Multiple Facets of Metadata

Froeschel et al [FWdV03] propose a five-dimensional approach to the classification and description of metadata. These five dimensions (or facets) are

- a *structure* facet (the "entity" dimension: what things are);
- a *view* facet (the "role" dimension: the different ways things are considered);
- a *form* facet (the "material" dimension: how things are represented);
- a *stage* facet (the "process" dimension: how and where things are used), and
- a *function* facet (the "agent" dimension: the purpose things are used for).

The details of these five facets are elaborated in the paper, but they serve here to reinforce the fact that statistical metadata serves many purposes.

3.3.2 Levels of Abstraction

Ideas can often appear at more than one level of specificity. For example, the idea of a variable can be applied to the actual data values that appear in a particular dataset, or to a standard designed for coding such a variable across several datasets, or to a more abstract definition of the idea that is intended to be represented by some measure, which could have several different coding schemes. Often there are three or four eas-

ily distinguished levels of abstraction, from the very specific, through more general, to very abstract and generic.

Much confusion can arise in the discussion of metadata ideas from the failure to be explicit about the level of abstraction being discussed, particularly when different discussants are thinking at different levels.

For the generic discussion of statistical models at the GAPM stage we need to think about variables at a level that is explicit about their intention (what they represent). Within particular statistical models, where variables are used in algebraic relationships, we need to be specific about their form, in terms of measurement or coding. It is only when we consider the fitting of models to data that we need to consider actual variables represented in real data files. So in this example we have three different levels of abstraction for variables, all of which are needed but at different points in our discussions.

3.3.3 Levels of Application

Statistical metadata can be used to support many different tasks.

1. Recording background knowledge (descriptions, definitions, motivations, assumptions, etc.) about a process or system in a textual form, so that it can be read by others. This textual form also supports discovery by searching the text.
2. Recording technical specifications of structures (including data sets), processes or systems. Doing this in a sufficiently formal way allows the information to be read and used by software, as well as by people. And where software is used to support the specification process it can generate the metadata directly.
3. Recording the application of process specifications, including the initial conditions, specifications and data sources used, intermediate stages and final outcomes.

The same information can contribute to different uses, and the users can have very different requirements. People generally need information presented in a way that explains and guides them and supports discovery and the development of understanding. On the other hand, systems that use metadata require a very formal structure so that necessary information can be found where expected and used as expected.

It is important to recognise the wide range of potential applications for metadata structures. Solutions for special cases or with a very limited range of application (even when that is the main focus for a task) are likely to exclude other uses and limit the options for re-use in other contexts. Instead, it is better and safer (if potentially more expensive) to recognise and address the complexity problem. The simpler cases can by handled by specifying defaults, so that the solution allows complexity to be ignored when it is not needed.

3.3.4 Metadata and Statistical Objects

Metadata does not exist on its own, it represents objects in a larger space or ontology. The more complete and precise the metadata specification, the clearer the function

and attributes of these objects become. This interaction between descriptions of data and the underlying concepts used for modelling or analysis needs to be understood.

An important (relatively recent) recognition (see [FWdV03]) is that advanced statistical metadata is tightly linked with the objects that are the building blocks for statistical methodology. These include things like sample designs, datasets, analyses, models, etc. An understanding of the structure and functionality of these objects (as classes at an abstract or generic level) leads us to understanding of the metadata needed for specific contexts (instances of the generic classes). This top-down approach is the basis for the UMAS (Unified Metainformation Architecture in Statistics) methodology proposed in that report.

A related but bottom-up approach is the Reference Model methodology (see [Karg03]). This is particularly good at getting a group of specialists to identify and agree on their requirements from metadata structures.

The implication of this for Opus is that in order to construct a model for metadata about statistical modelling, we must first refine and formalise our understanding of the statistical modelling process and the objects on which it operates. Then the metadata model will enable us to record information about these objects and processes.

4. THE REPRESENTATION OF STATISTICAL MODELS

4.1 Model Components

4.1.1 Overview

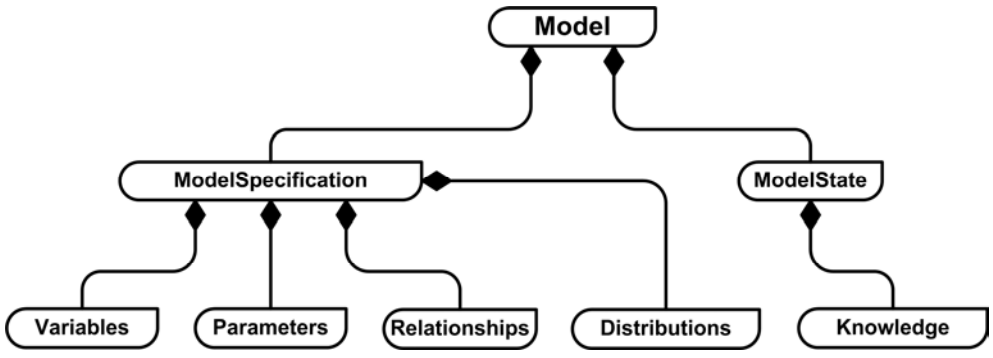
There are five essential elements for a statistical model. These are

1. Variables, which may or may not be observable, and for which we may or may not have any data.
2. Parameters (characteristics of the underlying system), chosen because we are interested in information about them, or because they are needed for our formulation of the system.
3. Mathematical relationships (of various forms) between the underlying constructs (variables and parameters) of the model, with parameters for the detailed specification of the relationships.
4. Probability distributions (from various families) for variables and parameters, with parameters and interdependencies.
5. Information (knowledge distributions) about the parameters in the model (for both the relationships and the distributions).

Note that the information (the fifth element) can be changed (updated) by calibration of the model against data – we can talk about a new or updated fit of the model to data.

The first four elements of the model form the model specification, and if we change any of these, we have a different model. Models may be closely related and inherit any or all of the elements, and it will be important to track such inheritance and change.

The fifth element represents the current state of the model, and the same model can have different states, for example, before and after calibrating the model with an additional dataset.



4.1.2 Model Specificity

We are concerned mostly in this report with the detailed models that are calibrated against actual data, for which all the elements are needed. However, the more abstract versions of models (up to the GAPMs proposed elsewhere) are also covered. These will generally have more detail about general structure (the first two elements) than will be applicable in a particular context, but less of the specific information represented by the last two elements. The links between models allow a specific model to be linked to the more abstract one on which it is based.

4.1.3 Data and Variables

Statistical models exist independently of any data. A model is unlikely to be useful unless at least some related datasets are available, but the conception of the model does not rely on the existence of specific datasets.

A model does, however, need variables. These need a firm conceptual basis (so that we can assign distributions and use them in relationships), and in many cases it will be necessary to be specific about the representation of the values of such variables, down to measurement methods or coding schemes. So we need a concept of variable that is at a level of abstraction (generality) above the specifics of fields in datasets, but that is more specific than the general description of motivations and objectives. We do not need to link to actual variables in data files: that is only needed when we use the model in a fitting process.

Notice that we will often need to use latent variables in models, that is, variables which are about members of a statistical population but for which no measurement process exists. We will also need variables which are characteristics (parameters) of the statistical populations (or the underlying system) rather than the observable units (members). In general, we can choose whatever formulation or characterisation of variables is most convenient for the specification of the model, because we can always transform this if needed for linking to data.

When we move on to trying to use real data to calibrate a model (by improving the knowledge associated with it) we will need to take account of mappings between the (conceptual) variables in the model and the actual variables measured in the data. This will cover differences in representation and coding, but will also need to take account of measurement processes, including sampling issues. In general these do not need to be included in the model, except where the model specifically includes aspects of the observation process.

An essential first step in constructing a model is to identify the (real and latent) variables that are of interest.

4.1.4 Parameters

These are characteristics of the underlying system. They can be thought of as variables that are about the underlying constructs in the system (populations, decision processes, the weather system) rather than about the observable data units that are the consequence of the system. Some of these parameters will relate directly to observable quantities in datasets. For example, a parameter which is the population

mean of an observable variable can be estimated by the mean value for the observations on that variable in a sample. Other parameters may be characteristics of distributions or relationships which only indirectly affect the observations in a sample (via their effect on other parameters or variables), and so can only be estimated indirectly.

Examples are the means (and other moments) of the distributions that we associate with variables, and slopes and intercepts in linear relationships between variables. Many relationships between variables will have parameters which represent aspects of the relationship about which we want information, or about which we are uncertain.

Similarly, we may be uncertain about the appropriate distributional form for a variable. We can represent this by introducing parameters that select a specific distributional form from within a wider class. Estimates for such parameters are obtained by finding the specific distributional form (from the family) that best corresponds to the data.

There can be relationships between parameters. For example, with two groups within a population we may choose to introduce a parameter which is the difference between the means of the two groups.

We often have some knowledge about the possible values of parameters, but there is always (more or less) uncertainty about them. We represent this uncertainty by associating distributions with parameters, just as we do with variables. These distributions are often referred to as 'priors', and when we have no prior information we use 'uninformative' or 'vague' priors.

Note that while the mathematics of the use of distributions is the same for parameters and variables, the interpretation is different, one being based on unknowable uncertainty, and the other on (potentially) observable variability. We can use this correspondence in the representation of the model, but we need to maintain the distinction between variables and parameters because they play different roles in the fitting (calibration) process.

4.1.5 Distributions

We assume that a set of useful mathematical forms for probability distributions is available as a primitive construct. This will include the Exponential Family[1], and any other continuous distributions that are needed. Similarly, a set of discrete distributions will also be available. All these will have parameters to make their form and moments specific when applied to particular variables. We will also need to be able to specify tables of multinomial probabilities for discrete classifications. Some form of parameterisation will be needed for these as well.

Joint distributions for related variables present more of a problem. The use of transformations and conditional independence means that in many cases, sequences of equivalent independent variables can be constructed. For example, a full rank set of

[1] The Exponential Family includes all the common continuous statistical distributions, such as Normal ($\mathcal{N}(\mu, \sigma^2)$), Chi-squared ($\chi^2(\upsilon)$), F, Cauchy, Exponential, Lognormal ($\mathcal{LN}(\mu, \sigma^2)$), Beta ($\mathcal{B}$), Gamma ($\Gamma$), etc. plus the derived discrete ones, such as Binomial ($\mathcal{B}$), Poisson ($\mathcal{P}$), Negative Binomial ($\mathcal{NB}$), Multinomial (or Dirichlet) ($\mathcal{M}$), … .

related normally-distributed variables can always be converted (through linear transformations) into an orthogonal (independent) set, from which the original variables can be reconstructed.

Similarly, mixture distributions (if needed) can be handled by a discrete selection stage followed by a simple distribution conditional on (chosen according to the outcome of) the first stage.

While many of the distributions that we use will have closed mathematical forms, sometimes we will only have a simulated estimate of the distribution. This happens frequently with the posterior distributions that result from Bayesian updating, particularly where the model structure is complex. Then we have to resort to MCMC methods to estimate the posterior distributions, and the result is a set of observations on the distribution. We can represent this estimate of the distribution either by retaining all the observations, or by using them to estimate the parameters of a suitable distribution chosen from our set of distributions, or by forming an empirical summary distribution, represented by a histogram or a smoother kernel estimate. Multivariate generalisation will probably be needed as well.

It will often be important to explicitly distinguish between different sources of variability when assigning distributions. For example, it may well be useful to take separate account of measurement variability (through faulty or concealed reporting by individuals, or counting error in equipment or by monitors), of response variability (where different members of a population behave differently, even after allowing for known predictive factors), and of underlying variability, representing unpredictability at the level of detail in the model (for example, the variability in the rate of flow of traffic along a segment for a given level of loading, or variability in the time taken for a given number of passengers to board a bus, or where the same person behaves differently on different occasions).

When dealing with a simple dataset these different sources of variability are generally rolled-up together. However, in a detailed model it will be important to be explicit about the point in the chain of variability at which relationships hold. For example, we will need to distinguish between the situation where a relationship affects the mean value for an underlying value, to which response variability is added, and that where the response is completely determined, but measurement variability obscures that value.

It may be useful to apply the idea of levels of abstraction to the use of distributions. At the concrete level we have actual observations about respondents. The first level of abstraction relates to the distributions used to summarise the variability in the observations. The next level relates to our knowledge, about the parameters of these data distributions and about the parameters of relationships between variables. A third level (not always needed), reflects uncertainty about the parameterisation, including uncertainty about precise distributional forms.

4.1.6 Relationships

The relationship component specifies how the variables in the model are related together. Every variable must appear in at least one relationship with other variables. Notice that there cannot be disjoint sets of variables in the model – if one set of vari-

ables has no relationship with some other set, then they are (by definition) parts of two disjoint models.

The only assumption we make about a relationship is that it can be expressed in a mathematical form. Many different types of relationship can be included.

1. Relationships between observations:
   $E\{V_i\} = \theta_1 + \theta_2 * v_j$, or, more generally
   $E\{\mathbf{V}_i\} = \mathbf{g}(\mathbf{v}_j, \boldsymbol{\theta})$,
   where capitals represent the abstract (random variable) form of a variable, lower case represents a realised observation and bold represents vectors. $\mathbf{g}$ is an arbitrary function.

2. Structural relationships between variables:
   $\mathbf{V}_i = \mathbf{g}(\mathbf{V}_j, \boldsymbol{\theta})$.
   Note that this is an exact relationship, so that all variability in the $\mathbf{V}_i$ is inherited from the $\mathbf{V}_j$, and the former are fixed (constant) conditional on the latter (though there may be observational error that introduces variability into the realised values).

3. Constraints on variables:
   $\mathbf{g}(\mathbf{V}_j, \boldsymbol{\theta}) = 0$.
   Note that this is actually a special case of the previous relationship.
   This form also includes inequalities, which can be thought of as logical expressions which must evaluate to the constant value 'true'.

4. Conditional independence:
   $\mathbf{V}_i \perp \mathbf{V}_j \mid \mathbf{V}_k$, or
   $f(\mathbf{V}_i, \mathbf{V}_j \mid \mathbf{V}_k) = f(\mathbf{V}_i \mid \mathbf{V}_k) * f(\mathbf{V}_j \mid \mathbf{V}_k)$.

Similar examples of relationship are allowed between parameters, though parameters cannot (logically) be defined in terms of variables.

We assume that the complete model can be specified by a set of relationship equations.

Specific statistical methodologies (such as Graphical Modelling or the Generalised Linear Model) fit into this framework, but may only allow a restricted set of relationship (and distributional) forms.

4.1.7 Knowledge

Our structural knowledge about the system being modelled is already embodied through the previous components, in the selection of variables, parameters, distributions and relationships. What remains to be expressed is our prior knowledge about the likely values of the parameters (the θs) used in these previous components.

We have already introduced distributions to express the uncertainty of our knowledge about the parameters. We now need to introduce whatever information we do have about the likely values. The mean of such a distribution will usually be our best estimate of the value of the parameter, and the variability (variance) will reflect our confidence (or uncertainty) in that mean value (so a distribution can be degenerate – constant, zero variability – in the unlikely case that we are certain about a value).

Usually we will use estimates from previous data or previous models, using the uncertainty associated with those estimates. We will often also have information about the shape of the distribution – is it limited to a finite range, always positive, symmetrical or skew? Where we have no information we will choose values that correspond to vague priors.

Note that parameters are almost always conceived as continuous – so we do not have to worry about discrete distributions here – and that a Normal distribution is generally a reasonable assumption for the uncertainty about a parameter that is an estimated mean.

We do not need to have any previous data about a parameter, we just need to be able to make a judgement and assign confidence to it through the form and spread of a distribution. If we are really not confident to do that, then we can introduce another level of parameterisation to express this uncertainty.

So, if we have a parameter (μ) that is a mean, and we have previous data about the mean, we may express this knowledge as $\mu \sim \mathcal{N}(m, s^2)$, where m and s are the mean and standard error previously estimated. This distributional statement represents our knowledge (or uncertainty) about the parameter μ, and the set of such statements constitutes out overall knowledge about the current state of the model.

In general, then, the last step in the model specification is the construction of a set of explicit distributional statements about parameters, where the distributions are specified through numeric values, not further parameters. We can characterise such parameters as 'terminal', in that out knowledge about them is not expressed in terms of other parameters.

Note that all other distributions in the model are defined through relationships or distributions that depend (directly or otherwise) on these terminal parameters.

4.1.8 Model Updating

As we have already said, when we update a model, the specification part of the model does not change, and all that can change is the knowledge part, represented by the distributions associated with the terminal parameters – we start with prior distributions and the updating process produces posterior distributions for these parameters.

The form of posterior distribution for the terminal parameters will be determined by the model and the data used for updating, and, in general, will not be the same as the prior distributions. If the posterior distribution for a parameter is the same as the prior, then we know that the data is not related to the parameter.

4.2 Examples of Simple Models

Here we present how a few simple models can be formulated using the components above.

4.2.1 Simple Regression

The standard formulation for this is

$Y \sim \mathcal{N}(\alpha + \beta^*x, \sigma^2)$,

which says that Y (the dependent variable) has a Normal distribution where the mean depends linearly on the (independent) variable x. Standard statistical methodology (the General Linear Model) provides us a way to obtain estimates of the three parameters in this model.

For the more general modelling formulation we need to make some small changes to this, and add in statements about prior knowledge.

Variables:

x, y, z

Distributions:

$Y \sim \mathcal{N}(z, \sigma^2)$

Relationships:

$z = \alpha + \beta^*x$

$\sigma^2 = 1 / \tau$

Parameters:

α — the intercept for the relationship
β — the slope of the relationship
σ — the variability of observations about the relationship
τ — the precision (inverse of variability) of the relationship

Knowledge:

$\alpha \sim \mathcal{N}(k_1, k_2)$
$\beta \sim \mathcal{N}(k_3, k_4)$
$\tau \sim \Gamma(k_5, k_6)$

We will have explicit values for k_1 to k_6, which are constants chosen on the basis of our knowledge (which might include prior data). Note that α, β and τ are terminal parameters, with explicit knowledge distributions, but that σ is not, since it is defined by its relationship with τ.

The formulation of the knowledge component is particularly appropriate for this situation of prior data analysed in the traditional way. However, we might have different knowledge. Suppose, for example, that we know (or believe) that the relationship between x and y is monotonically increasing, with slope near to 1 and passing near the origin, and that the average x value (and so also the average y) is around 1000. Then we might change the representation of this knowledge to:

Knowledge:

$\alpha \sim \mathcal{N}(0, 100)$
$\beta \sim \mathcal{Ln}(1, 0.5)$
$\tau \sim \Gamma(100, 0.1)$

Note the use of the lognormal distribution for β, which ensures positive values without imposing an upper limit.

There is usually a way of choosing the representation of knowledge that equates to no knowledge (vague priors), and so gives just about the same estimates from the model as the traditional method. However, we are more interested in the situation where we **do** have some prior knowledge, so this formulation is more appropriate.

4.2.2 Mode Choice

Consider the model for the choice between three modes of travel, say car, train and bus, by members of a population. The basic (and extremely simplistic) model for this choice might be:

Variables:

m – indicator for one of three choices

Distributions:

$M \sim \mathcal{M}(\pi_1, \pi_2, 1 - (\pi_1 + \pi_2))$ – multinomial distribution with 3 classes

Relationships:

$\pi_1 + \pi_2 < 1$

Parameters:

π_1 — probability of choosing car

π_2 — probability of choosing train

Knowledge:

$\pi_1 \sim \beta(2, 8)$

$\pi_2 \sim \beta(4, 6)$

The Beta distribution[1] is often used for priors of parameters that are probabilities, and these choices have means (initial best estimates) of 0.2 and 0.4 respectively, implying 0.4 as the mean for the third class.

Of course, the choice of a car is only possible if a car is available to the traveller. We can introduce that into the model as follows.

Variables:

m – indicator for one of three choices

c – indicator for the availability of a car, 1 if available, 0 if not.

Distributions:

$C \sim \mathcal{B}(\pi_c)$

$M \mid c \sim \mathcal{M}(\pi_1 * c, \pi_2 * (1 - \pi_1 * c), (1 - \pi_2) * (1 - \pi_1 * c))$

Relationships:

Parameters:

π_c — probability that a car is available

π_1 — probability of choosing car if it is available

π_2 — probability of choosing train if car is not chosen

Knowledge:

$\pi_c \sim \beta(6, 4)$

$\pi_1 \sim \beta(2, 8)$

$\pi_2 \sim \beta(5, 5)$

Notice that this parameterisation is different, in that π_2 now represents the conditional probability of choosing train rather than bus, independently of whether or not

[1] The beta distribution $\beta(\theta_1, \theta_2)$ is a distribution for values between 0 and 1. It has mean value $\theta_1 / (\theta_1 + \theta_2)$. For $\theta_1 = \theta_2 = 1$ it is the uniform distribution, and it is uni-modal if $\theta_1 > 1$ and $\theta_2 > 1$.

car is an option. We assume that the preference between train and bus does not depend on whether a car is available – this may or may not be a reasonable assumption.

This model could be extended further, for example by replacing π_1 with a function of other aspects of the context, such as the household income or the weather on the day the choice is made.

4.2.3 Models in WinBugs

The following model is the 'Eyes' example taken from the WinBugs manual. This models a single set of data as a mixture of two Normal distributions with separate means but the same variance, with unknown proportions in each. The specification in WinBugs (and the description from the manual) is as follows.

"The analysis involves fitting a mixture of two normal distributions with common variance to this distribution, so that each observation y_i is assumed drawn from one of two groups. $T_i = 1$, 2 be the true group of the i^{th} observation, where group j has a normal distribution with mean λ_j and precision τ. We assume an unknown fraction P of observations are in group 2, 1 - P in group 1. The model is thus

$y_i \sim$ Normal(λ_{Ti}, τ)

$T_i \sim$ Categorical(P).

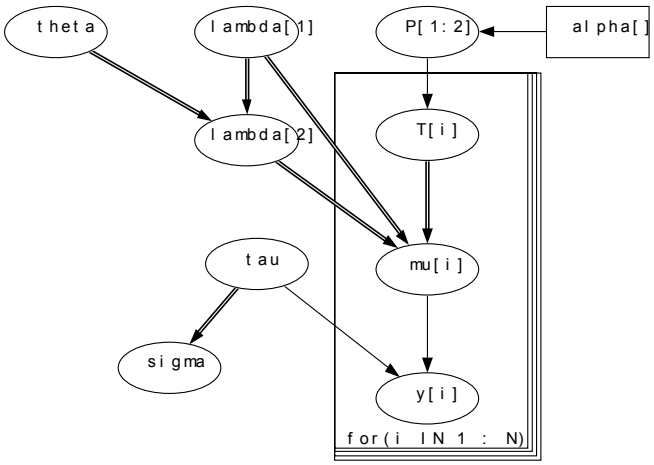
We note that this formulation easily generalises to additional components to the mixture, although for identifiability an order constraint must be put onto the group means.

Robert (1994) points out that when using this model, there is a danger that at some iteration, *all* the data will go into one component of the mixture, and this state will be difficult to escape from --- this matches our experience. Robert suggests a re-parameterisation, a simplified version of which is to assume

$\lambda_2 = \lambda_1 + \theta, \ \theta > 0.$

λ_1, θ, τ, P, are given independent "non-informative" priors, including a uniform prior for P on (0,1). The appropriate graph and the BUGS code are given below."

Graphical Specification	Textual Specification

	model
	{
	for(i in 1 : N) {
	y[i] ~ dnorm(mu[i], tau)
	mu[i] <- lambda[T[i]]
	T[i] ~ dcat(P[])
	}
	P[1:2] ~ ddirch(alpha[])
	theta ~ dnorm(0.0, 1.0E-6)I(0.0,
	)
	lambda[2] <- lambda[1] + theta
	lambda[1] ~ dnorm(0.0, 1.0E-6)
	tau ~ dgamma(0.001, 0.001)
	sigma <- 1 / sqrt(tau)
	}

This is in fact a simple system, with one observed variable, one unobserved one, no relationships (except to control the parameterisation), and no knowledge about the parameter values (non-informative priors). With our formulation, this model would be expressed as follows.

Variables:

y – observed measurement

t – population membership (1, 2)

Distributions:

$T \sim \mathcal{B}(\pi) + 1$

$Y \mid t=j \sim \mathcal{N}(\lambda_j, \sigma^2)$, $j = 1, 2$

Relationships:

$\lambda_2 = \lambda_1 + \theta$

$\theta > 0$

$\sigma^2 = 1 / \tau$

Parameters :

π — Probability of membership of group 2

θ — Difference in group means

λ_1 — Mean of group 1

λ_2 — Mean of group 2

σ — Variability of observations about the group mean

τ — Precision of observations about the group mean

Knowledge:

$\pi \sim \mathcal{U}(0, 1)$

$\theta \sim \mathcal{N}(0, 1000000)$

$\lambda_1 \sim \mathcal{N}(0, 1000000)$

$\tau \sim \Gamma(0.001, 0.001)$

4.3 Representation of Models

4.3.1 Variables

Existing proposals for the representation of variables can be used in this context. To be specific, we will need to list the variables assumed by the model, but all the details will exist in a separate component.

We will want to link to variables at a level that includes details of their representation (coding, etc). We do not need to link to actual variables in data files: that is only needed when we use the model in a fitting process. So we need a variable concept that is a level of abstraction (generality) above the specifics of fields in datasets, but that is more specific than the general description of motivations and objectives.

Because a model can include latent variables (which may not have been defined separately), it will be necessary to be able to add variable definitions to the variables component from within the model specification context.

4.3.2 Parameters

Parameters are internal to a model, but have many characteristics in common with variables[1]. It may thus be useful to use a similar mechanism to record definitions and motivations for parameters. We certainly need to distinguish between parameters and variables, because they have different roles in the model and the fitting process.

4.3.3 Relationships

We will need a representation of relationships that can both be displayed for users and interpreted by software for use in the calibration (fitting) process.

A number of systems exist for the construction (and execution) of mathematical expressions. Perhaps the best known is MathML[2], which is designed particularly for the rendering (display) of expressions in web pages. However, it has separate components for display (presentation) and semantics (content), with the latter being what is essential for our representation of models. A further system (OpenMath[3]) provides an extension of the MathML content system if that is not adequate.

Other systems, such as Mathematica[4], provide wider support for computer algebra and evaluation.

A few of the existing statistical data description systems make some attempt to include the definitions of variable derivation, and these may be of some interest.

A further area for investigation is the representation of functions and expressions in the standards related to Data Warehousing, such as XMLA[5] and CWM[6].

4.3.4 Distributions

Distributions are used for the variability of variables and for uncertainty (knowledge) about parameters.

A set of available distributions will be defined as primitives (so that the distributions can be referenced, not explicitly defined) in the model specification system. The definition of each distribution will specify a set of required parameters. A possible structure for this is shown in the following UML diagram.

---

[1] Note that, while variables and parameters may be associated, this linking is induced by the relationship and distribution parts of the model, and so should not be explicitly recorded with either the variables or the parameters. There may, however, be a more abstract level of metadata, relating to conceptual issues such as motivation and intention, to which both can be linked. This could then be used to discover that both a variable and a parameter were related to a concept of interest.

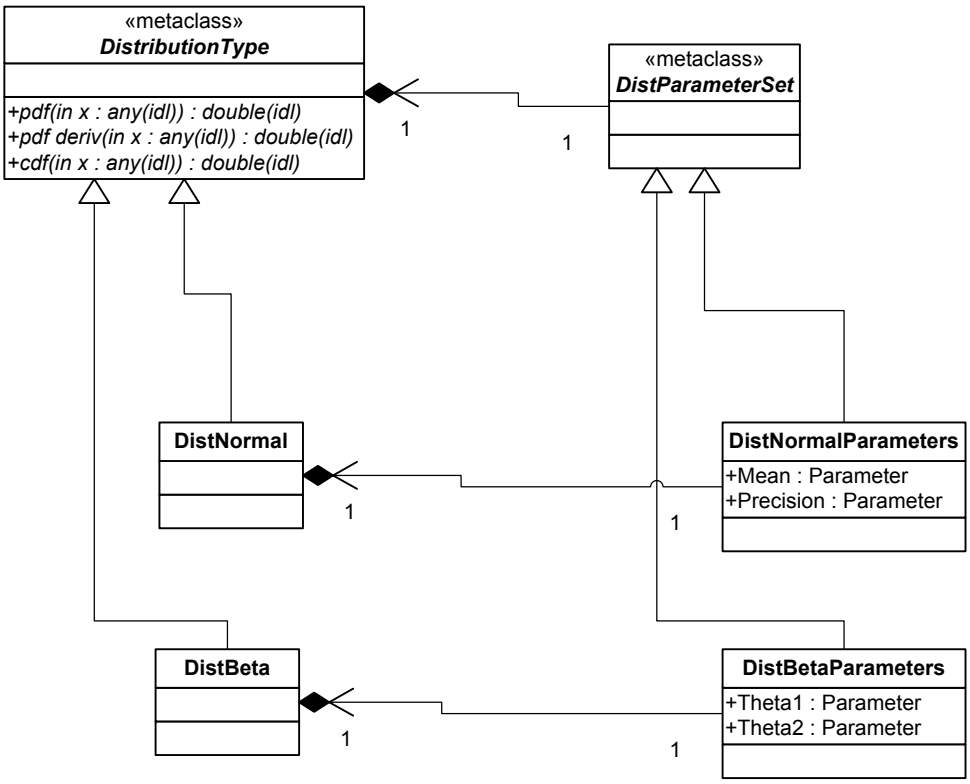
[2] See http://www.w3.org/Math/

[3] See http://www.openmath.org/cocoon/openmath//index.html

[4] See http://www.wolfram.com/products/mathematica/index.html

[5] XML for Analysis – see http://www.xmla.org

[6] Common Warehouse Metamodel – see http://www.omg.org/cwm



In this structure, DistributionType and DistParameterSet are general classes (abstract or metaclasses) which define general properties for all distributions. Two specialisations (for the Normal and Beta distributions) are shown as examples, and others will be needed.

Each distribution element of the model will consist of a reference to a distribution type, the variable(s) (or parameter(s)) being linked, and the parameters that make the distribution specific. Each parameter will be defined either through a relationship with other parameters, or through another distribution, or through a knowledge constant.

4.3.5 Knowledge

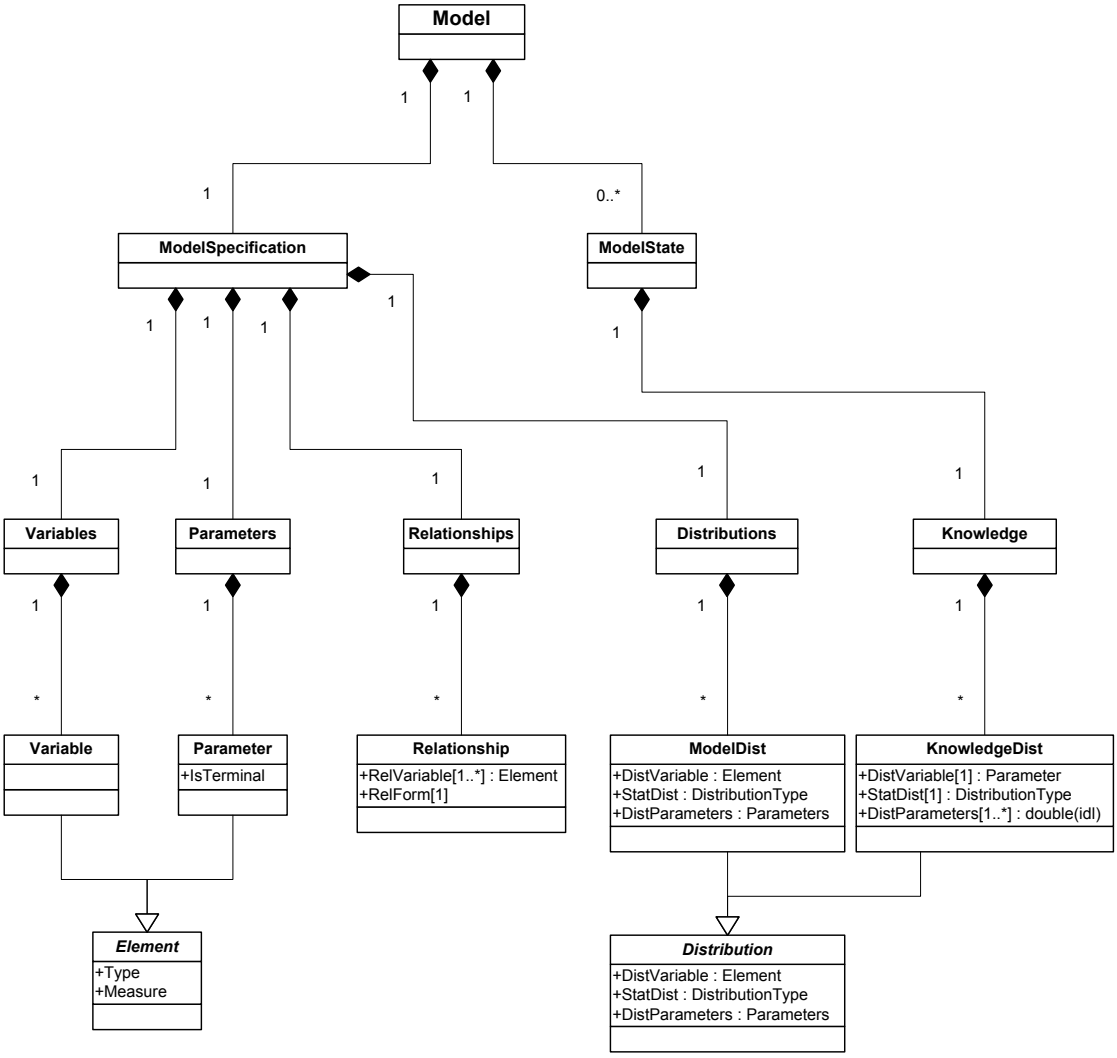
Knowledge (about terminal parameters) is here represented by fully specified distributions. These values represent our (prior) knowledge about the modelled system (before fitting to further data). After a fitting process we will have different (updated or posterior) distributions, which will feed into further fits. If the distribution associated with a parameter have not changed, then the data has told us nothing about this parameter (and the component of the system that it represents).

The knowledge in the system is thus represented by a set of specific distributions, each linked to a particular application point in the model.

4.4 A Metadata Structure for Models

4.4.1 Structure

The following UML diagram shows a possible structure for the metadata used to represent model specifications and states. This summarises the discussions in the previous sections.



4.4.2 Semantics

Amongst other things, this diagram shows that a Model consists of a ModelSpecification and a number of ModelStates. The former defines the structure of the model in terms of Variables, Parameters, Relationships and Distributions, while the latter represents the numerical Knowledge about the model. Distributions for model elements (ModelDist) and for terminal parameters (KnowledgeDist) are both special cases of Distributions, but the latter can only be about terminal parameters, and must be fully specified with numeric values for the distributional parameters. In contrast, the ModelDist can be about either parameters or variables, and the distribution must be specified through other parameters.

Initially a model will probably have no state information. This happens when we concentrate first on getting the model structure right, without trying to evaluate any knowledge. The first state for a model is likely to be based on guesswork, and includes the option for the uninformative state. Subsequent states can be derived by fitting the model to data – that process is discussed in the next chapter. Further states can also be introduced manually, perhaps to explore the effect of different initial values, or to represent different assumptions.

Variables and Parameters have many facets in common. This is represented in the (abstract) generalisation Element, which can be either, and which shows that both can have a Type and Measure information.

A Relationship is an algebraic expression involving one or more elements, with the precise form of the relationship probably being expressed in MathML. A Distribution applies to an element, and specifies the type of distribution to be used, together with the model parameters that specify the details of that distribution type. These parameters may be knowledge items or set by other relationships or distributions.

4.4.3 Additions

This diagram does not show the full specification of the representation of a model, and, in particular, additional information is needed in the form of constraints. These can be expressed in UML, but are not shown in the diagram. Examples of constraints are:

1. every variable must be referenced in at least one relationship;
2. every variable must either be set by a relationship or specified through a distribution;
3. every model parameter that is not flagged as a terminal parameter must either be set by a relationship or specified through a distribution.

5. MODEL FITTING AND MODEL RESULTS

5.1 Introduction

The fitting (or calibration) process involves the comparison of a model with one or more datasets, resulting in updated knowledge. This latter is represented by changed values for the constants in the knowledge component of the model, that is, by the transition from one state of the model to another.

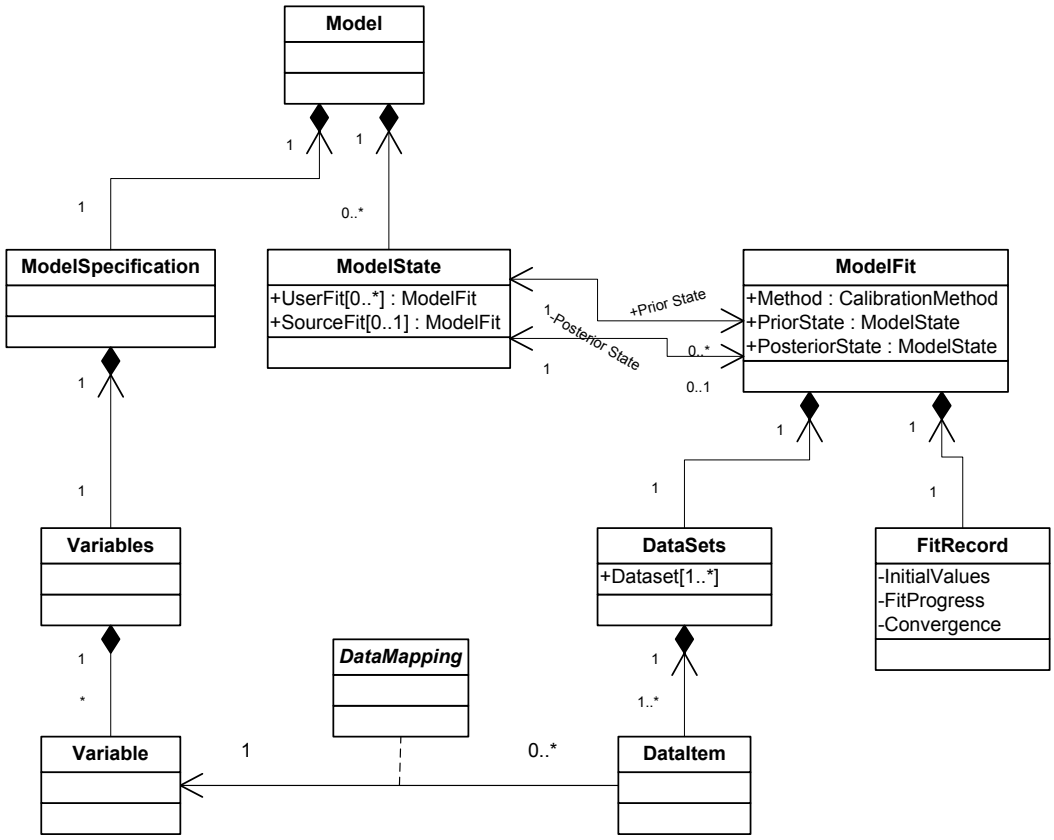
To record the fitting process we will need to know the model used and the datasets referenced. We will need to record both the prior and the posterior knowledge (parameter values). Because different fitting algorithms are available, all of which are (in general) approximate and involve randomisation, we will need to record technical details of the procedures used and the progress to the results (such as convergence measurements). This should allow us to make comparisons of the process if the same model (with the same prior knowledge) is calibrated again against the same data.

Results are a representation of a particular state of the model, and many results can be obtained from a single state. By using the linking from the state to the model specification and the fitting process, we can present aspects of the model with the results, and compare results based on different states of the same model.

5.2 Model Fitting

5.2.1 Structure

The following UML diagram shows how the model fitting process can be represented as additional structure associated with a Model. Only those parts of the model representation (as shown in the previous chapter) that are needed for this discussion are shown here.



5.2.2 Semantics

Each ModelFit is linked to a Model through its prior ModelState, and so has full access to all the elements of the ModelSpecification. Generally, there will be multiple fits based on the same model, and there can be different fits that use the same prior state. The result of the fit will be a new state (the posterior state) for the model. The fits thus act as links in chains of model states.

A fit will be based on data from a number of Datasets (often only one), from which various DataItems will be relevant. Each such item must have a mapping to a variable in the model. The mappings will allow for changes in the representation of information between the concrete versions in datasets and the slightly more abstract versions in the model specification. For example, Age and Income might be represented as continuous measures in a model, but only collected as grouped values in a dataset. A different example is where the model operates at a detailed level, but the data is aggregated.

The mapping specification must be such that it allows the likelihood of the data to be calculated from the distributional information associated with the variable in the model. Note that we do not require that every variable in the model has a mapping to data – unmapped variables will just not contribute (directly) to the likelihood calculations.

The FitRecord will record aspects of the fitting process that may be useful for later analysis of the process.

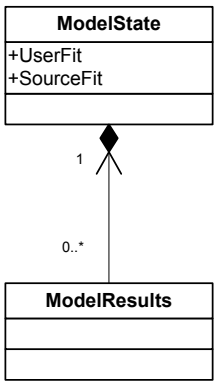
5.2.3 Link to Calibration Software

The CalibrationMethod attribute of the model fit is a surrogate for a link through to the software that actually performs the calibration process. WinBugs is an example of a package that might be used for this process.

Conceptually, the software can access the (instances of) the structure shown here and so extract all the information that it needs to perform the calibration process, and then it can add back the new information about the posterior state and the actual process. In practice, we will need to construct an interface to each such package. This will extract the information needed and convert it into the form that the package expects. It will also capture the results of the process and feed these back.

5.3 Model Results

Any results that we derive from a model are linked to a specific state, because we need the numeric values that complete the specification of the model state. We can then derive summary measures or generate synthetic data, using the parameters and the distributions from the model specification. This relationship is shown in the following (simple) UML diagram.

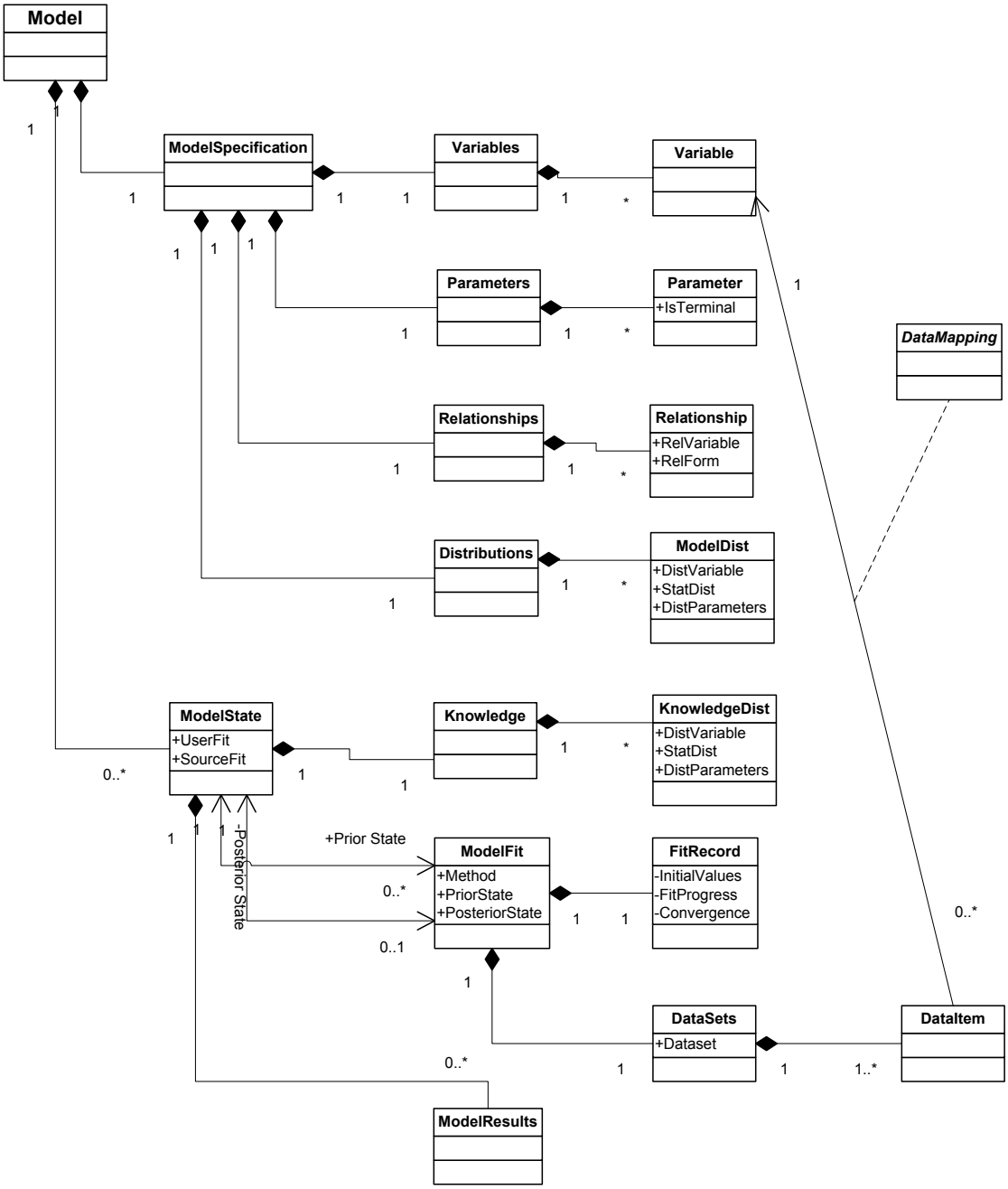


We can link results to the data used for the fit that created the state, and because the state is linked to a single model we can easily compare results created from different states of the same model.

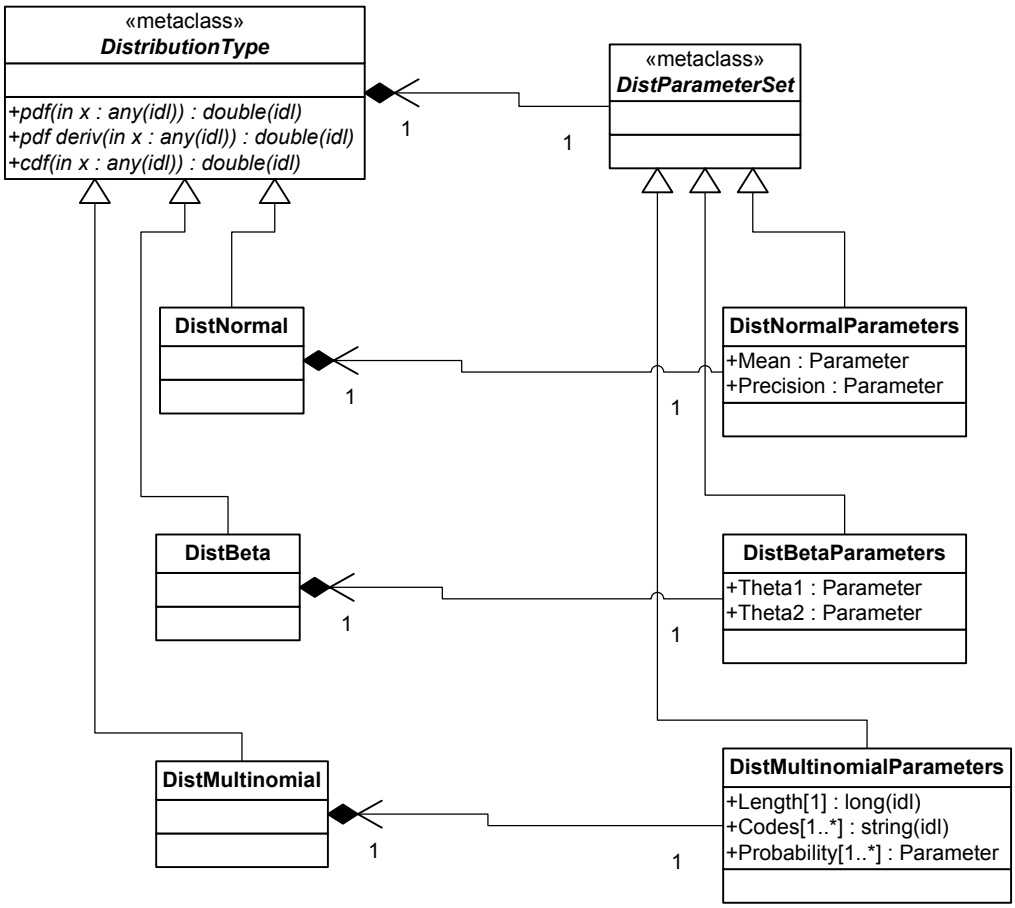
Discussion of the appropriate form for results from these modelling processes is beyond the scope of this report.

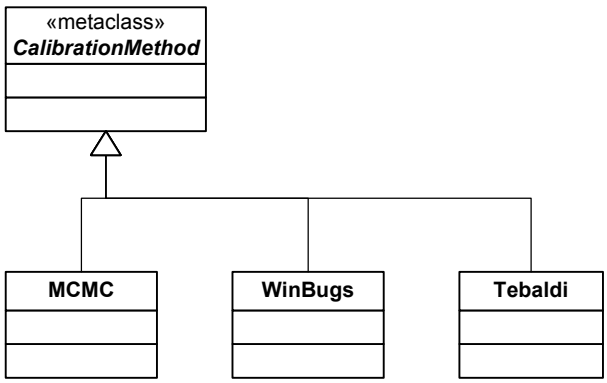
6. SUMMARY OF STRUCTURE

The following UML diagram shows all the elements of the model structure together. Some details that are included in the previous diagrams are omitted here, for clarity.



In addition, the metaclasses shown in the following diagram have been defined as generalisations that can be elaborated for specific distributions and calibration methods. A few such elaborations are shown, but more will be needed.


«metaclass»
DistributionType
+pdf(in x : any(idl)) : double(idl)
+pdf deriv(in x : any(idl)) : double(idl)
+cdf(in x : any(idl)) : double(idl)
«metaclass»
DistParameterSet
1
1
DistNormal
DistNormalParameters
+Mean : Parameter
+Precision : Parameter
1
1
DistBeta
DistBetaParameters
+Theta1 : Parameter
+Theta2 : Parameter
1
1
DistMultinomial
DistMultinomialParameters
+Length[1] : long(idl)
+Codes[1..*] : string(idl)
+Probability[1..*] : Parameter
1
1


«metaclass»
CalibrationMethod
MCMC
WinBugs
Tebaldi

7. CONCLUSIONS

7.1 Summary

We have analysed the conceptual structures needed for statistical modelling and proposed a basic conceptual structure for representing the metadata that is needed to work with such models and modelling processes. This structure is presented as a conceptual model in UML.

7.2 Further work

The next step is to bring the general concepts developed here to the real problems being addressed within the Opus project. This will lead to extensions to cover practical problems, refinements resulting from deeper understanding, and revisions where current structures prove impractical. This will all lead to the production of a revised version of this document towards the end of the project.

REFERENCES

[Fowl04] UML Distilled, 3rd Edition. ISBN: 0 321 19368 7. This is an excellent though terse guide to the content and use of UML, aimed at readers with some programming background.

[FWdV03] The Concept of Statistical Metadata (2003) by Froeschl, Grossmann, Del Vecchio, a deliverable from the MetaNet project, at www.epros.ed.ac.uk/metanet/deliverables/deliverables.html

[Karg03] Reference Models, developed by Reinhard Karge of Run Software under the MetaNet project, and maintained at www.run-software.com/downloads/documentation/ReferenceModel.doc

[MetaNet] See www.epros.ed.ac.uk/metanet.

[UML] See www.uml.org for information about UML 2.0. This is a standard developed under the auspices of the Object Management Group (www.omg.org).

[West02] XML and Standards, by Andrew Westlake. Available at www.sasc.co.uk/Guides/XML%20and%20Standards.zip.

[West03] Models and Metadata, by Andrew Westlake. See www.sasc.co.uk/Guides/models%20and%20metadata.htm.

APPENDIX 1: THE OBJECT PARADIGM

Much current (and recent) work on modelling is based on the Object Oriented paradigm, and this is the approach assumed for UML.

An *object* is a structured collection of information, an *instance* of a particular component (such as a classification). An object must conform to its definition, and the general definition of a particular type of object is called a *class* (not a particularly good choice of name). The specification of a class determines the structure and semantics of the objects that are instances of that class – the objects can contain different information, since they describe different instances, but their structure and behaviour is the same[1].

The specification of a class includes the *attributes* which form its structure – these may be simple (such as numbers or strings) or complex (effectively links to and collections of other objects).

Every object (instance) has a unique *identity*, and this can be referenced by other objects. Object identities are global, so object references do not need different forms for different types of object.

Classes support the idea of *inheritance*, *specialisation* and *generalisation*. One class can be defined as based on another, so that it inherits all the properties (structure and semantics) of its parent class. New structure and semantics can be defined for the child, but only those things that are different have to be specified. The child class is a specialisation of the parent, which in turn is a generalisation of the child. In particular, this means that a child class is also valid anywhere that the parent class can appear in a structure or an operation (because it inherits all its' parent's structure and behaviour). A child class can substitute its own behaviour for that of its parent if appropriate – this is called *polymorphism*. For example, a child could respond to a 'print' command differently from its parent, because it has extended content and/or more specialised understanding of how this should be presented.

A class can be *dependent* on another, in that it needs to know about the structure and semantics of the depended class, so that it can make use of it. This is a one-way relationship. Where one object makes reference to another (of the same or a different class) this is called an *association*, and this is usually bi-directional. For example, a data cube may be constructed with reference to a particular classification for one of its dimensions. In the implementation of the model the dimension could contain a reference to the classification, and the classification may maintain a list of all the dimensions that reference it.

References can be traversed without knowing what type of object is at the other end. Generally it is useful to design an object structure (class) so that references are organised according to their type, but it is always possible to follow a reference first, and then find out what type of object has been reached afterwards.

---

[1] The behaviour of an object may depend on the values it contains, but only in a way defined for the class as a whole.

INDEX