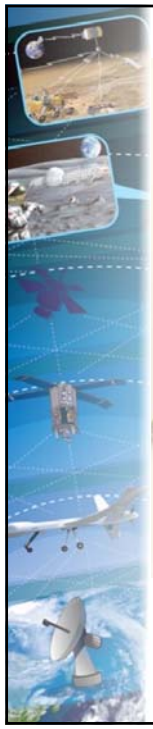
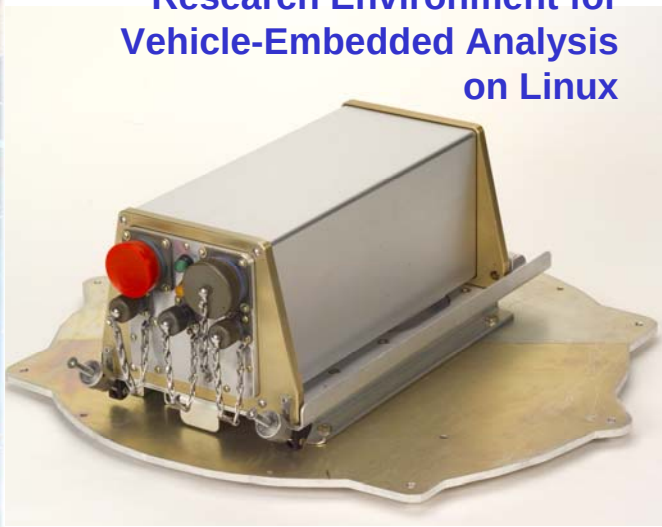
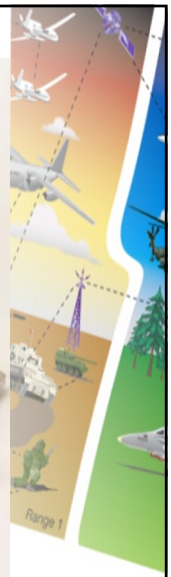


Research Environment for Vehicle-Embedded Analysis on Linux





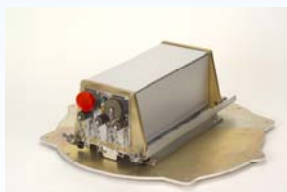
REVEAL: Thinking Inside and Outside the Box for The Future Earth Observation System

Airborne Science Platforms
Data Distribution System Working Group Meeting
NCAR Facility, Jeffco Airport, Boulder, Colorado
14 January 2005

Lawrence C. Freudinger
Advanced Test Technologies Lead
Test Systems Directorate
NASA Dryden Flight Research Center

What is it?

- REVEAL started as a research project into the maturity of Linux & Java for data acquisition, processing, & distribution applications

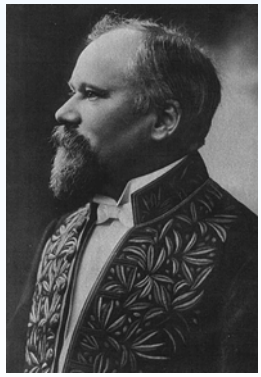


- The REVEAL systems we have today appear to be a significant step towards a sustainable open-standards based data system for Airborne science platforms.

On Making Measurements

"Experiment is the sole source of truth. It alone can teach us something new. It alone can give us certainty."

-Henri Poincaré, 1903



…but at what cost and how long does it take?

La Science et l'Hypothese. Flammarion, Paris, 1927. 1st . edition 1903.
Science and Hypothesis (Dover) 1952 (p. 140)

Outline

- About Us
- Big Picture Discussion
  - Technology Trends
  - Customer Needs
  - REVEAL Motivation
- REVEAL Software Overview
- REVEAL Hardware Overview
- End-to-End Data Distribution
- Concluding Comments

About Us

- DFRC/Advanced Test Technologies Development Group evolves net-centric solutions to test and measurement
  - Broad systems & application engineering experience
  - Vision: global reach intelligent observation systems & capabilities
- Our roots extend across Earth Science & Aeronautics
  - E.g. {CV-990;C-130;KAO;ER-2; DC-8,G-III,Altair, Proteus…}; {F-14, F16XL,F-18,…}; {X-29, X-38,X-31;X-43,…}
  - Skills: Data systems (h/w & s/w) design, prototyping & integration, test and application engineering, data analysis & on-line decision support
- Ten years of relevant award-winning R&D accomplishments
  - Data links: Network-enhanced aircraft communication
    - AeroSAPIENT/Connexion Turning Goals Into Reality 2001
  - Middleware: Collaborative signal processing caching networks
    - Ring Buffered Network Bus (RBNB) HM – NASA Software of the Year 1999; R&D Magazine Top 100 Technologies, 2000; US Patent No. 6,212,568 B1

  Sensor Web application development tools
    - Research Environment for Vehicle-Embedded Analysis on Linux (REVEAL) HM-NASA Software of the Year 2004

In the Spirit of Our Roots

"The research group has an independent responsibility for deciding what is needed to keep research knowledge ahead of application…

… to separate the real from the imagined problems and to make known the overlooked and the unexpected problems…"

Hugh L. Dryden
37th W. Wright Memorial Lecture
Royal Aeronautical Society
28 April 1949

…but not too far ahead of application!

Foresight

"…to enable men and computers to *cooperate* in making decisions and controlling complex situations without inflexible dependence on predetermined programs"

- J. C. R. Licklider, 1960

The lack of situational awareness causes lost opportunity.

→ *Decision-making webs are the reason the Internet exists!!!*

IRE Transactions on Human Factors in Electronics,
volume HFE-1, pages 4–11, March 1960. http://memex.org/licklider.pdf

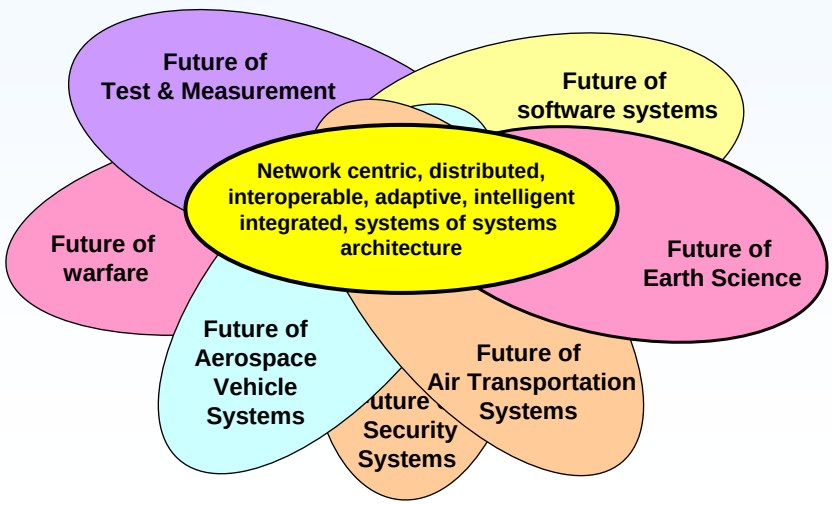
Technology Trends & Customer Needs

When in doubt, predict that the present trend will continue

Trends in Test & Measurement

- Why do people make measurements?
  - Observe; learn; create useful knowledge
  - Situational awareness; decision support; manage machines & processes
- **Big** changes are coming
  - Numbers of embedded networked sensors and computing devices growing fast
  - Starting to see *data acquisition networks* emerge, including aircraft connectivity
  - Intersection of embedded & distributed is a "next big thing" in computing & automation
  - Starting to see the role of network (layered) integration as dominant approach to complex systems-of-systems integration.
- "Sensor Webs" are all about making better decisions faster via convergence of data acquisition and network computing
  - Vision: (Instruments & controllers) + (computing networks) = (increased performance, safety, reliability, productivity)
  - Reality: Sensors don't currently "plug in" to network computing
  - Reality: Technology gaps thwarting productivity
    - REVEAL designed to accelerate gap closure

Big Picture: Industry Trends



Customer Needs: Earth Science

Vision:
Intelligent, Affordable EOS

NASA's Earth Science Enterprise envisions a future that leverages network communication to provide automated and cost effective situational awareness and decision support. The Earth-observing system enabling productive Earth science activities in the 21st century will increasingly employ intelligent constellations of sensors and platforms in a variety of locations. In the future, these observing capabilities will be dynamically linked for both data fusion and for tasking the observing system itself in response to model results or sudden events such as hurricane formation or volcanic eruptions. Geographically distributed research teams and computing systems will be connected in data grids that enable collaborative modeling activities employing the vast quantities of diverse data types provided by the observing system. End users will be able to receive customized information products directly at their desktops in near-real time at no more than the cost of an international telephone call today.

- adapted from ESE Strategy 2003, chapter 5

Decision Support

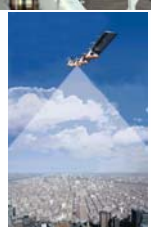
- Vision: Global, in situ, airborne, and space observations all feed into sustainable decision support infrastructures: Every observation, every decisionmaker
  - "… Establish a system of systems that can provide timely data and information for local, national, regional and international policy makers…. Near real-time observations are required to address specific disaster needs (e.g., submarine seismic and volcanic activity and tsunami propagation) and significant extreme events…. Consequently, some participating systems will need to provide real- or near real-time monitoring, early detection, and globally integrated observations. "
- "Enabling all users globally to receive the relevant data in a timely fashion is imperative for maximizing the successful exploitation of the data observations and products. This involves the collection …the transfer …and the dissemination of data and products to users."
- "GEOSS…software components must have open standards-based interfaces."

Ref. doc 203-1, http://earthobservation.org

Maximum value of Earth observation comes from faster, cheaper situational awareness.

Autonomous Vehicle Systems

Vehicle Systems Viewpoint



Airspace Capability
Routine Access To NAS

→ Intelligent/Autonomous Air Traffic Management

Vehicle Capability
Altitude
Endurance
Payload
Range
Navigation

→ Intelligent/Autonomous Vehicle C2

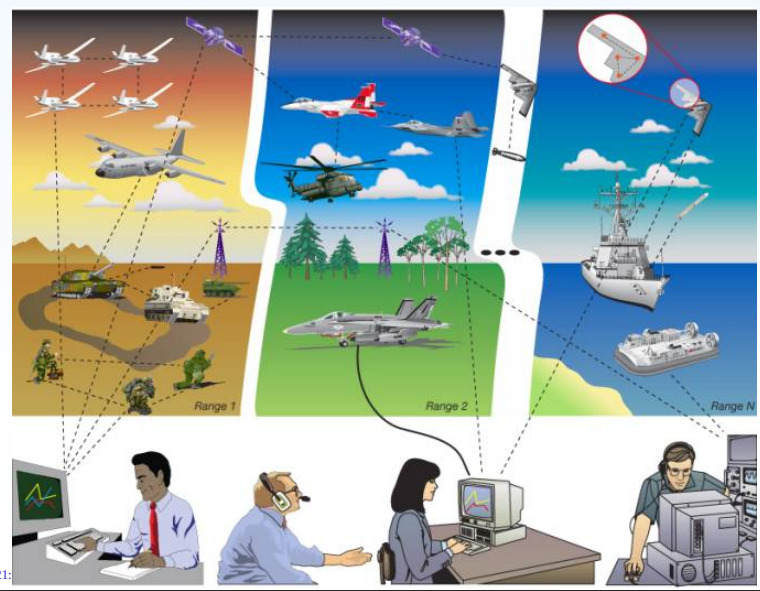
Mission Capability
Affordable…
Multi-aircraft ops
Multi-Instrument
Global reach

→ Intelligent/Autonomous *Observation Systems*
IVHM; Smart sensors,…
Fault tolerant instrument C4ISR

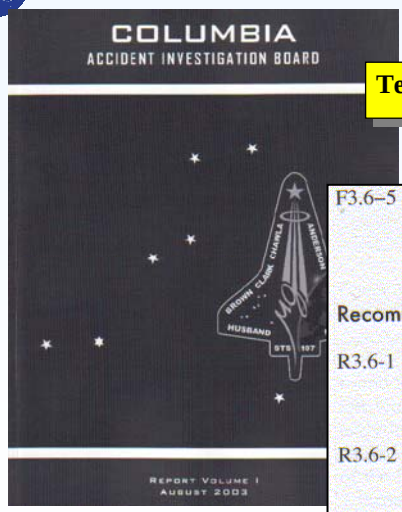
→ Intelligent/Autonomous *Communication Systems*
Fault Tolerant wireless nets…

Needs: Beyond Current T&E Practice

iNET CONOPS: Integrated Network-Enhanced Telemetry



Sensor Webs & Situational Awareness



Telepresence = Situational Awareness

F3.6–5 By the time data indicating problems was telemetered to Mission Control Center, the Orbiter had already suffered damage from which it could not recover.

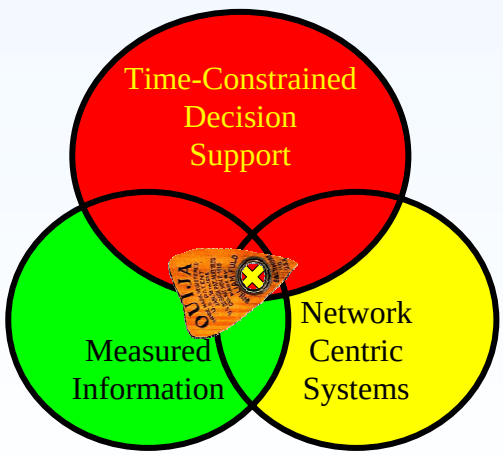
Recommendations:

R3.6-1 The Modular Auxiliary Data System instrumentation and sensor suite on each Orbiter should be maintained and updated to include current sensor and data acquisition technologies.

R3.6-2 The Modular Auxiliary Data System should be redesigned to include engineering performance and vehicle health information, and have the ability to be reconfigured during flight in order to allow certain data to be recorded, telemetered, or both, as needs change.

REVEAL design goals!

Divining the message…



The ES Customer needs realtime and near realtime distributed observation systems based on open standards

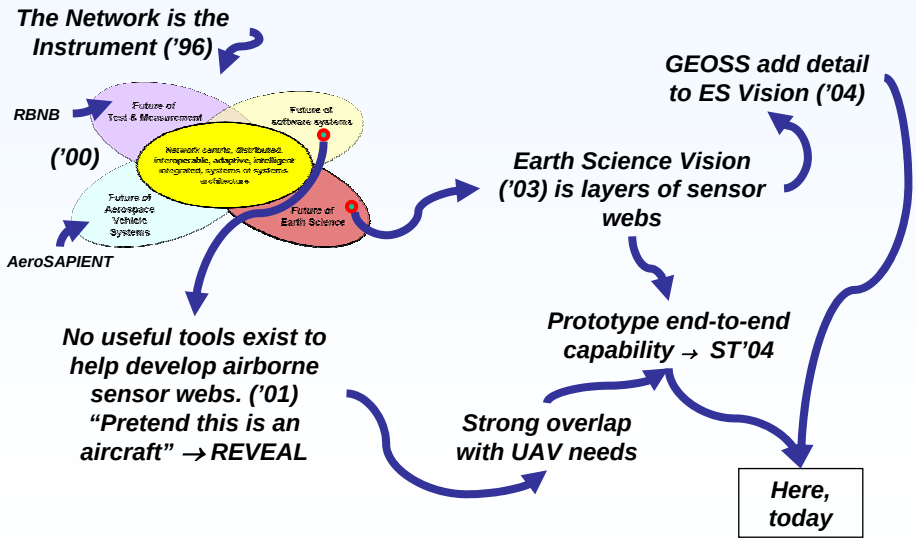
Sensor webs are all about making better decisions faster via convergence of data acquisition and network computing

For Airborne platforms: Data systems standardization efforts need to be leveraging networks to create capabilities expected by customers

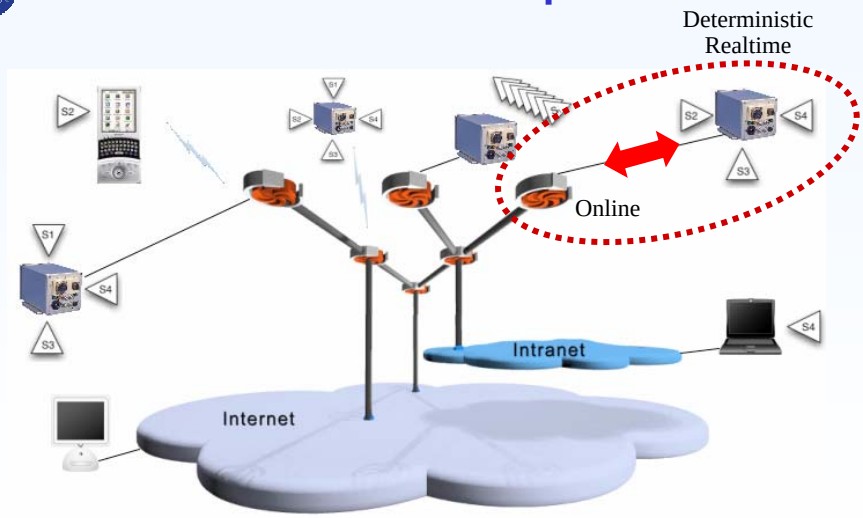
REVEAL Motivation

Where are all the cheap tools?

How we got here (notional timeline)

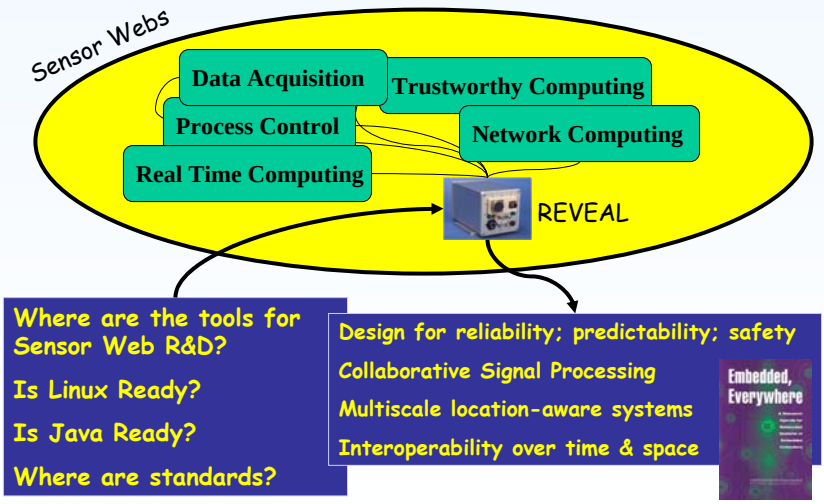


REVEAL Concept



Sensors and computers connected over networks enable increased productivity, reliability, performance, safety, via timely decisions that leverage measured data.

REVEAL is a Sensor Web Tool



Where are the tools for Sensor Web R&D?

Is Linux Ready?

Is Java Ready?

Where are standards?

Design for reliability; predictability; safety

Collaborative Signal Processing

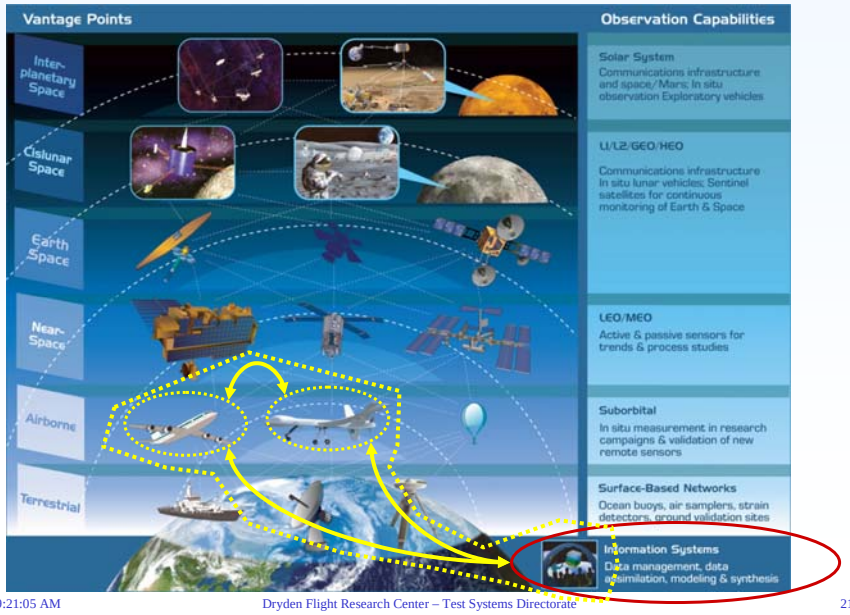
Multiscale location-aware systems

Interoperability over time & space

"Wicked Problems" here - partial solutions and prototypes are essential to developing a deeper understanding (study-build-study…spiral dev.)

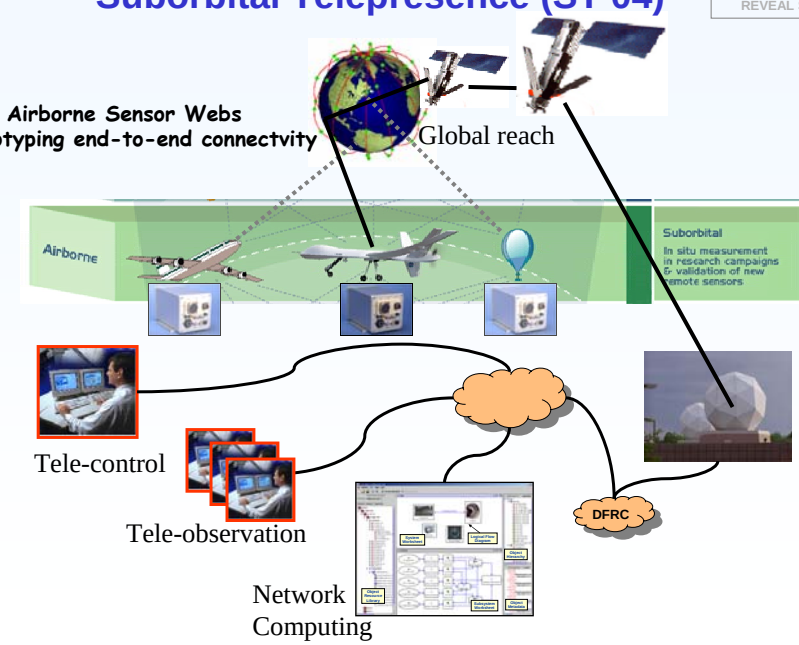
Layers of Sensor Webs

Let's establish repeatable lines of connectivity for suborbital layer



Suborbital Telepresence (ST'04)

Enabling Airborne Sensor Webs
By prototyping end-to-end connectvity



REVEAL Software

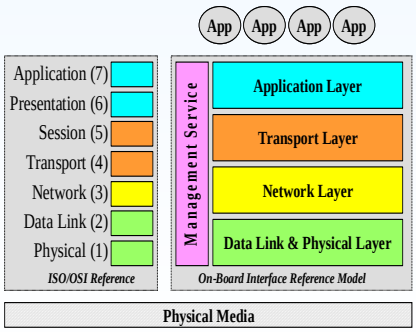
Everyone likes hardware… but the value is in the software

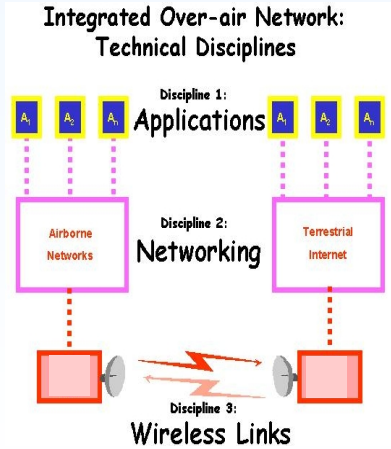
Software Challenges

An airborne platform without the researcher is like a spacecraft – there is nobody there to fix things or tweak knobs or interrogate observations being made by instruments. Need remote interactive access. Both humans and other machines need to be able to communicate with an instrument.

- Reliable & Predictable;
  - Self configuring; self-validating
  - Security features "designed in" from start
  - Communicate health & status
- Adaptable & configurable
  - Create its own interface control document
  - Remotely start/stop/run processes at natural/arbitrary rates
  - Accommodate realtime processes using non-realtime data sources
  - Easy to adapt to changing user needs
- Hardware independent & "Future proof"
  - CPU architecture independent (ARM, X86,etc)
  - Time: needs to be accurate, but not slaved to expensive clock
  - Support multiple simultaneous *de facto* & du *jour* app-layer standards
  - KISS

Layers are Important

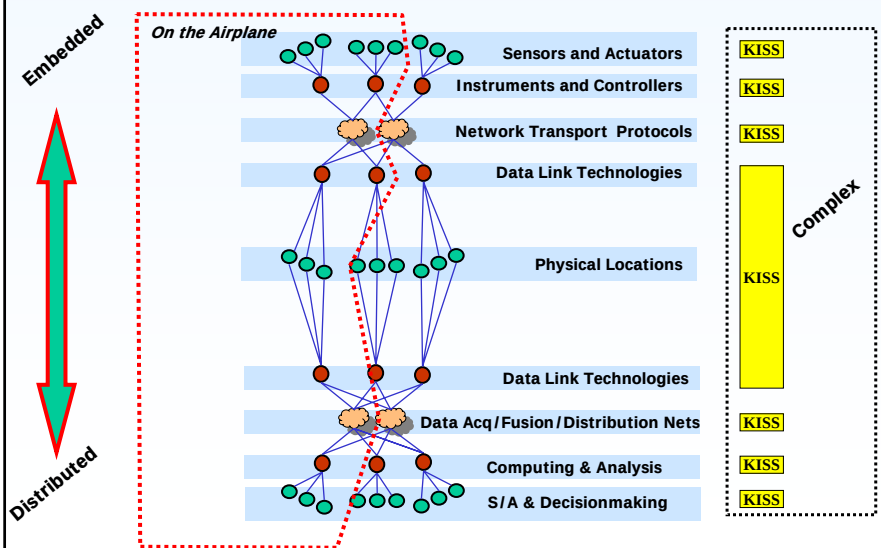




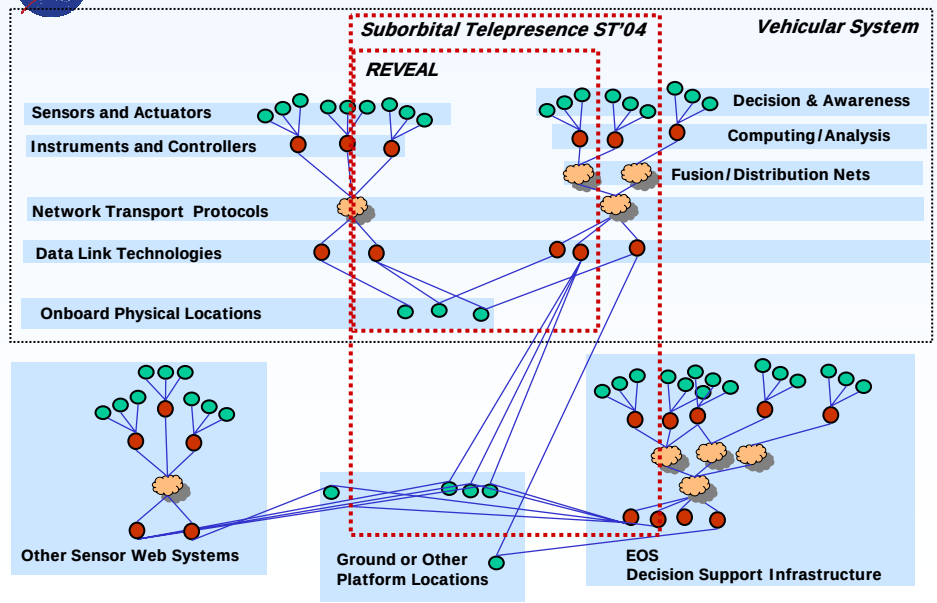
The overall system is separable into layers with problems specific to each layer

Layered Integration

Notional view of sensor-to-decision process automation
→ Simple components increase reliablity & robustness of complex system



Layered System-of-Systems Integration



Notional view

Some Features of REVEAL:

- Self-configuring software via simple XML documents
- Self-verifying via built-in tests and XML resource descriptions
- Self-documenting via XML output documents
- Self-generated metadata via XML output documents
- Acquires/processes data both locally and from remote peers
- Uses caching middleware for expanded interoperability, even to the desktop
- Open architecture and simple API make hardware and software additions much easier
- Security: processes/users can't bother or snoop on each other, by error or by purpose

About XML...

Choosing a schema: many good candidates (SML, IML, SensorML), but realizing that no schema will serve all purposes, we instead adopted a modular approach which lets applications use any desired schema

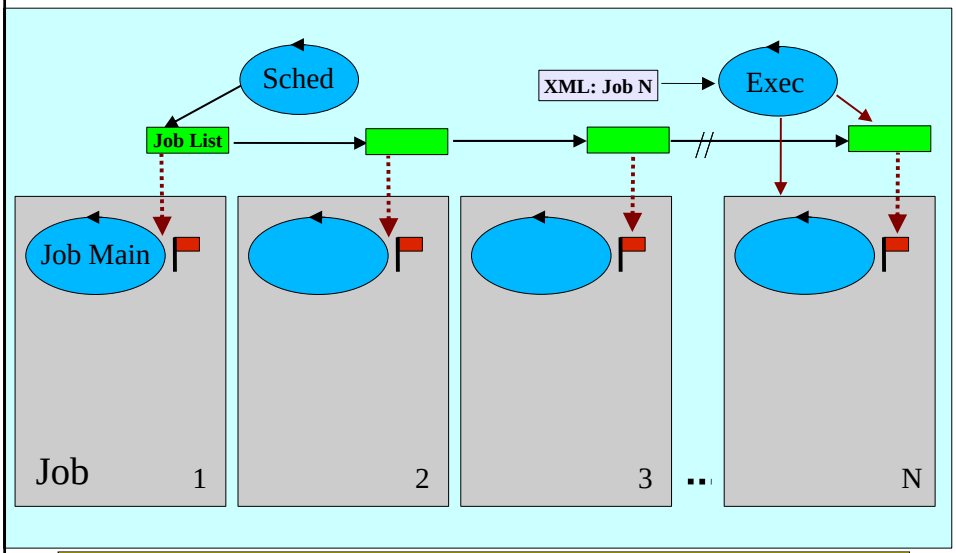
- XML leads to Open-Standards based interoperability
  - We humans can read XML and see what's going on
- XML enables web-based configuration; product generation
  - enable automated discovery and products, eventually
  - XML needed by a lower software layer is encapsulated as a character data URI, which upper layers need not parse or verify
- Software modularity/layers mirrored by XML modularity/layers
  - This greatly simplifies the XML documents
  - Enables web-based document component sharing and admin
- Multiple dissimilar schemas & apps *can peacefully coexist* in REVEAL (eg. SensorML& IEEE1451; Tempest)

Software Architecture Basics

REVEAL spawns and schedules jobs based on job requests and the system resource list. Analogy is **inetd** (the Unix internet super-server, which listens for port requests and spawns services based on a logical unit table)

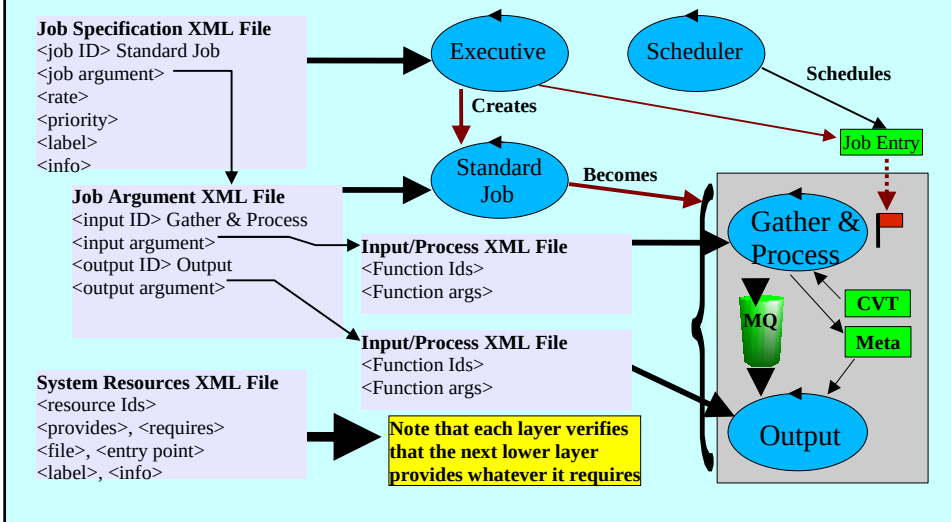
- Resources (software, hardware, data) defined in a simple XML file
- Jobs (user processing) are described by simple XML files
- Jobs are started at boot, or dynamically on request
- Jobs are owned by a userID, and are protected as such
- Jobs run at user-specified rate and priority
- The XML configuration and metadata files can constitute the (automatically verified) system configuration control records
  - Foundation for autonomous documentation generation/management

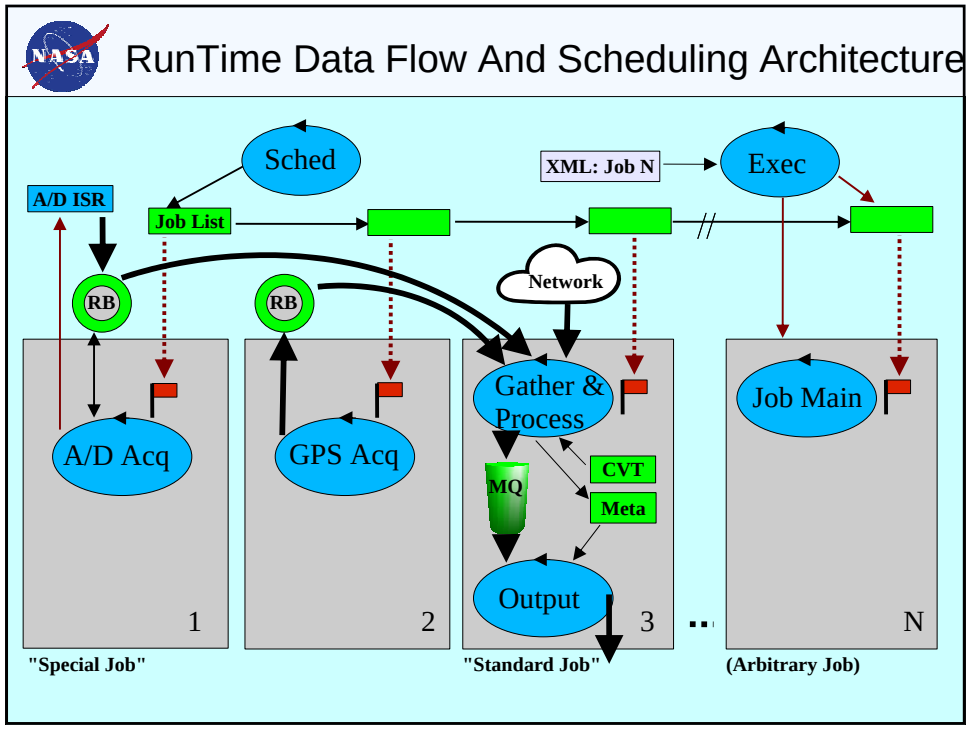
The System Executive Managing Jobs



Realtime: Each job gets it's semaphore ("flag") at that job's rate, +/- ~30µsec

Job Instantiation Via Hierarchical XML Documents





The Working Processes: Jobs

- Jobs are simply Unix programs with standard argc/argv convention:
  - argv[0]: The resource ID of the job, for naming purposes
  - argv[1]: User-supplied argument
  - argv[2]: System-supplied XML argument
- To provide job functionality (ie, have a resource attribute of <provides> = "job task for use in REVEAL"), a job executable will conform to these conventions:
  - Must verify it's arguments and exit if any are bad
  - Must pend on it's semaphore each time thru it's main loop
  - Must catch SIGTERM and shutdown itself and any children
  - If it has child processes, it must catch the SIGCHLD signal

Data Flow Across Determinism Boundary

Determinism is key to real-time systems, but these systems increasingly include non-deterministic interfaces. For example, merging of data from the network with locally-acquired data. **When multiple processes, with asynchronous rates and different priorities, are using multi-channel/multi-sample/multi-latency data objects, opportunities for race conditions and other problems is enormous**. Traditional methods (semaphores, locking) are undesirable because they cause the real-time process to pend, losing a system time slice!

- Simple methods enable data flow across the determinism boundary:
- Data is staged in a shared-memory ring buffer sized to hold about one second, with the producer setting a "current sample" index. Readers only have to check read duration, and no one has to pend.
- For one-to-one data flow (like the standard job) an IPC message queue buffers data between the real-time and non-RT processes

Timestamping and Synchronization

Time is a property of data, but is often used as just another parameter. This simplification can break down in complex distributed systems, when merging data streams from different sources, with variable links, data rates, software priorities etc. all causing problems. Every data object needs a precise acquisition time attribute available, so that any arbitrary downstream merging and analyses can remain valid.

- Timestamps come from Linux system clock: always available and very high resolution
- NTP used to monitor and adjust the system clock, with some special configurations and calibration strategies for performance
- GPS is used as NTP reference; or IRIG-B, GPS on other nodes, network time servers, or general agreement as a last resort
  - Adapts to conditions. Graceful degradation.
- NTP provides time filtering, metrics, monitoring and logging
- Since all nodes run NTP, a REVEAL network will attempt to synchronize clocks, with the reference and each other

But What About Security?

Linux security is typically very good, especially when stripped to just essentials. REVEAL uses the Linux security model, with some extra work, to help prevent hacks and keep jobs from bothering each other.

- Users do not need login/shell access (only an upload mechanism)
- Only approved software can be used (ie system resources)
- System info is private; jobs can't even see the other jobs running
- Job output is also private unless the data is explicitly shared
- Kernel/MMU keeps jobs in their own private memory space
- Resource limits (memory, files, etc) are used wherever possible
- Many traditional exploits are disabled via secure programming
  - Main::shieldsUp()
- Code is small and simple as possible (hard work!) - important for security (& large application base & commercial viability &…)
  - Only 12,434 SLOC including extensive comments (June 04)
- "Defense in depth" helps guard against faulty/malicious code

Linux Performance Issues

Linux's interrupt latencies and task-switching determinism is not as good as most RTOS; however, it is good enough for many projects and it continues to improve.

- Precision microkernels are available, which function as interrupt abstraction layer under the kernel (RTLinux, RTAI). These work well but use a different API and have no memory protection.
- "kernel preemption patch": much-improved preemption latencies
- "real-time scheduler patch": A working RTOS-style scheduler
- "POSIX timers patch": precise high-resolution timers
- On-going kernel improvements and faster hardware keep driving the non-determinism downward over time
- Hardware can drive faster rates for block acquisition (eg A/D card counter/timer)
- Higher scheduler rate can sometimes help, at expense of efficiency

Linux Advantages and Limitations

- Open Source means engineers have access to the source code; this is incredibly helpful, and the GPL is almost never an issue.
  - Free as in speech: You are free to use and change it how you want.
  - Free as in beer: Make it for free or buy value-added content or support
- Reliable, secure, and usually leading edge where it counts
- Huge pool of resources and talent available on the internet

However

- Support from traditional vendors can be critical for some projects
- Reliable, but "DO-178B level A" certifiable is beyond horizon
- Best suited for systems with less-demanding latency requirements (so-called 80% solution)

Linux is no panacea, but it has significant advantages for aerospace and data systems applications, especially in government/education projects.

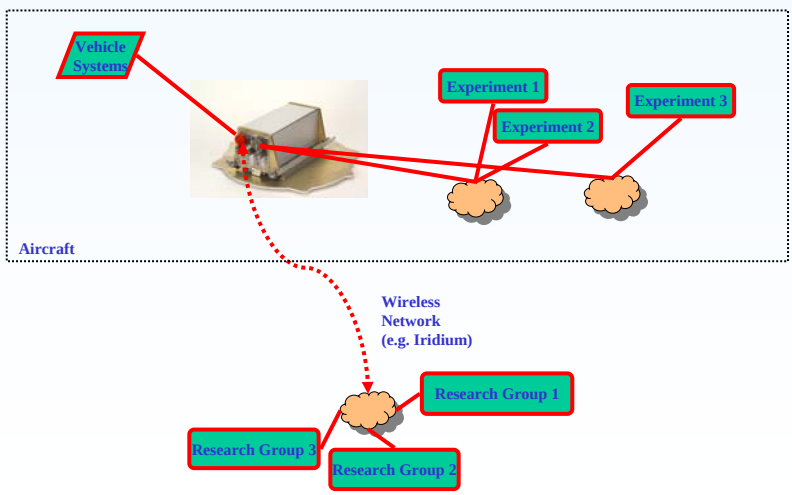
REVEAL Hardware

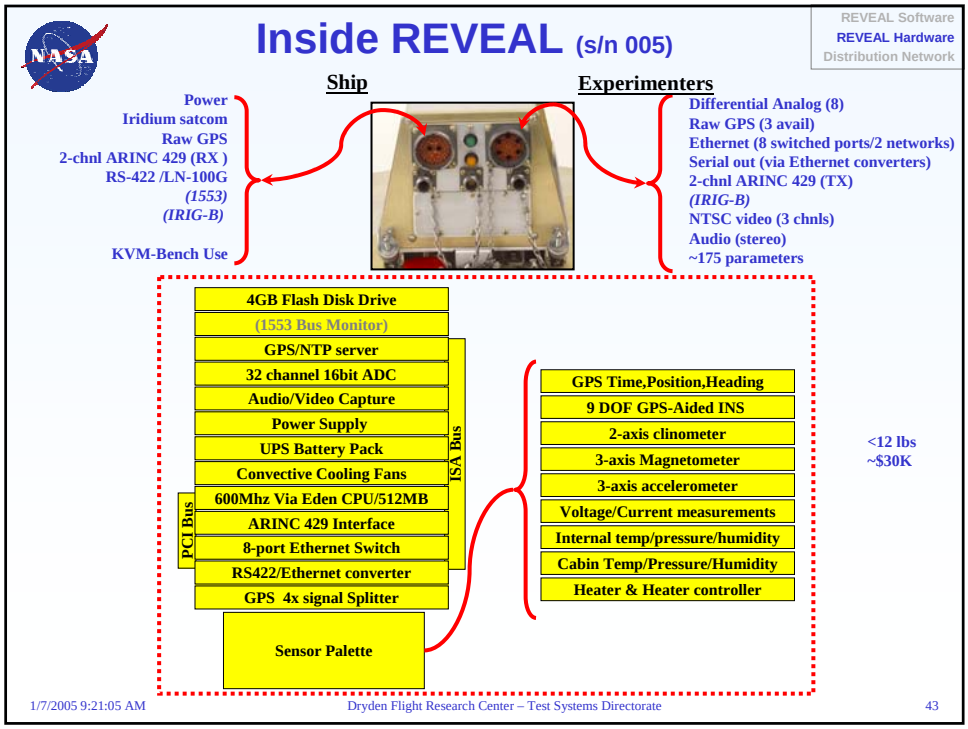
s/n 005-s/n 006

Hardware Focus

- Our ST'04 project targeted UAV applications
  - Embedded sensors ("strap-down")
  - Platform independence
  - Low cost {integration and total cost of ownership}
  - Power/weight/volume/thermal constraints
  - Usability (global reach network C4I)

Nominal Use Scenario



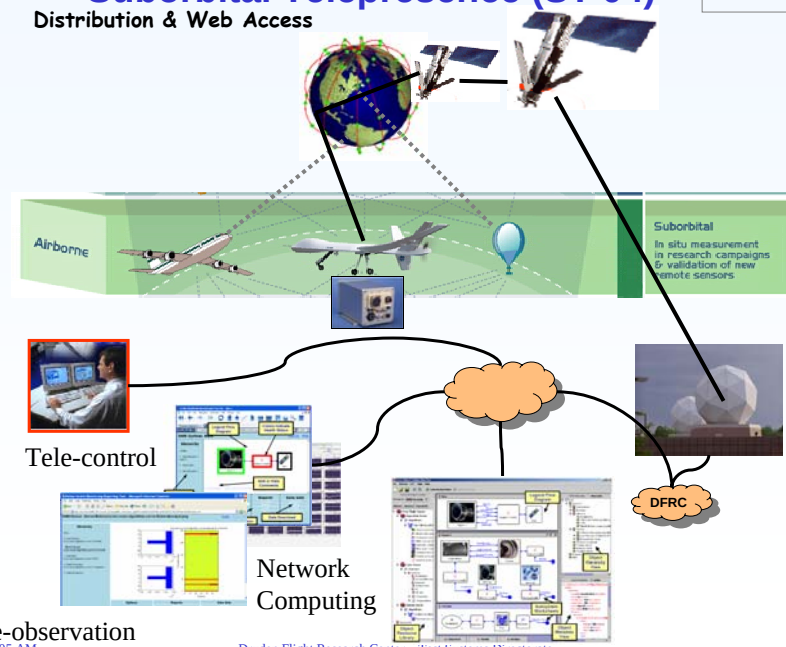


Data Distribution & Network Access

Get the right information to the right place at the right time
or
Remember that decisions are time constrained

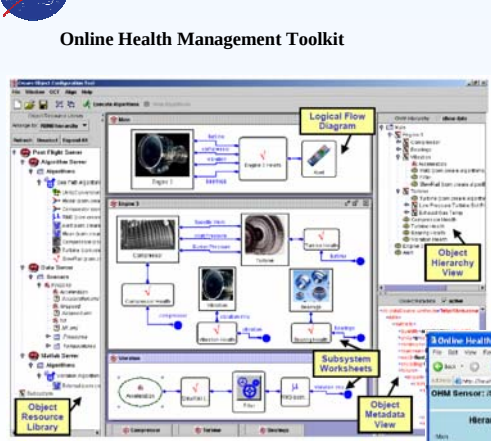
Suborbital Telepresence (ST'04)

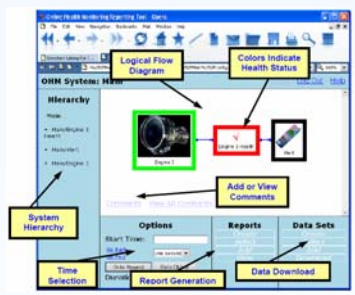
Distribution & Web Access

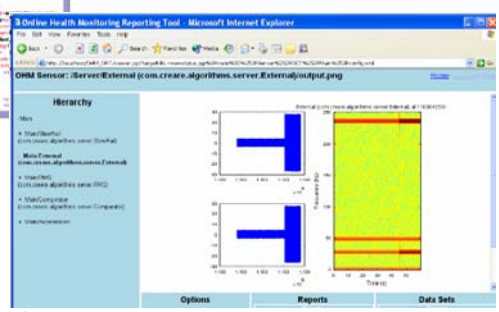


Distribution & Web Access

Online Health Management Toolkit



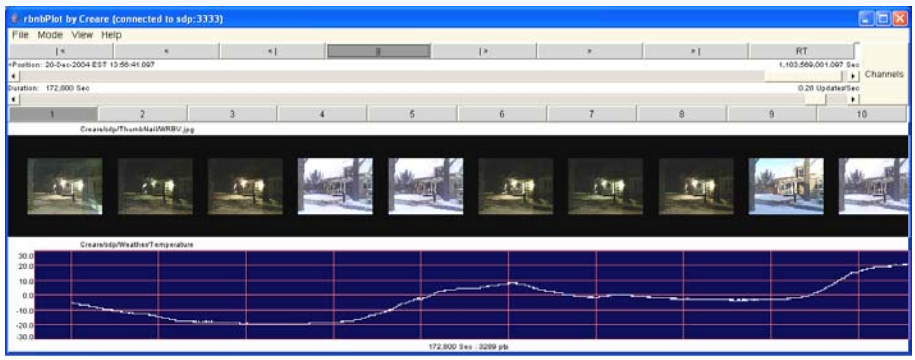




Tools exist to help the remote researcher develop and manage sophisticated automation and monitoring tools of both live and archived REVEAL data. Platform operators can easily enable web-based access to information products and data-on-demand for remote customers.

Looking Forward

- Vehicle-embedded analysis enables intelligent processing of sensor data
  - Bandwidth management
  - Health/status of science measurement operations
  - Signal analysis & decision support



This prototype widget demonstrates dynamic decimation/compression/fusion of most recent 5 GB of live data feeds (from a 30GB cache) for viewing over constrained network links.

Concluding Comments

- REVEAL *software* conceived to fill gaps in low cost tools for general sensor web R&D
  - Technology trends indicate network-enabled systems will dominate
  - Customer needs indicate network enabled systems are the means to end
  - We built a solid foundation for network-enabled open source realtime data acquisition/analysis/distribution system development.
  - NASA owned and open source are significant
  - Designed for adapting to experiment, not other way around
- REVEAL/ST'04 *hardware* is a prototype standard strapdown data system for Airborne Science Platforms
  - ~$30K each in parts (incl. internal sensors); ~175 vehicle & instrument parameters
  - Highly configurable and network-enabled flightworthy system
- REVEAL/ST'04 *system* is a leading example of next-generation functionality for Airborne Science community
  - On path to lowest cost/highest value measurement infrastructure
  - Currently in use - airborne science & others
  - Good starting point