Semantic Description of Signal and Image Databases

Pedro Contreras(1), Münevver Köküer(1), Mireille Louys(2) and Fionn Murtagh*(1)
(1) School of Computer Science, Queen's University Belfast, Belfast BT7 1NN, UK
(2) LSIIT, Ecole Nationale Supérieure de Physique de Strasbourg, Bd. Sebastien Brandt, 67400 Illkirch, France, and Observatoire de Strasbourg, 11 rue de l'Université, 67000 Strasbourg, France.

ABSTRACT

Information navigation and search on the part of a user requires thorough description of the information content of signal and image datasets and archives. Large signal and image databases need comprehensive metadata to facilitate user access. There is no unique way to describe the semantics of images and signals. Therefore a conceptual model serves as an initial platform. From the conceptual model, a database design can be derived, or a definition of metadata. The different steps from model to description can benefit from tools such as the Unified Modeling Language (UML) for the conceptual model, standard Entity/Relationship (ER) models for database design, and eXtensible Markup Language (XML) for metadata description. As examples of the process of conceptual design and semantic description, we consider the case of a signal database, and the case of astronomical image databases.

1. INTRODUCTION

A conceptual model of any signal or other database is important for reasons that include the following.

1. Separation of concerns, e.g. between data objects, processing phases, and descriptive elements, is of importance in ensuring a clean design which in turn will be of importance in maintenance of the database.
2. It is assumed that the database is not static but instead requires modification over time. Furthermore it is assumed that scalability is also of importance.
3. Collaborative work, document maintenance, and verification of coverage, are all helped greatly.
4. Cross-walking between data and metadata description frameworks is aided [2,5].

In the Fifth Framework project Aqua-Stew, a new technology is being developed which seeks to discriminate between liquid properties by measuring tensile properties. Signal traces result from this. With an increasing number of tensiograph instruments in operation, and an increasing range of liquids (related to environmental technology, medicine, beverages, oils, and so on) being analysed, the long-term metadata and data storage aspects need to be planned.

We used Power Designer which is part of the Sybase[6] Enterprise Modelling tool. It provides a lightweight interface for modelling and design. Characteristics of this package include:

1. Entity/Relationship data modeling techniques, providing conceptual and physical data models. The latter can be translated directly into scripts for generation of database tables.
2. Standard UML (Unified Modeling Language) use case, class, and sequence diagrams.
3. Code generation for languages such as Java, PowerBuilder, C++, and Visual Basic from a class diagram.
4. Support of specific characteristics of major RDBMS (relational database management systems) such as tables, columns, indexes, stored procedures, etc.
5. Database information reverse engineering into physical and conceptual data models.
6. Existing business logic – action rules – reverse engineering into a class diagram.

* Corresponding author: F. Murtagh. E-mail addresses: {p.contreras, m.kokuer, f.murtagh}@qub.ac.uk, mireille.louys@astro.u-strasbg.fr

7. Forward and reverse engineering of XML (eXtensible Markup Language) applications and class diagrams.
8. Multi-model management support, including synchronization of objects, models, and databases.
9. Multiple model reports created in one document (RTF, Rich Text Format, or HTML, Hypertext Markup Language).

2. A SIGNAL DATABASE

The following examples show (Fig. 1) a conceptual data model, (Fig. 2) a derived physical data model, and (Fig. 3), for illustration purposes, a small use case diagram.

In the conceptual model diagram (Fig. 1), <M> indicates a mandatory field. Associations are for the most part one to one (1,1) or one to many (1,n). A is a character field, and <pi> denotes primary key. Between Staff and Lab there is use of a special relationship, a dependency relationship, which passes a foreign key to provide a primary key.

The physical model diagram (Fig. 2), derived from the conceptual model diagram, uses the CODASYL standard. A foreign key is denoted fk, and a primary key is denoted pk. From this physical model diagram, a script can be generated to produce tables in accordance with any one of about 30 different DBMS standards. Such a database can then be populated.

The illustrative use case diagram (Fig. 3) focuses on data security and authorization. Use case diagrams are used to display user profiles. Use profiles, in turn, allow for

1. different usage-related views of the database,
2. different levels of authorization of access and of writing to the database, and
3. triggers, or implementations of business rules, such as for saving query results back into the database, or for backups.

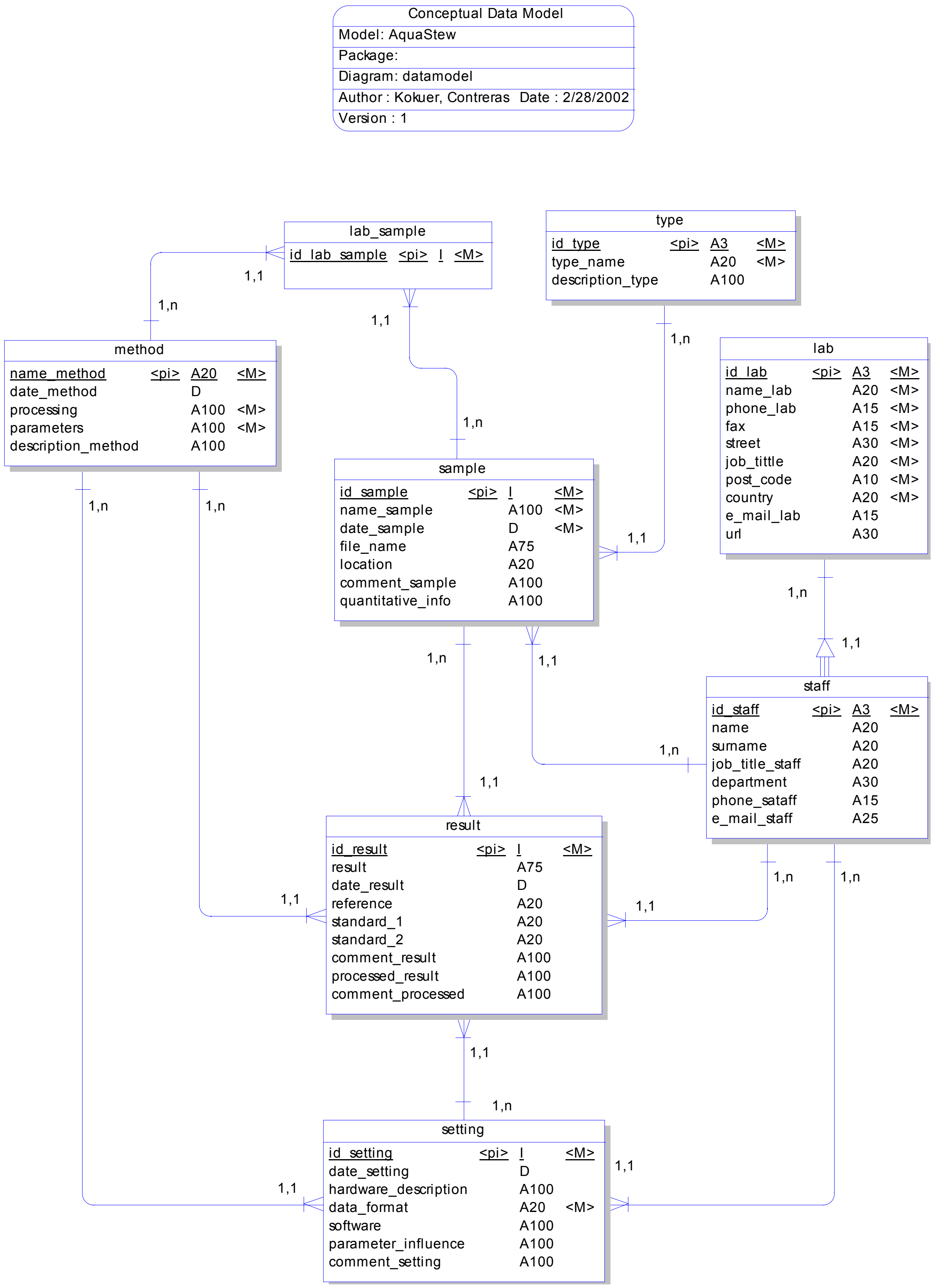


Fig. 1: A conceptual data model.

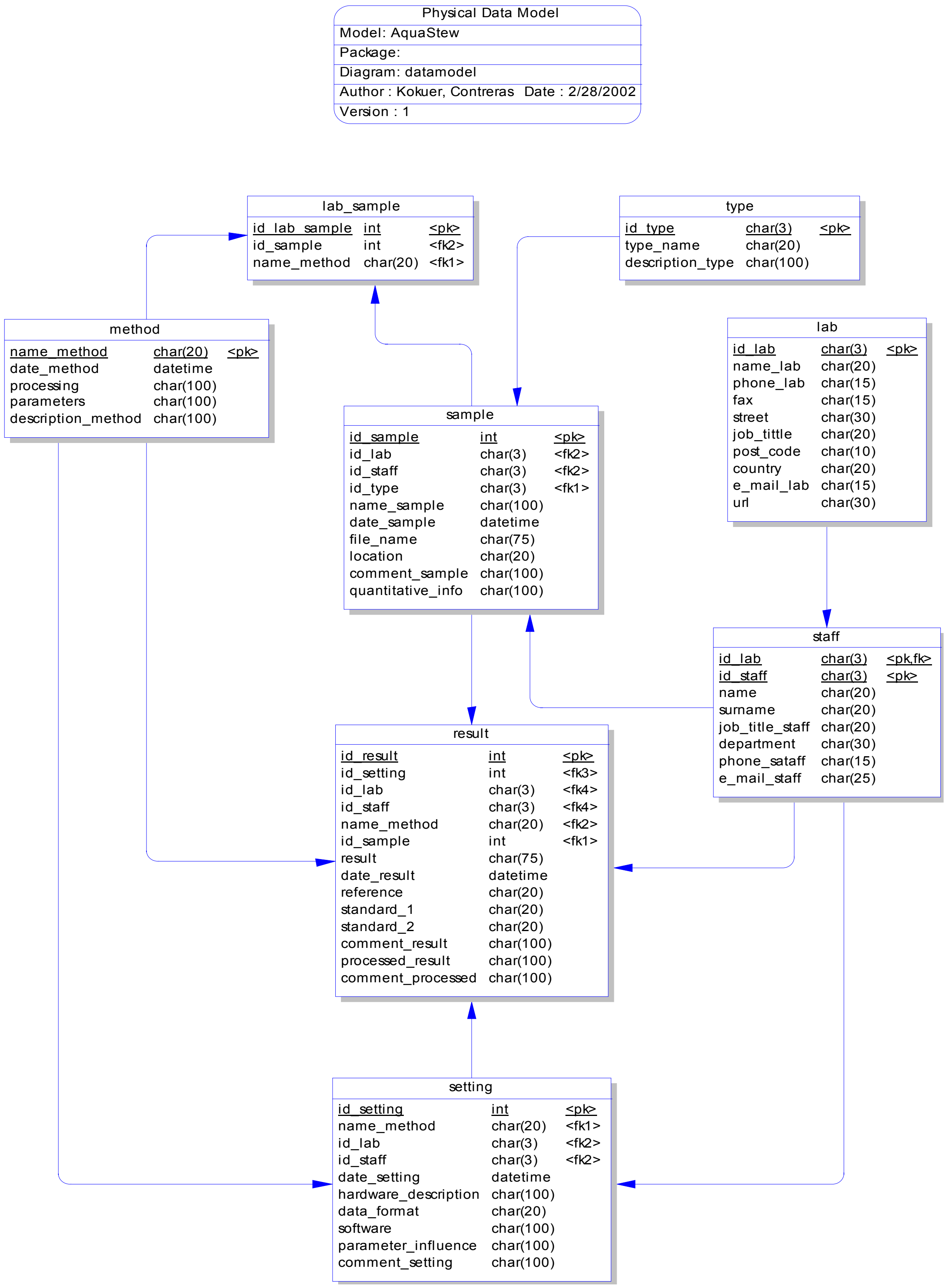


Fig. 2: A derived physical data model.

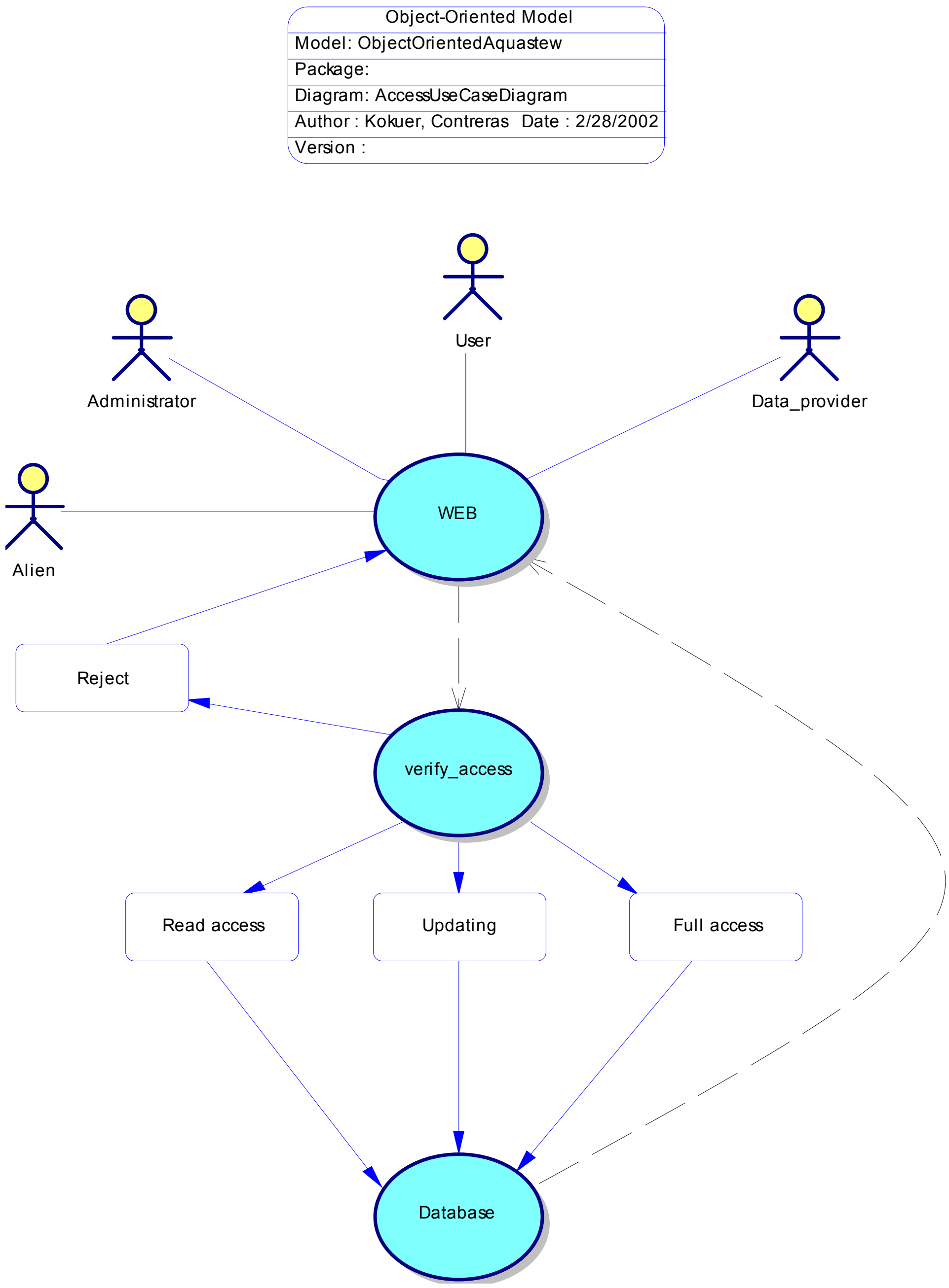


Fig. 3: A small use case diagram.

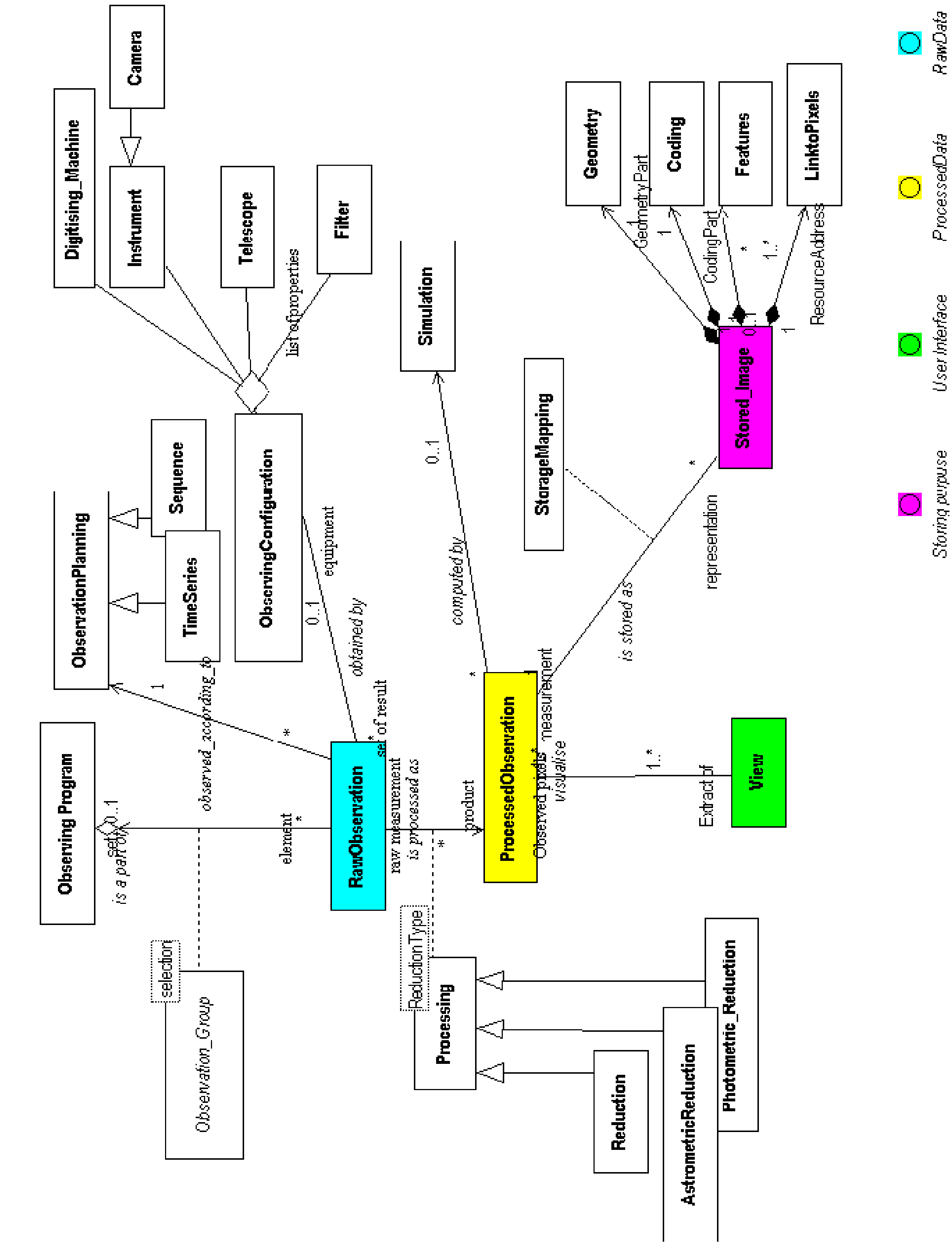
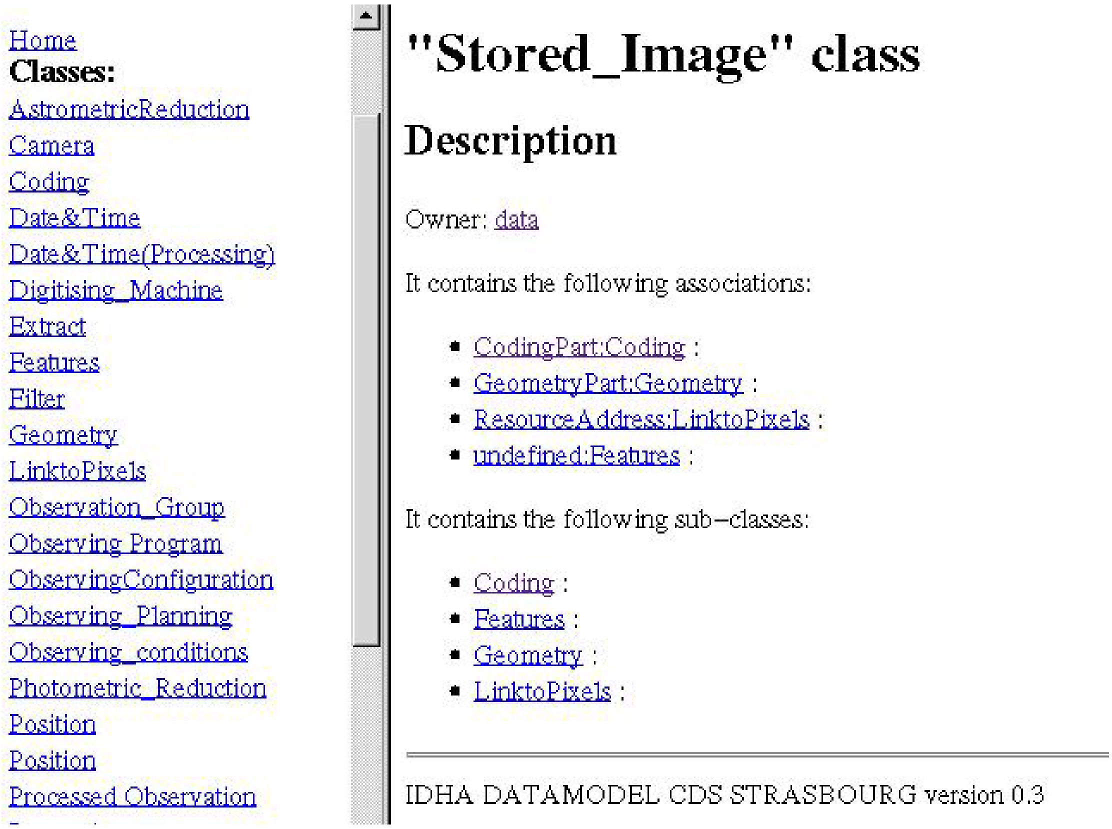


Fig. 4: Preliminary data model for image archive.



Home
Classes:
AstrometricReduction
Camera
Coding
Date&Time
Date&Time(Processing)
Digitising_Machine
Extract
Features
Filter
Geometry
LinktoPixels
Observation_Group
Observing Program
ObservingConfiguration
Observing_Planning
Observing_conditions
Photometric_Reduction
Position
Position
Processed Observation

"Stored_Image" class

Description

Owner: data

It contains the following associations:

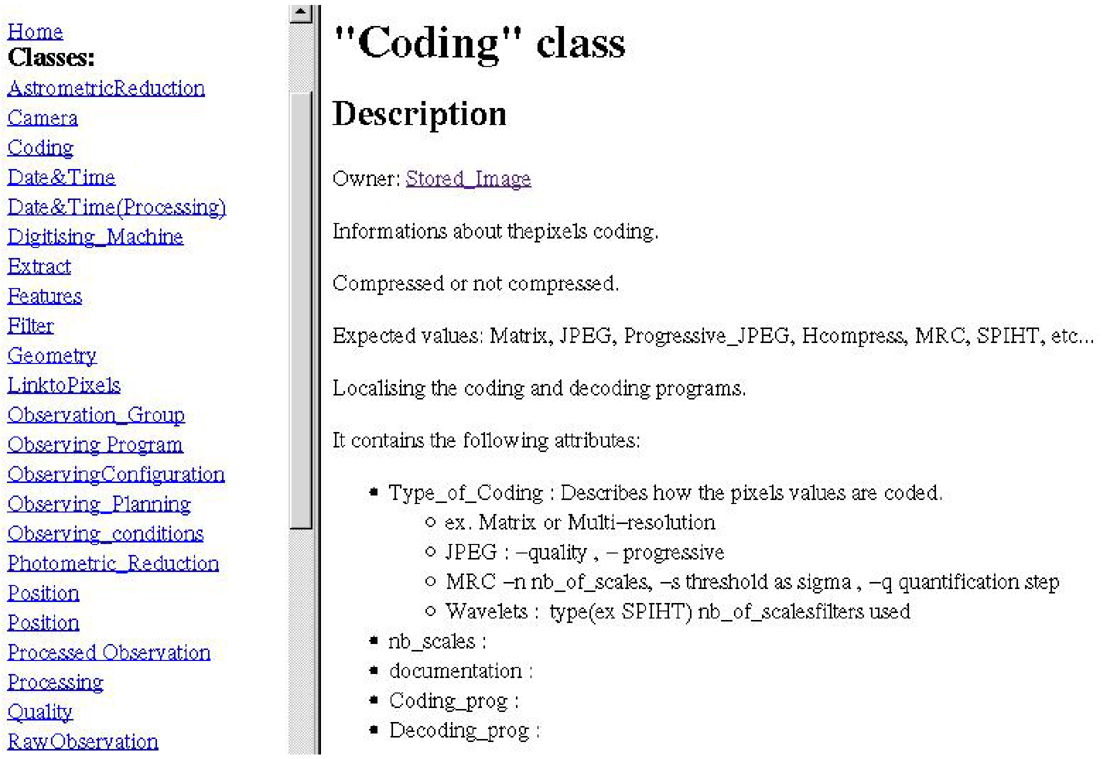
- CodingPart:Coding :
- GeometryPart:Geometry :
- ResourceAddress:LinktoPixels :
- undefined:Features :

It contains the following sub-classes:

- Coding :
- Features :
- Geometry :
- LinktoPixels :

IDHA DATAMODEL CDS STRASBOURG version 0.3

Fig. 5: The “Stored Image” class used in Fig. 4.



Home
Classes:
AstrometricReduction
Camera
Coding
Date&Time
Date&Time(Processing)
Digitising_Machine
Extract
Features
Filter
Geometry
LinktoPixels
Observation_Group
Observing Program
ObservingConfiguration
Observing_Planning
Observing_conditions
Photometric_Reduction
Position
Position
Processed Observation
Processing
Quality
RawObservation

"Coding" class

Description

Owner: Stored_Image

Informations about thepixels coding.

Compressed or not compressed.

Expected values: Matrix, JPEG, Progressive_JPEG, Hcompress, MRC, SPIHT, etc...

Localising the coding and decoding programs.

It contains the following attributes:

- Type_of_Coding : Describes how the pixels values are coded.
  - ex. Matrix or Multi–resolution
  - JPEG : –quality , – progressive
  - MRC –n nb_of_scales, –s threshold as sigma , –q quantification step
  - Wavelets : type(ex SPIHT) nb_of_scalesfilters used
- nb_scales :
- documentation :
- Coding_prog :
- Decoding_prog :

Fig. 6: The Coding class used in Fig. 4.

3. DATA MODEL FOR DISTRIBUTED IMAGE DATABASES

In Louys et al.[3] a conceptual design for an image archive is carried out using the Objecteering UML modeling tool[4]. The primary aim is to define partial views of the conceptual model in the form of XML descriptions, dedicated to different user profiles and usages. The entire range of properties of observational image, including spectral and multiband data, is addressed. This includes observing conditions, instrument characteristics, calibration applied, data analysis methods applied, and links to related resources such as catalogs, pipeline processing parameters, observing campaigns or projects, scientific mission or project or survey, and so on.

Fig. 4 shows the preliminary design, which is available online at http://alinda.u-strasbg.fr/IDHA/lastmodel. Figs. 5 and 6 show further detail, related to "Stored Image" and "Coding". A metadata standard, VOTable ("Virtual Observatory Table" standard) has been developed in recent months (see http://www.us-vo.org/VOTable) for support of astronomical catalogs or tables (i.e., relational tables). The objective of our work on the IDHA distributed image model is to provide similar support for images repositories.

4. CONCLUSION

A UML-based conceptual model provides a disciplined framework from which a range of other aspects of data handling and processing can provide input or can be derived: use case modeling, which gives priority to how users will work with the data; database design for storage; and metadata design and definition, to allow users to find information and then to manipulate it.

In further work in the Astrogrid[1] project, the Together UML modeling tool[7], is being used for use case modeling, to be followed by the design and implementation of Grid-based web services.

ACKNOWLEDGEMENT

The signal database work is supported by the Fifth Framework project "Aqua-Stew – Water Quality Surveillance Techniques for Early Warning by Interface Sensors". The image database work is supported by the French Government ACI-Grid project (Action Concertée Incitative – Globalisation des Ressources Informatiques et des Données) IDHA – Images Distribuées Hétérogènes pour l'Astronomie, 2001-2003.

REFERENCES

1. Astrogrid, www.astrogrid.org.
2. S. Cranefield and M. Purvis, "UML as an ontology modeling language," 8pp., 2002.
3. M. Louys, F. Bonnarel, P. Didelon, P. Fernique, F. Genova, F. Murtagh, F. Ochsenbein and M. Wenger, "IDHA image archive model," *Toward an International Virtual Observatory - Scientific Motivation, Roadmap for Development and Current Status*, Garching, Germany, June 2002. http://alinda.u-strasbg.fr/IDHA/lastmodel
4. Objecteering UML Modeling Tool, www.softeam.fr.
5. D. Skogan, "UML as a schema language for XML based data interchange," Proc. 2nd International Conference on the Unified Modeling Language, UML'99, 1999.
6. Sybase, www.sybase.com.
7. Together, www.togethersoft.com.