

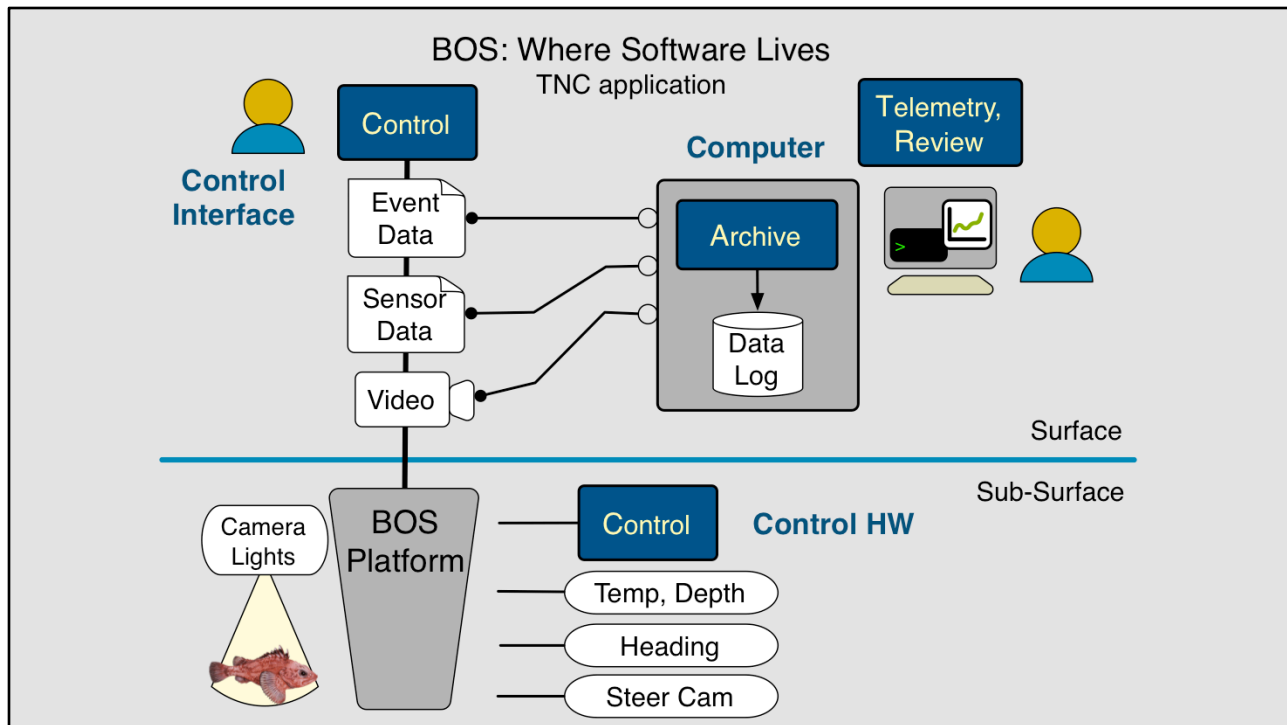
BOS Software Content Outline

- Where the software lives
- Functions, requirements, options : CART
- At the corner of software and hardware
- Implementation approaches
- Selection and recommendations

...

We're in the home stretch...I have just ~10 slides on software
should be pretty quick

Will start by looking at where software may be needed in BOS system...



Here: another version of BOS block diagram, this one showing required software components in dark blue.

Everything below the blue line is underwater, everything above on the surface...probably a boat

Starting at the bottom

- of course: cameras, lights, with some control system for switching power, start/stop recording, etc
- Contextual sensors: TempDepth, along w/ heading sensor and steering cam to facilitate positioning

There is a control interface at the surface that reflects those control operations on the platform.

Steering video and heading data needs to be available during operations, and sensor data along w. control events need to be archived.

And of course, we need to be able to review platform video and data in a post deployment context.

CART: Required Software Features

Control

- Power control
- Start/stop recording

Review

- Data workflows, post-op

Archive

- Record timestamped data
- Contextual sensors:
 - Temperature, depth
 - Heading, [pitch, roll?]
- Control events:
 - Camera/lights power
 - Start/stop recording

Telemetry

- Sensor data
- Video

Those blue boxes make up four software functional requirements for software, which I'll shorten to CART

For control, archive, review and telemetry

I'd point out here that data time-stamping is an important archiving requirement
We need to be able to know when control events occur (lights, camera start/stop)
And align that other contextual observations and video data

Otherwise I think we've covered these, but I'll refer to the CART requirements as we go along.

Optional Features

- **Computation**
 - Parse data in real time
 - Generate new derived data streams
 - e.g. tether turns count
- **Configuration**
 - Sample rates
 - Instrument configuration

In addition to the functional requirements, there are some nice-to-have options
That we should keep in mind – at least not design out

Computation

e.g. pull ship heading from another data stream, combine it with the platform
heading to count tether turns

Configuration (...also maybe network)

There could also be configuration features that could make our lives easier
e.g. automating instrument configuration

Constraints : Design Goals

- Minimize software development for first iteration
- Keep it Simple:
 - Make tech transfer easier
 - Minimal codebase and development environment
 - Minimize system complexity
- Architect for future expansion
 - TBD contextual sensors
 - Real time, distributed/mobile displays
 - Control interface
 - Configuration

There are a few guiding principles and constraints to keep in mind as we narrow down concepts for software and system

Given that schedule and resources are tight, trying not to take on too much SW development would reduce some schedule and technical risk

As software tasks can be hard to estimate

For any software development we might choose to take on,
We should make it as easy as possible to use and extend software, with an eye toward the day-to-day operators of the system like TNC and transfer partners like MARE

We can help by reducing custom code, and if we do write software, choosing well supported languages and tools, and keeping parts count to a minimum.

We also want to be sure to leave room future expansion, and do what we can to overlap with requirements of potential MBARI users

At the Corner of HW & SW

Platform selection : combining HW and SW to cover functional requirements (CART)

HW and SW selection decisions are closely coupled

Hardware choices involve software considerations:

NRE	Lifecycle
HW/SW Cost	Power
Firmware	Sampling Performance
Compilers, Tools	O&M, Tech Transfer
Integration	

In the earlier slides, we looked at some of the hardware being considered for BOS [PLC, OASIS, COTS embedded]

I'll cast those here as as being key parts of a platform:
by which I mean a combination of hw and sw that together satisfy our functional (ie. CART) requirements

Of course HW and SW choices are somewhat symbiotic

Here I just want to draw attention to some of the ways that hardware selection can affect

Software development, and the project as a whole

I won't cover the whole list, but which hardware we choose has a big impact

As it locks down choices that determine

how much software we develop, how long it will take, and how much the total package going cost

And especially since this is something we intend to transfer, it's critical to keep an eye on what

Expertise requirements we'll be passing on to operators and developers down the road.

Implementation Approaches

- Many options to satisfy requirements
- Key decisions: HW/SW partitioning, Build vs Buy

Platform Technology Options			Selection Criteria
	Hardware	Software	
Buy	PC	Wedge	• CART Coverage • Cost • Schedule • Performance • Risk • Tech Transfer • Roadmap
	PLC	LabView	
	COTS Relay	Vendor Utils	
	Stream Logger		
	General Logger		
COTS Embedded			
Build	OASIS	OASIS	
	MMC	Flux	
	BOS new HW	SIAM	
		BOS new SW	

- Lots of ways to satisfy functional requirements
- Can build system with no software development – may not be best performance across the board, but could be done
- The decisions focus on
 - what requirements to address with HW and which with SW
- When it makes sense to build it ourselves (usually for performance reasons) or use COTS (to help with schedule, performance and/or mitigate risk)
- In short, what platform – combo of hw, sw – is best...and what does best mean?
- On the left: some of the many technologies we could use to cover CART
- Short list highlighted in orange
- On the right: some criteria for judging what's best – no surprises there
- The takeaway is that here are some things we considered, and the basis we used to down-select

I think most of these are familiar – OK w PLC? LabView?

COTS relays: industrial control boards functionally like PLC w/ different specs and different software interfaces

- Before moving on, I'll briefly describe OASIS and Flux
 - OASIS: custom mbari hw/sw,
 - instrument controller/data logger, isolated applications like moorings and

Software Data Wedge

"Wedges" between instruments and applications

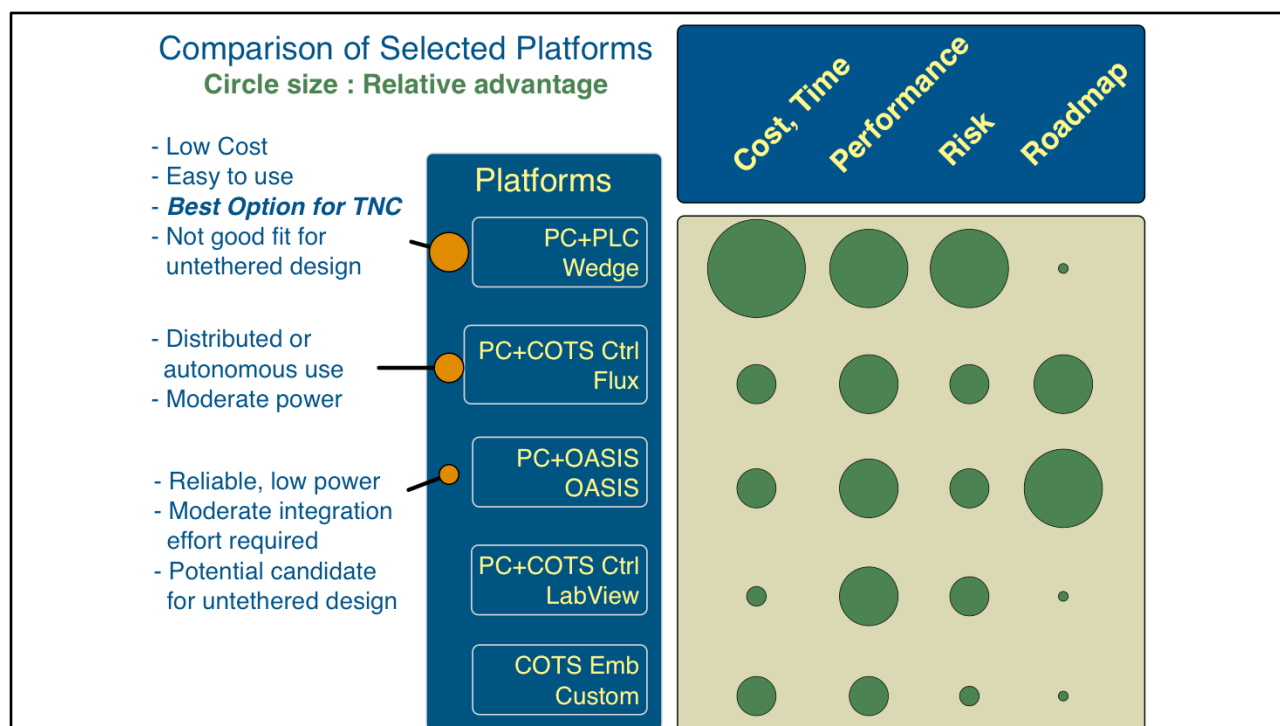
...data sent to files, web server, databases, other applications.

Wedge connects to instruments via serial or TCP...

...wedge acquires data via polling or streams...

...which may be timestamped and/or parsed ...

...even used to create new data via expressions...



What I've done here is to use the short list to make 5 platforms that potentially cover CART requirements

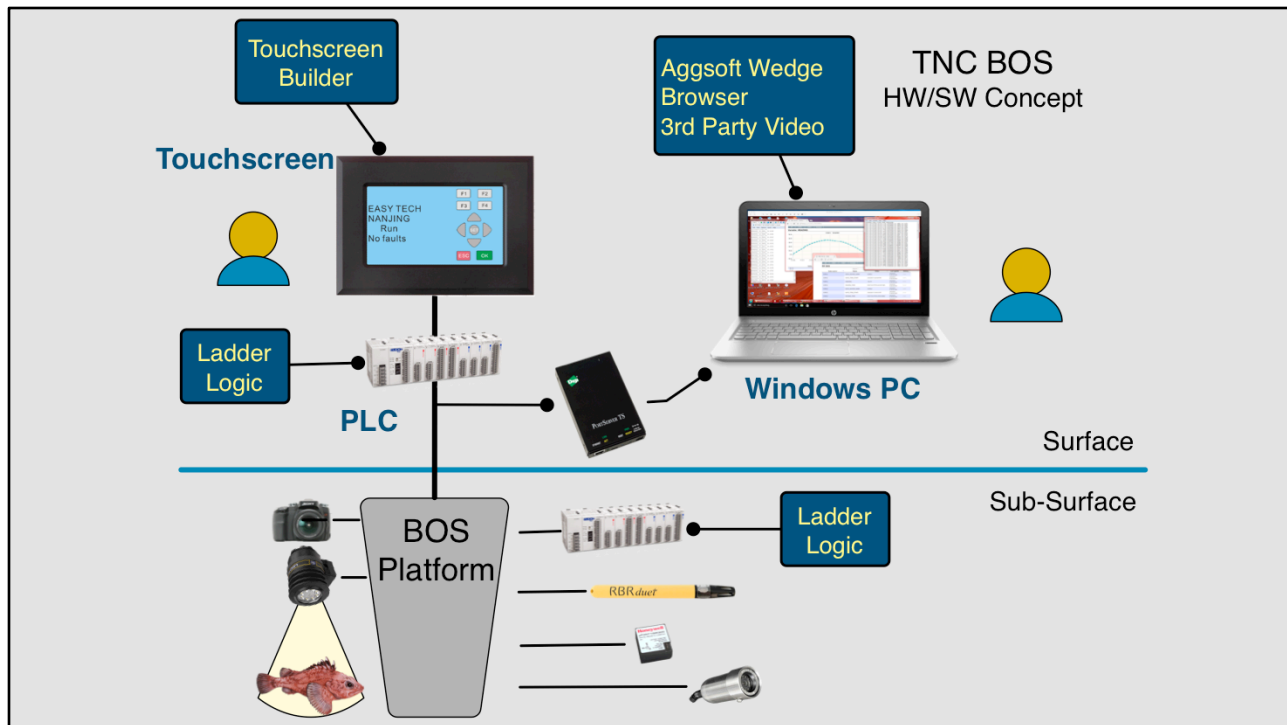
In each platform box: HW on top, SW underneath

Here, a comparison of relative advantage (bigger in cost/time means better in that category...not higher cost, e.g.)

For me, the top pick is to combine a PC with PLCs for control, and use Wedge software to do archiving, review, telem

- PLCs are simple to configure and use, widely available and low cost
- They address the control requirements with almost no software development (more configuration than coding)
- A PLC touchscreen simplifies control interface, implementing distributed control interfaces w/ some configuration, but little/no code
- The wedge is very inexpensive, and fills the Archive and Telemetry requirements out of the box – once again, zero coding required
- There is no custom HW or SW development, and there are a number of vendors offering PLCs and wedge software, so it is somewhat modular

As you can see, that platform doesn't rate well in the roadmap category: mostly, this is due to power, knowing that the MBARI Pis are interested in long



So, to summarize, here is a reprise of the first block diagram, with the specific software concept I'd propose using for TNC implementation of BOS

Again, the blue boxes are software

- PLC and touch screen covering Control
- PC and Wedge cover archiving, telemetry and (to some extent) review
- Video uploads are to be done manually and stored on this or another PC
- Video telemetry and review would be done using third-party applications that run on the PC

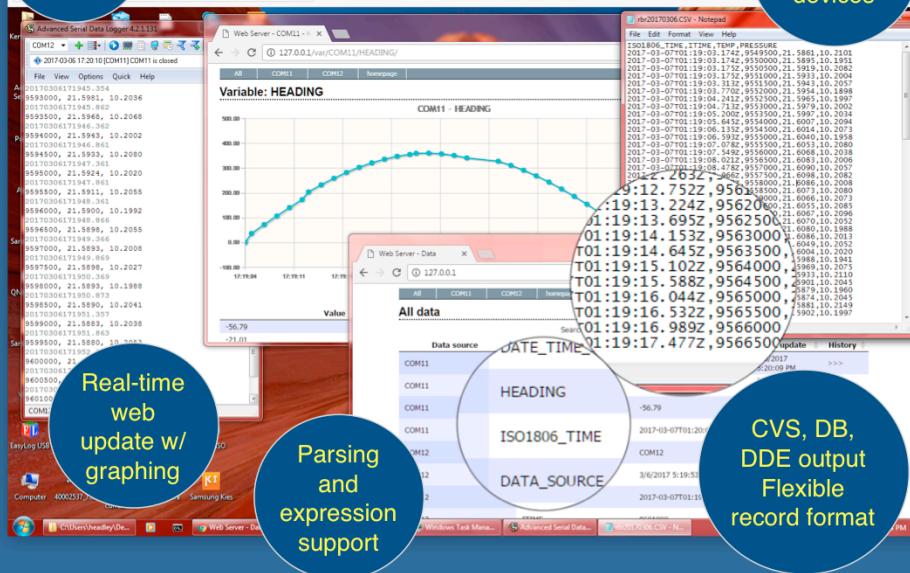
While it's not super exciting as a software project :o),

it's a cost-effective solution that will reduce near term risk, meet requirements and transfer well



Intuitive
user
interface

Polled, Streaming devices



Real-time
web
update w/
graphing

Parsing
and
expression
support

CVS, DB,
DDE output
Flexible
record format