# MBARI's Video Annotation and Reference System

B. M. Schlining and and N. Jacobsen Stout
Monterey Bay Aquarium Research Institute
7700 Sandholdt Road
Moss Landing, CA 95039 USA

*Abstract*-**The amount of video information used for scientific purposes is increasing as new ocean observing systems are deployed. To date, underwater video has primarily been collected by SCUBA divers, remotely operated vehicles (ROV) and human-occupied submersibles. However, large quantities of video information are likely to be collected by other platforms in the future, such as cabled observatory systems (for example, the VENUS, MARS and NEPTUNE projects) and autonomous underwater vehicles (AUV). This video information is extremely useful for science but locating events or objects of interest from hundreds or thousands of hours of video requires a system for effectively managing video annotation information.**

**The Monterey Bay Aquarium Research Institute (MBARI) uses high-resolution video equipment to record over 300 remotely operated vehicle (ROV) dives per year. Over the past sixteen years, 16,000 videotapes (equates to over 13,500 hours) have been archived and managed as a centralized institutional resource. MBARI has developed a software and hardware system, Video Annotation and Reference System (VARS), to facilitate the creation, storage, and retrieval of video annotations based on ROV dive tapes. The VARS components reference a knowledge database of over 3500 biological, geological and technical terms. This hierarchical information allows for consistent and rapid classification, description, and complex querying of objects observed on video. MBARI currently has over 1.8 million annotations stored in the VARS system. These annotations have been used in over 200 professional publications and presentations, in addition to numerous education and outreach projects. These publications span multiple sub-disciplines within biology, geology and chemistry.**

**An operational VARS suite consists of a relational database, a videotape recorder (VTR) that supports the Sony 9-pin protocol over RS-422, and the three VARS applications: Knowledgebase, Annotation, and Query. The relational database is used as a centralized repository for storing the knowledgebase data and annotations. The VARS applications interact directly with this database and many VARS users can access the database simultaneously. Each VARS application is responsible for different aspects of the information stored in the database. The Knowledgebase application is used for managing the controlled vocabulary for creating and searching for annotations. Annotations and images captured from the video are added to the relational database through the Annotation application. The Query application provides simple, convenient access to the annotations, video images, and related ancillary data. The VARS applications use a shared API for interacting with the relational database. This API could be adopted by other software that makes use of video annotations, such as automated event detection applications.**

**In 2005, VARS was released as open-source software. VARS was designed so that other institutions could easily adopt it. Written in Java, VARS can be run on a variety of operating systems. In addition, VARS is not tied to a particular database; groups interested in deploying VARS can use one of a number of relational database systems. The following paper will give an overview of the capabilities and technical aspects of the VARS system. More information about VARS as well as a VARS demonstration application can be obtained at http://vars.sourceforge.net.**

## I. Introduction

### A. *Underwater Video*

As technologies have improved and the costs of deploying systems capable of collecting underwater video have declined, video has become a common method for data acquisition in ocean sciences. A search on Google Scholar [1] for 'video sea' returns over 64,000 references. In the next few years, several extensive ocean observatories will be deployed (e.g. the VENUS, MARS, and NEPTUNE projects). With abundant power, high data throughput, and 24/7 operations, these observatories could collect vast amounts of video information. This data is extremely useful for scientists but locating events or objects of interest from thousands of hours of video requires an effective video *annotation* management system. At its most basic, a video annotation is composed of 3 things: a reference to a video source, a description of some object of interest, an index to locate this object in the video source. Most video is recorded onto videotapes, so the video source is a unique identifier for a particular tape; the index is typically a time code. In order to associate an annotation with other oceanographic data including navigational, chemical, and physical properties, the precise recorded time is essential.

### B. *Video Annotation and Reference System*

The Monterey Bay Aquarium Research Institute (MBARI) uses high-resolution video equipment to record over 300 remotely operated vehicle (ROV) dives per year. More than 16,000 videotapes (equates to over 13,500 hours) have been archived and managed as a centralized institutional resource since 1989. MBARI has developed a software and hardware system called *Video Annotation and Reference System* (VARS) to manage this abundant video collection. VARS allows researchers to create, store, and retrieve video annotations based on the ROV dive tapes. VARS is a third generation system, incorporating lessons learned from early prototypes.

In 1989, MBARI began recording video from all dives performed with the ROV *Ventana*. During post-dive processing trained staff watch the video and make annotations of all events and objects of interest. The first generation annotation system simply used an off-the-shelf program that captured time code and allowed for free-text entry. The most significant problem with this approach was that the

annotations were not consistent. The same organism type could be annotated a number of different ways, such as with the common name, genus, genus and species, family designation, etc.; misspellings were also common. This made searching and retrieving annotations for specific animals nearly impossible. Another major issue was that the taxonomic hierarchy was not represented. For example, a query for 'squid', would not return any squid that were annotated with a genus or species name (e.g. *Loligo opalescens*), thus requiring every researcher to be an expert on every possible naming convention.

Recognizing the shortcomings of a free-text approach, MBARI began developing a customized video annotation system called VICKI (Video Information Capture with Knowledge Inferencing). The key enhancement to this generation was a hierarchical knowledgebase that constrained the terms that annotators could use. This hierarchy also allowed phylogenetic trees to be represented so that more powerful searches could be performed. The central data type of the knowledgebase is a *concept*. A concept is represented by one or more names; one of these names is designated as a primary name. For example a single concept might contain the names *Sebastes*, rockfish, rockcod, and rosefish. The knowledgebase also allows for detailed descriptors to be added for each concept such as taxonomic history, physical descriptions, references, and images. Using the VICKI annotation interface, users were able to begin creating and retrieving more consistent and meaningful identifications and descriptions.

There were, however, two significant issues with VICKI: 1) it was written in Smalltalk [2], which quickly became obsolete and nearly impossible to maintain and 2) it wrote annotations to a file. Writing data to files created a data management system that proved very inefficient. In order to search the annotation data the files had to be loaded into a database. If the annotations needed updating, the files would have to be unloaded from the database, updated within VICKI and then reloaded into the database. This required a complicated versioning control system that became exponentially difficult to maintain.

In 1998, MBARI began a rewrite of the annotation system using lessons learned from the two previous systems. Some of the design goals for this system included: cross-platform compatibility, maintainability by MBARI software engineers, a customizable user interface to accommodate different annotator styles, refined data management capabilities, and most significantly, data that was written directly to a database, tracked with precision, and instantly accessible to users. This system was also to be developed in a manner that would be distributable to other institutions and users outside of MBARI.

## II. Overview of VARS

### A. VARS User Interfaces

VARS consists of a suite of three applications: Knowledgebase, Annotation, and Query. These applications each interact with a shared relational database server in order to perform a distinct function. They are written in Java [3], which allows VARS to run on different operating systems without recompiling or modifying the source code. However, some components of VARS use the QuickTime for Java API [4], which is only available on Windows and Mac OS X, and so the full functionality of VARS is only available on those platforms.

The Knowledgebase application (Fig. 1) is used to create, maintain, and modify the knowledgebase data. The knowledgebase data function as a dictionary and thesaurus of terms, relationships and descriptors; annotations can only be made using the terms found in the Knowledgebase. To preserve data integrity, user permissions can be set to administrator, maintainer, or read-only access. Read-only users are not allowed to make any modifications to the knowledgebase. Maintainers can make changes but these changes are not finalized until approved by an administrator. Administrators have full control of the system.

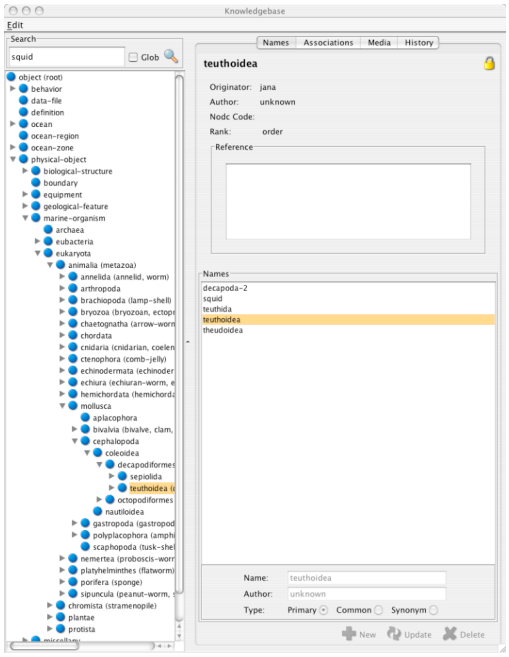


Figure 1. The VARS Knowledgebase application

The Annotation application (Fig. 2) features a customizable user interface; concepts can be dragged from the Knowledgebase onto tabbed panels allowing the user quick access to frequently used terms. These customizations are saved into the shared database to preserve a user's configurations regardless of locale. Users can annotate using any of the terms stored in the Knowledgebase; however, the Annotation application resolves the primary name for a particular annotation and stores it in the database. A series of events occurs when an annotation is made: first the time code is read from the VCR; it is then associated with the selected concept, and then immediately written to the shared database.

Figure 2. The VARS Annotation application

With the proper hardware and storage installed, the Annotation application can also capture images from the video. These images are associated with an annotation and can be viewed immediately in the annotation application or retrieved later using the query application.

Annotations can be retrieved using the Query application. The Query application allows for complex querying and sub-setting of the annotations. Researchers can search using common names or synonyms and ancillary data, such as position, salinity, temperature, etc., that are associated with an annotation can be retrieved. At MBARI, the Query application is available using Java Web Start [5]. This allows the Query application to be deployed throughout the institute without requiring the end user to manually install any software applications. The results of a query, including textual data and images, can be saved to the local computer. MBARI makes a subset of the archived video annotations available to the public; the public Query application can be launched from http://varspub.mbari.org/webstart/varsquery.jnlp.

*B. VARS Data Model and API*

The VARS data model has two components. One models the Knowledgebase information (Fig. 4), the second models the annotations (Fig 5).

The VARS API can be used for custom applications and processing. Each object follows the standard 'JavaBean' conventions [6] of using 'get' and 'set' methods to access properties. The API provides a set of data access objects (DAOs). Simple DAO method calls are used for querying, inserting, deleting and updating data in a database. Some examples of custom applications might be: importing data from automated event detection applications into VARS, combining data from oceanographic instruments with VARS annotations, or web applications that may act as an interface to VARS annotation or knowledgebase data.

Figure 4. The knowledgebase data model

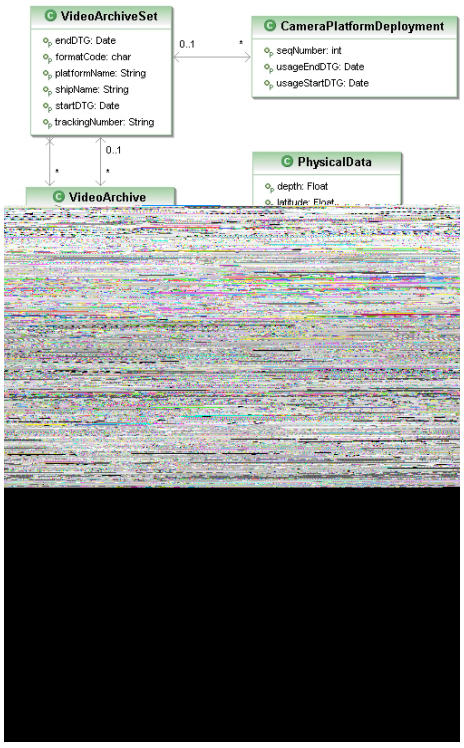


Figure 5. The annotation data model

## ACKNOWLEDGMENT

The authors would like to thank MBARI employees, Kyra Schlining for her editing assistance; Andrew Chase, Kevin Gomes, and Mike McCann for their valuable contributions to the software development; Lonny Lundsten, Susan von Thun, and Linda Kuhnz for feedback and testing of the VARS system.

## REFERENCES

[1] Google, "Google Scholar", *http://scholar.google.com/*, 2005.
[2] A. Goldberg, D. Robson, *Smalltalk-80: the language and its implementation*. Boston: Addison-Wesley, 1983.
[3] K. Arnold, J. Gosling, *The Java programming language*. New York: Addison-Wesley, 1998.
[4] Apple Computer, "QuickTime for Java", *http://developer.apple.com/quicktime/qtjava/*, 2003.
[5] Sun Microsystems, "Java Web Start Technology", *http://java.sun.com/products/javawebstart/,* 2002.
[6] Sun Microsystems, "JavaBeans", *http://java.sun.com/products/javabeans/*, 1994.
[7] Free Software Foundation, "GNU Lesser General Public License", *http://www.opensource.org/licenses/lgpl-license.php*, 1999.
[8] W. Guttmann, "Castor JDO – An Introduction", *http://www.castor.org/jdo-introduction.html*, 1999.
[9] Apple Computer, "QuickTime", http://developer.apple.com/quicktime/, 2006.