# Extensible efficient handling of metadata for e-Science

Andrew R. Price, Marc Molinari, Zhuoan Jiao, Simon J. Cox
*School of Engineering Sciences, University of Southampton, Southampton, Hants, SO17 1BJ, UK*

UNIVERSITY OF Southampton
School of Engineering Sciences

## Abstract

The Geodise data management platform provides a Grid-enabled distributed file and data archive and enables users to annotate their data with rich descriptive metadata in the form of XML. We present a process model for the evolution of the Geodise data management system from a free and flexible repository to a structured scalable relational database for production purposes. The model is designed to aid scientists and engineers to develop XML Schemas describing their metadata requirement which can be registered in the XML-enabled database to provide a scalable back-end solution without any impact to the existing client side application.

## Introduction

Large collaborative scientific investigation is resulting in increasingly large data collections which need to be available to distributed user communities, with rigorous access and authentication policies and efficient and reliable data delivery. In order to manage data on the petabyte scale collaborative projects are increasingly turning to Grid technology. A particular challenge in managing data in distributed environments is enabling users to identify and locate the data they want. Metadata is critically important in e-science to provide the ability to locate and characterise data, to provide provenance, enable data re-use and express community standards.

## Geodise Data Management

The Geodise data management platform provides a Grid-enabled distributed file and data archive and enables users to annotate their data with rich descriptive metadata in the form of XML typically serialised from data structures in a problem solving environment. The system exploits the Oracle *10g* XML database (XDB) to support storage, query and retrieval of XML instance documents. In the standard Geodise database system, users are free to annotate their data with any valid XML and can subsequently query their metadata to find items of interest. This provides considerable flexibility and extensibility. In contrast to systems like the Globus Metadata Catalog Service (MCS) and the Metadata Catalog MCAT of SRB the Geodise user can add new metadata attributes without adding to or updating any underlying database tables. However, this flexibility comes at the expense of scalability and a strategy for imposing structure on the user-defined metadata is required.

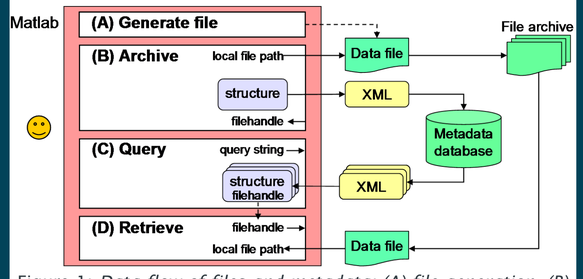


Figure 1: *Data flow of files and metadata: (A) file generation, (B) archive of file and user metadata, (C) querying of metadata, and (D) file retrieval.*

## Geodise Database Development

The raw query performance for the standard Geodise database server is critically dependent upon the XPath query engine of the underlying database solution. Unfortunately, when dealing with XML documents that have no pre-defined structure the database is essentially limited to a raw text search of the data. The performance of the standard Geodise system therefore does not scale well. We present a process model for the evolution of the Geodise data management system from a free and flexible repository to a structured scalable relational database for production purposes. The model is designed to aid scientists and engineers to develop XML Schemas describing their metadata requirement which can be registered in the XML-enabled database to provide a scalable back-end solution without any impact to the existing client side application. The process follows four steps:

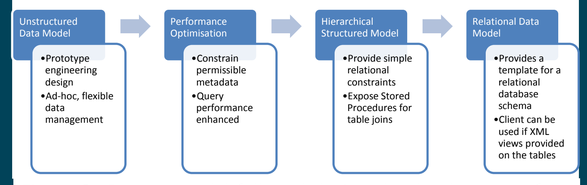


Figure 2: *Process model followed during the development of the GENIE data management system.*

1. **Unstructured storage**. Studies exploit the standard Geodise database server to archive, share and process data and collaborative study of small problems is enabled in the virtual organisation. The system supports small scale experiment and design activities but is likely to hit scaling issues if data volumes become sizeable.

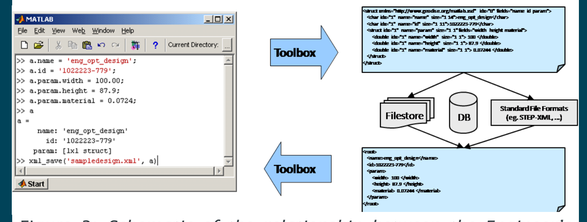


Figure 3: *Schematic of the relationship between the Engineer's problem solving environment and the database system.*

2. **Performance Optimisation**. Oracle provides the means to register a XML schema which can be used to provide storage in object-relational rows for any XML documents conforming to the schema. If such a schema can be provided then the data can be stored in tables that the SQL engine can perform optimisations upon. Subsequent query performance is significantly improved. In this phase of the Geodise database development users are encouraged to look for commonality in their metadata representations and to develop a XML schema that can be registered in the database.

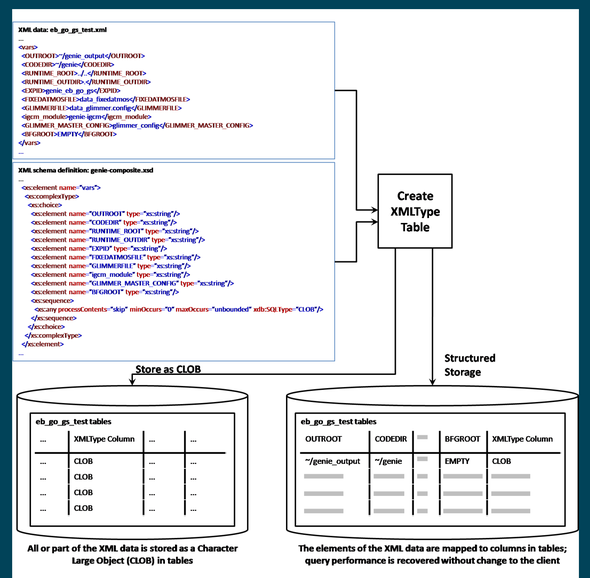


Figure 4: *Un-structured and structured management of XML data in Oracle 10g (diagram modified from Chapter 5 of the* **Oracle® XML DB Developer's Guide**). *User's identify a schema that describes their data.* `<xs:any>` *elements are used to provide extensibility as structure is gradually imposed.*

3. **Hierarchical Structured Model**. XML schemas are written to describe the encoding of information in XML for a particular domain. A useful feature of XML schema is the `<xs:choice>` element which allows the schema to assert that a valid document will contain only one of a number of defined elements. Using such a construct the schema can be written such that valid metadata may conform to one of many different data structures in the host problem solving environment. In this way the schema can define data types that are acceptable for storage in the Geodise database. In conjunction with a data grouping mechanism this provides the user with a means to easily construct related logical aggregations of data and to describe those aggregations with metadata.

4. **Relational Database Prototype**. With a suitable schema registered in the database the user can create hierarchies of datagroups, each describing a meaningful entity for the management of their domain data. To provide a consistent view on the data we provide the ability to apply constraints on the many-to-many datagroup_datagroup table that they restrict associations of XML entities and therefore enforce the desired data schema. The XML schema can also be designed such that any nodes in an instance document that are not specifically declared in the schema are stored as CLOB objects in the database and the system is therefore extensible.

## GENIE Exemplar

The XML Schema for the GENIE system is a composite of individual component code descriptions (maintained by the code developers, capturing all metadata required to describe the component and manage its input/output), entities for describing and managing Earth System model studies in a Grid environment (e.g. metadata associated with Experiments, Simulations, Compute tasks, Resources, etc.) and domain specific metadata (external data sets, observational data, etc.). The nature of the framework means that the data schema has to be extensible to allow new component codes to be added. Our system allows the developer to produce an XML schema describing their code. This is simply added to the composite XML schema document and registered in the database. The GENIE framework can archive, query and retrieve metadata conforming to the schema without any change to the client application. The web service interface to the database system also allows other consumer software to be written that can utilise the database for GENIE model studies.

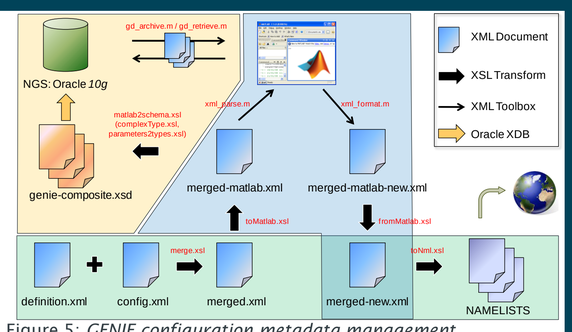


Figure 5: *GENIE configuration metadata management.*

## Conclusions

The Geodise database provides a flexible means to enable users to describe, archive, query, share and retrieve their data from the desktop. However, the costs of querying the metadata catalogue scale poorly when the XML instance documents have no defined structure. We have outlined a process model for systematic improvements to the Geodise database server to enable an extensible XML schema to be designed to improve the management of a user's domain data. By generating a schema to constrain the permissible metadata in the catalogue the underlying database can generate native storage for the conforming XML instance documents. Native database scaling performance is recovered for queries on the metadata catalogue. Through the use of the `<xs:choice>` element in the XML schema a number of data types can effectively be defined. By associating instances of these types to datagroups, the logical aggregators of the Geodise database, the user can construct meaningful data structures within the database. If a firm data schema emerges for the relationships between these types then the system can be further augmented to impose those relations through constraints.

GENIE
Grid ENabled Integrated Earth system model
www.genie.ac.uk
NATURAL ENVIRONMENT RESEARCH COUNCIL