Integrating Information about Complex Systems:
The Role of Metadata in the Acceptability of Results from Models

Andrew Westlake and Marcus Wigan
A.Westlake@Imperial.ac.uk

Imperial College London
Centre for Transport Studies

OPUS Urban and Regional Systems — Optimising the use of Partial Information in Systems — IST-2001-32471

Complex Systems:
It is impossible to capture all aspects of the behaviour of a complex system in a single dataset. Need to combine information of multiple types to get a complete picture of the system. Expanding sample size for a single sample design improves precision but not coverage.

Opus Methodology:
Construct a **Statistical Model** of (appropriate parts of) the real system, including variability of observations and uncertainty about parameters, as well as relationships (deterministic and stochastic) between observations and parameters. Likely to be complex.
Use model fitting processes to extract **Evidence** from different types of dataset covering different aspects of the system. Use Bayesian methods to combine the evidence. This becomes **Knowledge** that reduces uncertainty about parameters.

Results from Models:
All knowledge extracted from the data by the fitting processes about the system as seen through the model is in the final state of the model. Summarise in estimates of real-world measures, with uncertainty limits. Can also use the specification of and knowledge from the model to generate **Synthetic Data**, to allow exploration of details of the model parameters and relationships. Generation process will include randomness from both variability and uncertainty.

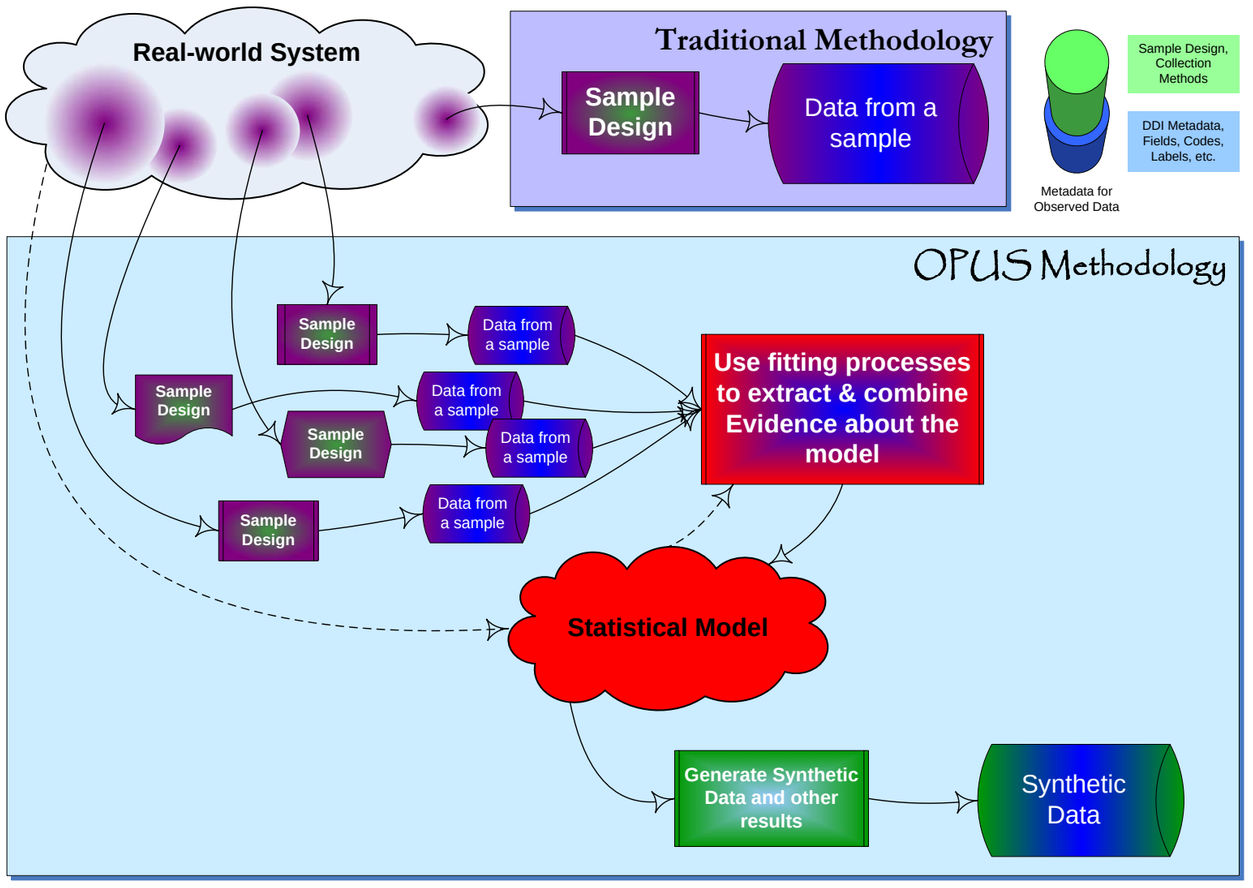
Interpreting results from Synthetic Data:
Need to understand the model, the fitting processes and the generation process in order to interpret the synthetic data correctly (as with sample design for survey data). Can store this information as **Metadata**.
Also need functionality to present the information in ways that facilitate exploration.

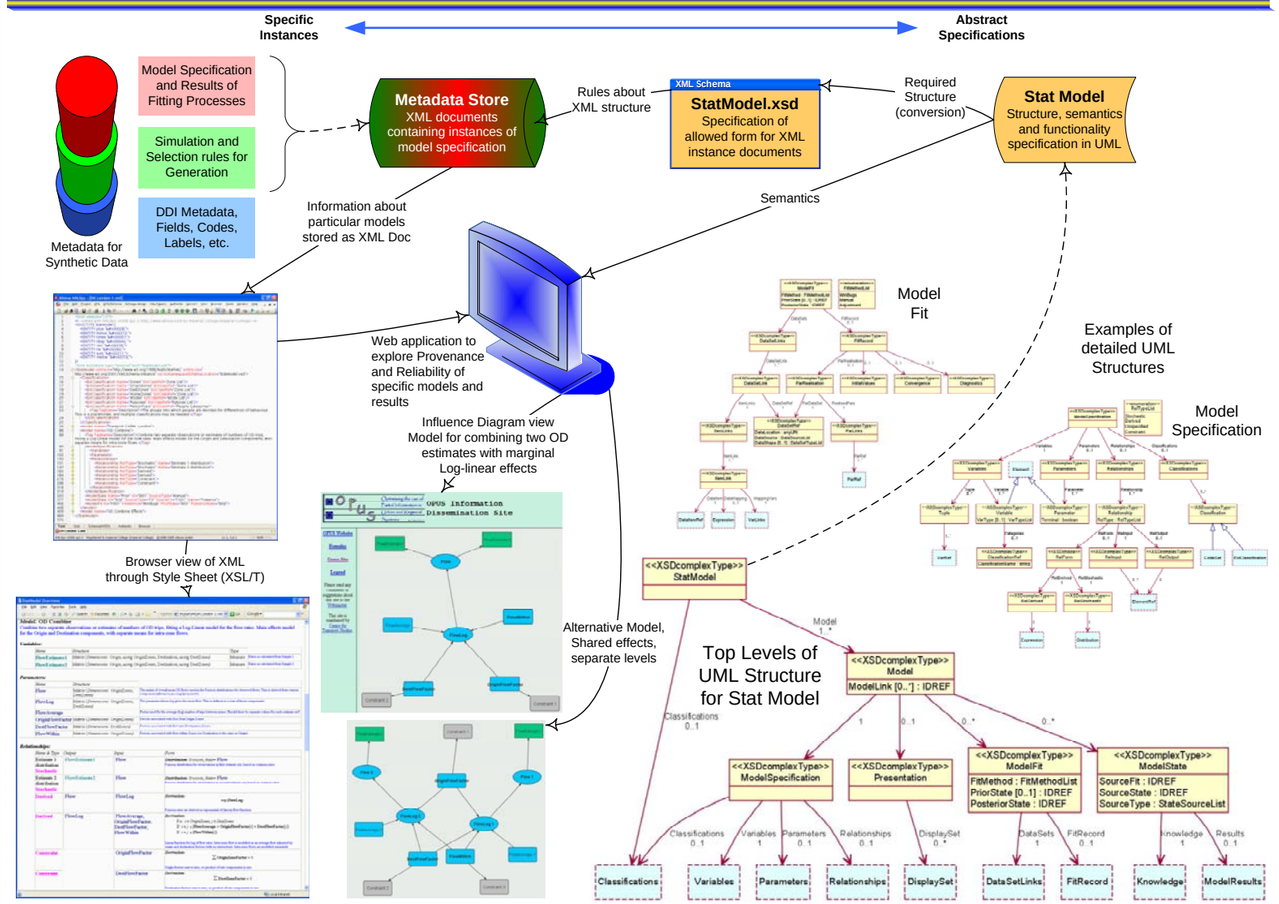
Metadata in OPUS:
All statistical data needs metadata to specify:
Content – fields, codes, labels, … – information needed for the calculation of statistical results – all in DDI standard.
Provenance – sample design, collection methods, … – things that affect the interpretation of derived statistical results – much in DDI standard.

Provenance of Synthetic Data:
Synthetic data provides a view of the knowledge in the final state of a statistical model. It does not tell us directly about the real-world system. To interpret statistical results from synthetic data correctly we need to understand the form of the statistical model, the fitting procedures used, and the rules used to generate the data.

Reliability of Results from Statistical Models:
Because the model is central, its correctness and reliability are crucial. These can be assessed by exploring the uncertainty remaining with model parameters, and the contribution (evidence) from particular model fitting steps.

Metadata for Models – Stat Model:
UML is used to design the structure, semantics and functionality for the storage and manipulation of metadata about models, including model specifications and knowledge about parameters. The metadata allows users to explore the **Provenance and Reliability** of results derived from a specific model. **Instances** of information about particular models are stored as **XML** documents, using a schema generated from the UML.
This information is presented in a **web application** that includes listings, mathematical specifications, influence diagrams and uncertainty plots.





How is it done? Methods and Tools

Statistical Models

Bayesian approach, so that knowledge can be combined – bring *prior* knowledge into each fitting step, and process yields combined *posterior* knowledge.

Construct *Generalised A-Priori Model* (GAPM) of domain as starting point, and then specialise this for fitting models to address specific issues. Many models fall into class of *Graphical Models*, with conditional independence.

Models include stochastic (statistical) components for explicit representation of the variability in observations, due to sampling, measurement processes or randomness of behaviour. Also use stochastic components to represent uncertainty (lack of knowledge) about parameters.

Mathematical relationships used to show how parameters interact with each other and in their influence on observations. Can use parameters with uncertainty where the form of relationship is not well-known.

Model Fitting

Bayesian mathematics usually too complex for explicit solution, so MCMC (*Monte-Carlo Markov Chain*) methods used, with *Gibbs Sampler* or similar. These use simulation to yield empirical estimates of the multivariate posterior distributions of parameters.

Use standard packages such as ***WinBUGS*** for Graphical Models (implies model forms an acyclic directed graph, no feedback loops). Some specialised algorithms available for more complex situations, implemented using C++ programs.

Statistical language ***R*** used as scripting and integration environment.

Transport example in OPUS poster by Logie, Lindveld and Polak. Many technical issues in implementation, particularly convergence and scalability.

Results

Joint posterior distribution of all parameters estimated by empirical (simulated) data – each record is a joint set of parameter realisations. Can be summarised by fitting statistical distributions (often just Normal with mean and SD).

Empirical distributions for other measures can be derived by applying model relationships to the full empirical posterior distribution.

Add measurement variability for synthetic data. Possible to impose assumptions at this point, with specific focus or estimates. Can also take input factors (e.g. socio-demographic measures) from a real sample.

How is it done? Metadata Models and Instances

UML Specification for *Stat Model*

HyperModel Workbench used for UML Structural model. Includes stereotypes for XML Schema and schema generation facilities, uses standard XMI structure for specification interchange with other UML packages. *Poseidon* used for extensions to the UML design. Existing structures used where possible, e.g. Datasets (*DDI*), Classifications, *MathML* for equations.

XML Processing

Generated XML Schema checked in *XMLSpy*. Presentation stylesheet (XSL/T) developed initially in *StyleVision*, then refined by hand.

Model Instances

Statistical model developed in host package (e.g. WinBUGS). XML instance documents constructed manually with XMLSpy, though future automation is possible. Equations in *MathML* constructed using *Formulator*.

Presentation Application

Web site built using Java applets in Apache Web Server, with links to *R* for statistical (distribution) displays.

References :
Gilks, W. R, Richardson, S., Spiegelhalter, D.J. (Eds.) (1996) *Markov Chain Monte Carlo Methods in Practice*
WinBUGS (free license). www.mrc-bsu.cam.ac.uk/bugs.
The *R* project for statistical computing (free). www.r-project.org.
Object Management Group – *UML2.0*. www.omg.org.
Carlson, D.A. et al. – *HyperModel Workbench* (freeware). www.xmlmodeling.com.
Gentleware – *Poseidon for UML* (free Community edition). www.gentleware.com.
DDI Alliance – *Data Documentation Initiative*. www.icpsr.umich.edu/DDI.
Altova – *XMLSpy* (free Home edition), *StyleVision*. www.altova.com
MathML – the *Mathematical Markup Language*, www.w3.org/Math/.
Hermitech Laboratory – *Formulator* (free editor for MathML) – www.hermitech.ic.zt.ua.

Acknowledgements
The work reported in this paper (and the whole Opus project) is funded as Project IST-2001-32471, part of the Fifth Framework Information Society Technologies programme of the European Community, managed through Eurostat.
Thanks to Opus colleagues Miles Logie and Saikumar Chalisani for contributions to the development of the original ideas, and to Rajesh Krishnan for the presentation application.