Next-generation Annotation Management

Brian Schlining
A Little History
A Little History

circa 2005

RDBMS

Cross-platform

Video Annotation and Reference System
A Little History

circa 2005

RDBMS

Cross-platform

Java

Frame capture hardware

Deck control via RS422

videotape

VARS was designed for...
Today VARS is used for .. with some caveats

RDBMS

Cross-platform

jav

videotape

Video Files

Image Collections
Image collection at MBARI
Lessons Learned

- Spelling Matters
  - Constrained vocabulary required!
Lessons Learned

- Spelling Matters
  - Constrained vocabulary required!
- Database >>> Files
  - No files to manage
  - Data always in sync
  - Easy to share data
Lessons Learned

- Spelling Matters
  - Constrained vocabulary required!
- Database >>> Files
  - No files to manage
  - Data always in sync
  - Easy to share data
- One tool to rule them all
  - Researchers have different needs
Lessons Learned

- Spelling Matters
  - Constrained vocabulary required!
- Database >>> Files
  - No files to manage
  - Data always in sync
  - Easy to share data
- One tool to rule them all
  - Researchers have different needs
- Power users need power tools
Lessons Learned

- Spelling Matters
  - Constrained vocabulary required!
- Database >>> Files
  - No files to manage
  - Data always in sync
  - Easy to share data
- One tool to rule them all
  - Researchers have different needs
- Power users need power tools
- Digital video is hard
  - Many formats
  - Large file sizes
  - Many, many files to track
Lessons Learned

- Spelling Matters
  - Constrained vocabulary required!

- Database >>> Files
  - No files to manage
  - Data always in sync
  - Easy to share data

- One tool to rule them all
  - Researchers have different needs

- Power users need power tools

- Digital video is hard
  - Many formats
  - Large file sizes
  - Many, many files to track
Software - Design Requirements

Goal: Continue to enable scientists to use MBARI collected video for qualitative and quantitative science.

1. Accommodate future changes to digital video (future-proofing).
2. Efficiently deliver video.
3. Track the locations of videos.
4. Manage videos annotations.
From This ...

- RDBMS
- Cross-platform
- Video Files
- Image Collections
- videotape
To This ...

IDC
- Video asset manager
- Annotation service
- Knowledgebase

IDC

Video asset manager
Annotation service
Knowledgebase

IDC

Image Collections

Sharktopoda

Operating-system specific video player

Java

Cross-platform

No Image

Web protocol

videotape

Web protocol

Video Files

Web protocol
Our Near-term Goal

- Discrete services
- A service
  - provides a web API
  - owns its own data
  - does one thing and does it well
Our Near-term Goal

- Image Collections
- Video Files
- Knowledgebase
- Annotation service
- Operating-system specific video player
- Web protocol
Our Near-term Goal

- Apps use only services needed
- Programming language agnostic

- Web protocol
- Cross-platform

- Video asset manager
- Annotation service
- Knowledgebase

- REST/JSON
- Operating-system specific video player
- Web protocol
- Sharktopoda

- Video Files
- Image Collections
import requests
import json
r = requests.get('http://foo.org/kb/v1/concept/Nanomia')
j = json.loads(r.text)

j['name']
j['alternateNames']
j['rank']
j['media']
j = webread('http://foo.org/kb/v1/concept/Nanomia')

ej.description
nej.alternateNames
nej.rank
nej.media
install.packages('rjson')
library('rjson')
j <- fromJSON(readLines('http://foo.org/kb/v1/concept/Nanomia'))

j$name
j$alternateNames
j$rank
j$media
Using a Service - Perl

```perl
$j==0&&($C++,$C>=$a&&($C=0));$j==2&&($C||($C=$a),$C--);
$j==3&&($R++,$R>=@B&&($R=0));$j==1&&($R||($R=@B),$R--);
KP($) {push@S,shift}KJ() {pop@S||0}KX() {@S[-1,-2]=@S[-2,-1]}KR()
{push@S,$S[-1]}KW($$) {"Z".$_[0]."Z,K{".$_[1]."},"}KG($){($_)=(@a=@_;
y/`/>/;W$a," X;P(J$_ J)"KD($){($a=@_;W$a,Zip Z.$a)KE($){($_)=(@a)
=@_;y/0123/>^<v/;W $_,"N $a"};y/Z\'/;s/K/s$1 /g;my($R,$C,@S);
eval$_;}$_=$x=W '_','N(J?2:0)';y/_02/|31/;$x.=$_ #r^>"J eg"1+T,,,l#
```
Our Near-term Goal

- Video asset manager
- Annotation service
- Knowledgebase

vampire-squid

annoSaurus

IDC

Web protocol

Sharktopoda

Operating-system specific video player

Video Files

videotape

Image Collections

Web protocol
Efficiently Deliver Video to Users - Data Collection

Deployment
8 hour dive is ~1TB

Video Segments
32x 15-min videos
Efficiently Deliver Video to Users - Data Collection

Deployment
8 hour dive is ~1TB

Video Segments
32x 15-min videos

Master
~40GB

Mezzanine
~4GB

Proxy
~400MB
Efficiently Deliver Video to Users - Data Archiving

- Capture resolution
- Minimal compression
- For detailed annotations

Master
~40GB

Mezzanine
~4GB

Proxy
~400MB
Efficiently Deliver Video to Users - Data Archiving

- Capture resolution
- Minimal compression
- For detailed annotations
- Compressed
- For generating proxies

Master ~40GB
Mezzanine ~4GB
Proxy ~400MB
Efficiently Deliver Video to Users - Data Archiving

- Capture resolution
- Minimal compression
- For detailed annotations

- Compressed
- For generating proxies

- Highly compressed at various resolutions
- For outline annotations

Master
~40GB

Mezzanine
~4GB

Proxy
~400MB
Efficiently Deliver Video to Users - Data Delivery

- Physical delivery mechanism
- Prefetch via web
- Streaming via web
- Streaming via web

Master ~40GB
Mezzanine ~4GB
Proxy ~400MB
96 videos / deployment
X
300 deployments /year
------------------------
~30,000 videos / year
For a given deployment and/or moment in time:

- What videos are available?
- Where are they?
- What are their:
  - codecs
  - containers
  - resolution
  - framerate
  - fingerprint (SHA512)
vampire-squid

A simple custom video asset manager.

- [https://github.com/underwatervideo/vampire-squid](https://github.com/underwatervideo/vampire-squid)
- Open source

docker run -p 8080:8080 hononuuli/vampire-squid
Annotation API

Goal: Make it easy for researchers to create and edit annotations from their own applications

- An evolved VARS annotation data model.
- Rich, flexible data model
Annotation API

Goal: Make it easy for researchers to create and edit annotations from their own applications

- An evolved VARS annotation data model.
- Rich, flexible data model

Want to search for any kind of squid eating any kind of Myctophidae between 200 and 400 meters in June 2007?
Index is timestamp

2017-02-21T16:11:09Z

Can have multiple images at one index. e.g.
- left/right stereo image
- raw/color corrected image
Annotations apply across all versions of a video

annosaurus + vampire-squid

Index into Media (Elapsed-time and/or Timecode)

3943735 millis / 01:05:43:22

Master
Mezzanine
Proxy
Annotation - Software Infrastructure

Video and Image annotation service

Annosaurus

- [https://github.com/underwatervideo/annosaurus](https://github.com/underwatervideo/annosaurus)
- Open source

docker run -p 8080:8080 hohonuuli/annosaurus
Status

IDC

Video asset manager

IDC

Annotation service

IDC

Knowledgebase

Sharktopoda
Status

- Video asset manager
- Annotation service
- Knowledgebase

In Production

Web API is Alpha

Alpha - ready for integration

Rewrite required
Existing app to be rewritten to new APIs

Completed

Sharktopoda
Potential Applications - Automation

Color correct, remove marine snow etc.

Register corrected images with annosaurus
Potential Applications - Automation

Store classifier output in annosaurus

- Video asset manager
- Annotation service
- Automated Detector and Classifier
- Knowledgebase

Sharktopoda

Operating-system specific video player
Potential Applications - Extract training sets

Extract training sets of images from annotated video

IDC
- Video asset manager
- Annotation service

Web protocol

Extract Training Set (App)

Sharktopoda
- Operating-system specific video player

Knowledgebase
Potential Applications - Real-time multi-users

Web protocol

IDC

Video asset manager

IDC

Annotation service

IDC

Knowledgebase

Video stream
Ecosystem of Tools

Develop special tools while using existing applications.
Annosaurus Tutorial

This python3 notebook demonstrates the usage of the Annosaurus API which is used for creating and editing video annotations. To get started you will need to start annosaurus. If you have Docker installed you can spin up annosaurus for testing with:

```bash
docker run --name=anno -p 8880:8880 hohonuuli/annosaurus
```

If you do not have Jupyter notebook installed you can launch it with:

```bash
docker run -i -t --name=jupyter \
-v /Path/To/Dir/With/This/Notebook:/opt/notebooks \
-hohonuuli/jupyter /bin/bash \
-c "/opt/conda/bin/jupyter notebook --notebook-dir=/opt/notebooks --ip=* --port=8888 --no-browser"
```

Annosaurus Overview

Annosaurus is a simple REST/JSON service. When you send an HTTP request, it will respond with JSON content (usually). The JSON response is very simple to parse in most modern programming languages.

In the sections below, the various REST calls and there responses are demostrated.

Define URLs

**Important:** In order to run this script you will need to provide your computers inetaddress or computer name as the endpoint.

Normally in an app or script you just define the endpoint and build the other API urls from that.

```python
In [1]: # Define endpoint.
endpoint = "http://10.0.1.251:8880"

annotation_url = "%s/v1/annotations" % (endpoint)
image_url = "%s/v1/images" % (endpoint)
image_reference_url = "%s/v1/imagerferences" % (endpoint)
imagebcd_moment_url = "%s/v1/imagedmoments" % (endpoint)
observation_url = "%s/v1/observations" % (endpoint)
association_url = "%s/v1/associations" % (endpoint)
```
The end