

## PROBLEMS OF STORING ADVANCED DATA ABSTRACTION IN DATABASES

Kelly Nunn-Clark<sup>1</sup>, Lachlan Hunt<sup>1</sup>, Teo Meng Hooi<sup>1</sup> &  
Balachandran Gnanasekaraier<sup>1</sup>

<sup>1</sup> School of Information Technology, Charles Sturt University

**ABSTRACT:** There has been for some time, and will continue to be an increasing need for the storage and retrieval of complex data structures. Databases are already a major part of the operations for organizations and businesses; however, they are now increasingly being used for research purposes, for storing complex data structures that arise but are not reflected in the business paradigm. The aim of this paper is to discuss the need for, and the use of, object-oriented databases for storing complex data structures and linked images, specifically focusing on a current research application — the design of a database for use with the Akamai Vision System. To achieve this aim we will investigate the current methods used within the Akamai prototype for the storage and retrieval of its data; as well as consulting other relevant literature on the storage and retrieval of objects. We will investigate the current use of object-oriented databases within other large research projects; including the reasons for using the object-oriented model rather than other alternatives, such as relational- and hybrid object-relational databases. We will also examine the options available for object-oriented database management systems, briefly discussing the advantages, the disadvantages, and some examples of the current usage of each. The paper will present an analysis of the way complex data structures are managed within the Akamai Vision System and how an object-oriented database system will facilitate this. Finally we will present our design for the new database, which will provide optimized access to the data, and enable it to be used within the system to learn and generate additional information.

### INTRODUCTION

The purpose of this Akamai Vision System Database project is to provide a suitable data management system for the Akamai Vision System (AVS). The AVS is an artificial intelligence Program designed to assist doctors with the analysis of images to find anomalies such as tumours or micro-calcifications in mammograms. The first section of the paper will describe the types of database models available, and briefly describe what each is useful for. In the second section, we will describe other research projects, and give their reasons for using object-oriented databases. The third section will be an overview of our research project, the AVS, and the current methods employed for storage and retrieval of data. We will explore the shortcomings of relational and hybrid object/relational models with regard to the AVS and the perceived benefits of utilizing an object-oriented database model. We will describe our current work underway in designing a database for use with the AVS including both the windfalls and the pitfalls we have encountered and their potential solutions. In the fourth section, we will examine the Object Oriented Database Management System (ODBMS) we have chosen, briefly discussing the advantages, the disadvantages, and some examples of its current usage in other projects. Finally, we hope that the lessons we have learned from researching and designing an object-oriented database system will increase the knowledge and understanding about the existence and benefits of object-oriented database models.

### DATABASES

Databases are already a major part of the operations for organizations and businesses; however, they are now increasingly being used for research purposes, for storing complex data structures that arise but are not reflected in the business paradigm. While many businesses are still turning to relational models, there are some shortcomings found when applied to research areas. As a result, many research organisations are implementing object-oriented databases to solve their problems of data storage, retrieval and processing.

Relational databases can store simple data types within tables and can manage complex relationships between these data types. Hybrid "object-relational" databases are seen as enhanced relational databases in that all persistent (database) information is still in tables; however some of the tabular entries can have richer data structure. Although this approach provides more structure for modelling more complex data and freestanding functions, minimal queries involving object attributes are possible because it lacks the fundamental object requirement of encapsulation of operations with data [M97]. Both of these models are well suited to businesses where the level of complexity that software must model and support is low. However, neither of these database models can effectively manage the storage, retrieval and relationships of complex data structures in an easy manner, consistent with the needs of many large research projects, particularly projects built upon an object-oriented or object based architecture [ER98]. Pure ODBMSs are evolving to fill these requirements of the research community by developing high-performance storage mechanisms that are compatible with object models. These databases integrate object-oriented programming with database capabilities, allowing complex data structures modelled to hold data and rules about how the data can be accessed and manipulated. As a result, more research organisations are implementing object-oriented databases to solve their problems of data storage, retrieval and processing. With the design and implementation of an object-oriented database we hope to show that they are an effective way to manage and process complex data structures.

#### CURRENT PROJECTS USING OBJECT-ORIENTED DATABASES

There are many projects in progress using object-oriented database technology for the storage, processing and retrieval of their data. These include the Biomedical Imaging Resource (BIR), which has many similarities to the AVS, and the Globally Interconnected Object Databases (GIOD) which addresses the data storage and access problems posed by particle physics with their particle colliding experiments.

##### Biomedical Imaging Resource

The BIR was started by the Mayo Foundation and has become a repository for a large number of two- and three-dimensional images [SAR98]. Although the purpose of the BIR differs from that of the AVS, it is similar in that it is a database designed for the storage and retrieval of medical images. Initially, the system began as an application that directly accessed images from the directories they were stored, in the UNIX file system. The users of the system added images to these directories, most of which had different methods of naming the files, and keeping their own personal notes, on where the files are, in their own notebooks. This causes a problem when other users wish to continue work previously done by another user, because it is difficult for the new user to find the correct image. Another problem encountered was that moving all images from their existing location to within the database would interfere with current operations. The solution to this problem in the BIR was to continue adding the images to the directories as normal; however, a program was written to *capture* these images along with the metadata about these images from where they were located in the file system. The actual image files are left intact in the file system, and are linked to by using the information collected and stored. This creates a problem if the image is moved from the hard drive to a tape archive; however, at the time this information was written, research was still being done to develop a multi-level storage system.

##### Globally Interconnected Object Databases

The GIOD joint project between Caltech, CERN and HP will be addressing the data storage and access problems posed by the next generation of particle colliding experiments which will start at CERN in 2006 [B03].

The scientific goals and discovery potential of the experiments will only be realized if efficient worldwide access to the data is made possible.

Particle physicists are thus engaged in large national and international projects that address this massive data challenge, with special emphasis on distributed data access.

There is an acute awareness that the ability to analyse data has not kept up with its increased flow. The traditional approach of extracting data subsets across the Internet, storing them locally, and processing them with home-brewed tools has reached its limits.

They tested usability and performance of Versant ODBMS, and concluded that Versant would offer an acceptable alternative solution to Objectivity, if required.

More information on the GIOD project can be found at <http://pcbunn.cithec.caltech.edu/default.htm>

## THE CURRENT SYSTEM

Akamai is a vision system being developed as a cooperative research project between Griffith University, University of Technology Sydney and Charles Sturt University, to implement a distributed and collaborative image processing system for medical imaging. Created by Dr. Phil Sheridan, Akamai is an artificial vision system implemented in software. Eventually Akamai will be utilised as a companion to doctors and radiologists alike to assist in the correct classification of concepts like tumours in mammograms.

Akamai incorporates many computational principles inherent in the human vision system [SHA99]. One of Akamai's unique features is its ability to learn and see from examples captured through a graphical user interface. By loading images into the Akamai application (e.g. mammogram or x-ray) Akamai is then taught by showing what correctly constitutes an example of a concept (eg. tumour). Each object on the image is classified and stored as an example. The learning is then done on the system whereby Akamai generates a rule based on the captured examples. As more and more examples are captured, Akamai's rules for a particular concept evolve – becoming more accurate.

This work was done without much regard to where the images, examples, rules and accompanying metadata were stored. For this reason the file system used is very poorly structured, and in desperate need of reform. Currently, image files are simply stored in a few directories, accessed directly by the application. This is also true for saved examples and generated rules. There is no method for searching these images, and no record of which images were used. For testing purposes these images are rarely used, since most of the images used are randomly generated and not stored at all. However, as the AVS moves from a prototype to a full application, this will not be the case — images; predominately mammograms, will come from real patients, and be stored permanently. Thus, there is a need for a persistent method of storage to manage, store and retrieve Akamai's images, examples, rules and metadata to form the foundation for AVS. Without this Akamai would be hindered in its future goals toward creating a distributed, collaborative application. With this in mind, we aspire to choose a database model that meets the following requirements:

- Scalable: Distributed Akamai is planned for the not to distant future;
- Reliable: A necessity to build confidence in this system;
- Secure: Safety of stored data structures will be paramount from inception of project;
- Speed: Retrieval and storage of data structures. The database system must be fast enough to increase confidence and usability from the user's point of view; and
- Capable of dealing effectively with complex data structures.

Benefits of using ODBMS for Akamai

Object-oriented databases:

1. Are optimised to support object-oriented applications.
2. Support any type of structure including trees and composite objects.
3. Give each object a unique immutable ID.
4. Support complex data relationships.
5. Allow programming code (eg. Java) to be used to program both the application and the database; there will be no need to translate from application to database, using a language such as SQL. [AG92]

There are two primary benefits in using this technology. Both benefits reflect a basic idea, when we use an ODBMS, the way we use our data is the way we store it. The first benefit can be found in development: When we use an ODBMS, we will write less code than if we were writing to a relational database management system (RDBMS). In many cases, this code is as much as 40 percent less. The corollary to this is that any data structure that we can imagine in Java or C++ can be stored directly without translation in an ODBMS. Since the AVS was written in Java, these benefits will help to achieve the aim of the project.

The second benefit occurs in production. If we are working with complex data, an ODBMS can give us performance that is ten to a thousand times faster than an RDBMS. This is because, when the data is read off the disk, it is already in the format that Java or C++ uses — no translation is needed. The range in performance gain depends on the complexity of the data. We require a very high performance database for the AVS since it is dealing with a large number of images.

#### Current Progress for the Akamai Vision System

As of now, we have completed the first four phases of the software development life cycle. They are: Project Initiation and Approval Phase, User Requirements Definition Phase, Software Requirements Definition Phase, and Architecture Design Phase. All the documents that we have produced follow the IEEE Standard for Software Engineering [IEEE98].

Following the Project Initiation and Approval Phase, we moved on to the creation of our Draft User Requirements Document (Draft URD), which briefly outlines the requirements stated by the client. It was also during this phase that we began to look at the types of databases that we might use for Akamai. Due to the nature and complexity of the data to be stored, we realized that the best option was to use an object-oriented database.

In the User Requirements Definition Phase, a much more in-depth study on the Draft URD was done to determine the functional requirements required for the Akamai database. This includes the creation of various unified modelling language (UML) diagrams to help us understand the interactions between the various objects within the database. During this phase, we began to look at the various types of ODBMS available to determine which one is most suitable for our project.

In the Software Requirements Definition Phase, based on the requirements defined in the URD, we proceed with the creation of the Software Requirements Document (SRD). The SRD defines the requirements for the database that we are currently developing. Once again, new UML diagrams were created and existing diagrams were updated to aid us in understanding the system requirements more thoroughly. It was during this phase that we selected ObjectStore as the database to be used for Akamai. This was for various reasons including its capabilities and interface for the java language, which Akamai was written using, but also because of the availability of the ObjectStore software.

In the Architectural Design Phase, the Architectural Design Document (ADD) was drawn up based on the requirements defined in the SRD. The ADD defines the main architecture for our database. Along with the ADD, various new UML diagrams were created to aid in the understanding of the architectural design.

#### Database Design

As a result of the previously mentioned project phases, we developed the following architectural model of the AVS database requirements. The whole system is structured around four core components — Images, Users, Image Types and Concepts. The other objects within the system are the metadata associated with each of the core components. An image is the object that represents the actual image file. A final decision as to whether the image will remain in the operating system directory structure, or whether it will be incorporated directly as binary data in the object, is undecided. A user is a user of the system. There are three types of users: Administrator, Manager and General User. These three users correspond, respectively, to the three different modes of Akamai: *God-like Mode* (The ability to create, change or delete anything), *Man Mode* (The ability to create users, Image Types and Concepts, and add images and its metadata just as in Evolution Mode), and *Evolution Mode* (Where the system learns, causing its rules to evolve by what is being taught by the user). Image Types are categories of images such as mammograms or x-rays, and Concepts are the anomalies found within those Images Types, such as tumours or micro-calcifications.

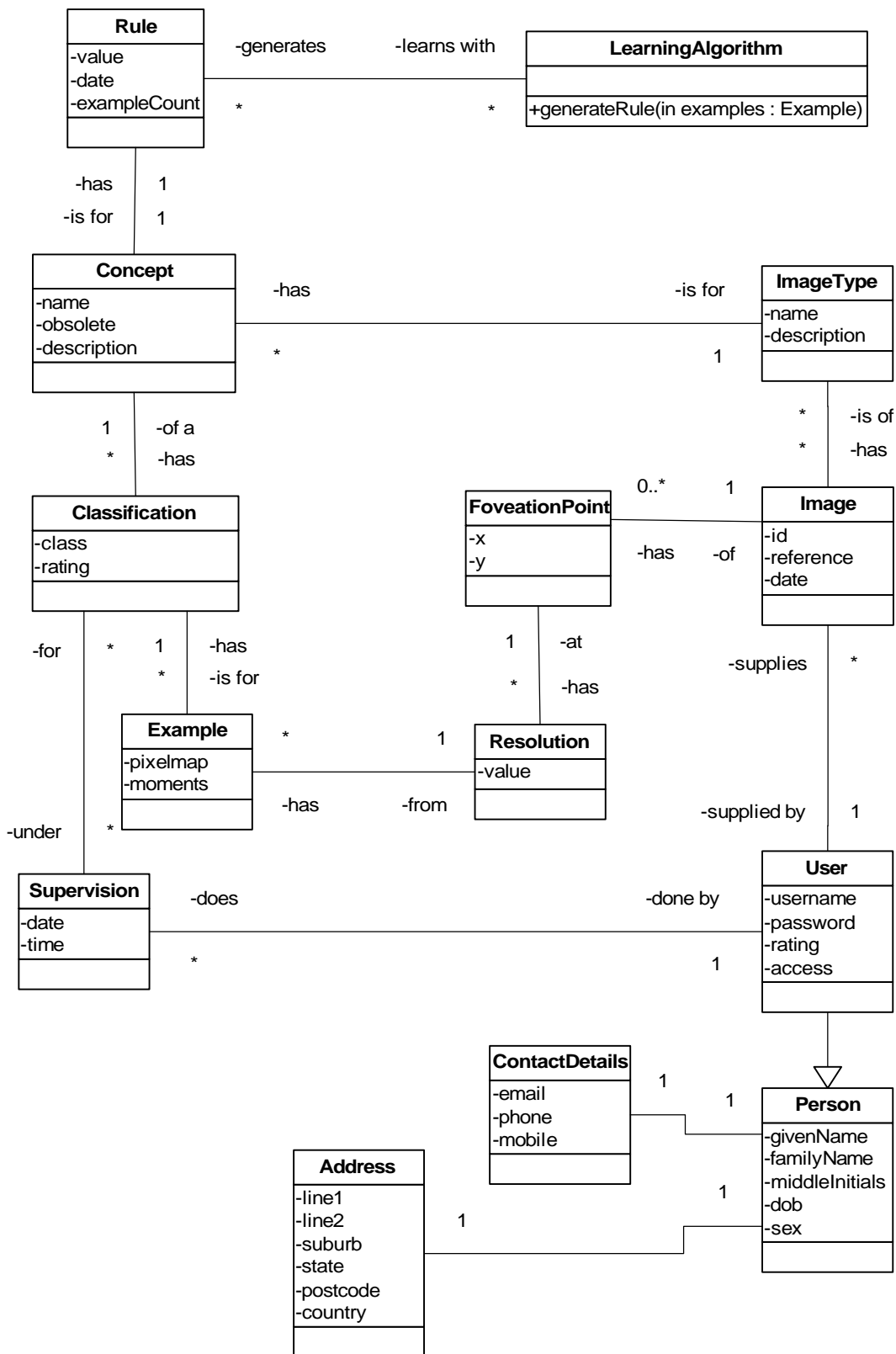


Figure 1: UML Diagram showing the architectural design of the database for the Akamai Vision System

## LESSONS LEARNED

Our approach at the inception of this project was to design and implement an object-oriented database that looked and acted like a relational database. This was due to ignorance about the capabilities of an ODBMS. We were thrilled to find out that this database system is more like an extension of the program application. We get to program the database components in Java just as you would program any Java objects. Designing with this in mind should allow us to become more efficient in programming, and give our system a better object-oriented structure.

## ACKNOWLEDGEMENTS

We would like to thank Dr Phil Sheridan of Griffith University, Dr David Tien and Dr Wayne Moore of Charles Sturt University for their supervision and guidance given to us for the Akamai Database Project.

We would like to acknowledge Progress Software Corporation for providing us with 'ObjectStore' - High performance data management software for the duration of our project.

## REFERENCES

- [AG92] Andleigh, P., Gretzinger, M., *Distributed Object-Oriented Data-Systems Design*. PTR Prentice Hall, New Jersey, pp.87-88, 1992.
- [B03] Bunn, J. *The GIOD Project*. 19 June 2003, Caltech, CERN and HP Joint Project <<http://pcbunn.cithep.caltech.edu/default.htm>>
- [ER98] Eaglestone, B., Ridley, M., *Object Databases: An Introduction*. McGraw-Hill Book Company, pp. 332-342, 1998
- [IEEE98] Institute of Electrical and Electronics Engineers, Inc., *The IEEE Standards for Software Engineering, 1999 Edition*, New York, NY, October 1998
- [M97] McClure, Steve. Object Database vs. Object-Relational Databases. IDC Bulletin #14821E – August 1997.
- [SAR98] Stacy, M., Augustine, K., Robb, R., *An Object-Oriented Image Database for Biomedical Research*. Object Databases in Practice, pp. 95–111, 1998.
- [SHA99] Sheridan, P., Hintz, T., Alexander, D., *Pseudo Invariant Image Transformations on a Hexagonal Lattice*. Paper University Technology Sydney, 1999