CI Instrument Life Cycle Concept of Operations

Version 2-00
Document Control Number 2115-00001
02/15/2010

Consortium for Ocean Leadership
1201 New York Ave NW, 4th Floor, Washington DC 20005
www.OceanLeadership.org

in Cooperation with

University of California, San Diego
University of Washington
Woods Hole Oceanographic Institution
Oregon State University
Scripps Institution of Oceanography

Document Control Sheet

Version	Date	Description	Originator
1.0	October 28, 2008	Final Version Released for FDR	Chave, Alan
2.0	February 15, 2010	Revised and updated	Chave, Alan

Table of Contents:

Key

Resource

Participant/Role

Actions or Interactions

Authentication/Authorization

Governance/Policy

Identity

1. Manufacture

1.1 Build

After having been designed, tested and manufactured, instrument Alpha still must be programmed. Along with the latest version of the instrument software, the manufacturer, Beta Devices Inc., must create a metadata description of the instrument, and store it in the on-board memory dedicated to that purpose. This process is one of the final steps of manufacture for several reasons: the instrument electronics must be powered to enable memory writes; at Beta Devices the serial number is not known until the instrument is almost fully functional; the configuration of each manufactured device may be customized for particular customers; and any notes about the process of manufacturer, including the serial number and configuration, can then be included in the metadata description.

The same software interfaces that are used to upload (write) the metadata into the instrument will later be used to download (read) the same metadata from it. This software is embedded into the instrument's software package, although the original implementation of the protocol was provided by the OOI instrument protocols group at no cost to Beta Devices (or the other manufacturers who use it). Alpha is a Gamma-type instrument (model Delta-123), so it is an Ethernet device like all of the other instruments of that type; however the instrument protocols group also provides software and hardware that are compatible with RS-232 devices. (It is also possible to connect protocol support units for Ethernet and RS-232 instruments that do not natively support it.)

1.2 Calibration/Test

After the software is installed and Alpha instrument is powered on, it must be calibrated and tested. The calibration steps are performed electronically, and the computer system controlling the Alpha logs the results and the resulting calibration. Both are written into the metadata description in the appropriate sections, and include calibration-specific attributes (date and location of calibration, operator, and expected accuracy of measurements based on calibration results).

Instrument testing is also performed automatically, but in this case a sensor component failure is identified. The sensor (fortunately not one requiring calibration) is replaced manually; this change is recorded manually in the metadata using a software application called Epsilon that can read, update, and write metadata back into the instrument. The Epsilon tool, also provided for free to the manufacturers, can perform numerous types of updates: serial number changes, adding comments to a log record, replacing component descriptions in the XML, and referencing external files or results in the internal metadata record.

2. Operator Commissioning

The Zeta Co. has been implementing an ocean observing system called OOIota for the Ocean Observing Initiative. While the implementation is not complete, many functions are operational while development continues. Zeta's development team is closely coordinating with their operations team that has responsibility for the actual operational activities.

2.1 Acquisition and Logistics

Zeta's development team has procured 12 Alpha's from Beta Devices, Inc. Of course, before the development team sees the units, Zeta's Shipping and Receiving group must receive the incoming instruments. Shipping and Receiving reports a packaging problem in an email message to the development team member who is the contact for the shipment, Eta Garbo -- the packaging on the unit with serial number Alpha 1234 is damaged. (Note that for some providers, one package might contain multiple instruments.) After receiving Eta's promise that she will test the unit and report the results to Accounting so they can pay the invoice, Shipping and Receiving delivers the shipment onward.

Meanwhile, Eta has marked the emails for inclusion in the life cycle log for Alpha 1234. While the life cycle log could be maintained in the on-board metadata for most instruments, the Zeta observatory design maintains it in a separate database, and stores only the most relevant details with the instrument itself. The Zeta team only creates an entry in the database when they receive the instrument, though they dream about integrating information from Procurement and Shipping and Receiving departments into their database someday.

Now that Eta has received the instruments, she powers up all 12 units, plugs them in to her local LAN, and starts the Instrument registration application iRegister. Since these are Ethernet instruments, and follow the standard protocols described earlier, the iRegister tool discovers them on the local network and probes them using the standard discovery protocol for Ethernet instruments. Miraculously, all but one instrument responds, and iRegister goes through its standard process:

1. Compare the device's UUID in its metadata description against the observatory database.
2. Upon finding the instrument is not in the database, records all of its metadata of interest (in this case, all of it).
3. Generates a new observatory section for the instrument's metadata record, including several key items:
    a) observatory identification information (name, location, contacts)
    b) instrument identification information (if it is not the UUID)
    c) the event of the instrument's registration
    d) a private key encrypted using the observatory public key (trusted systems can use the observatory private key to learn the instrument's private key, and thereby represent it in a secure, authenticated way on the network, should that be necessary)
    e) a public key for the instrument, known to all and used by them to verify messages from it
    f) date of the addendum
    g) authenticated signatures showing what observatory and user really did make this entry
4. Update the instrument metadata with the observatory section now included. This observatory section will be updated throughout the instrument life cycle.
5. Produces a printout for each instrument that Eta can use to compare the internal metadata with the identifying information outside it.

Eta has a few further details to take care of. First, she selects the 'Log comment' command from iRegister, selects an instrument by its serial number Alpha1234, and copies and pastes the email exchange with Shipping and Receiving into the record. This same process could also be used if there are any apparent physical flaws with any of the instruments.

Suddenly, she realizes that if iRegister had an email handler, she could just forward the emails to iRegister with the right subject line. She selects the 'Suggest improvements' option of iRegister and describes this excellent suggestion, hitting the Submit button.

The idea is forwarded automatically to the iRegister requests database, and if implemented, will be downloaded automatically with the next iRegister update.

Eta is thankful the instruments all registered automatically, as it is painful to enter metadata by hand. Usually in that case, she will just enter the most basic metadata needed to create a new instrument record, since it gets returned to the manufacturer immediately if its metadata cannot be read.

Finally, Eta confirms that the instrument labels match the key metadata: Manufacturer, Model Number, and Serial Number. Minor discrepancies (e.g., serial number or model on the label doesn't include the 'Alpha' prefix) are fixed by editing the instrument database using iRegister, and appending an image of the label if there is a chance of confusion (e.g., if the label is scratched). Significant discrepancies, like a difference in serial number, mean the instrument gets returned to the manufacturer.

2.2 Configuration, Calibration, and Test

Each of these steps follows a process similar to the initial setup Eta performed, but in this case another Zeta team member, Joe Theta, begins the checkout. The checkout processes are facilitated wherever possible by automation software. Such software makes the performance and documentation of instrument configurations, discovery and documentation of calibrations, and testing and documentation of performance as automated, and closely coupled, as possible.

Without such automation support, two problems quickly arise for large observatories:

- Managing instrument checkout and preparation becomes tremendously time-consuming.
- The metadata for an instrument stops reflecting what actually happened to it.

The former causes steps to be skipped, reliability to decrease, science to be lost, and additional staff to be hired. The latter causes an informal, duplicate metadata system to be created, augmenting the official one, and resulting in the same outcomes as the first problem.

Joe Theta moves the instruments to the ocean testing lab that provides the necessary facility. As each preparation step requires moving the instruments to different locations, the most efficient observatories (like OOIota) provide a way to "check in" the instrument at each step. Usually this is done, or at least facilitated, by the automation and support software since the results of each step must be logged anyway, but sometimes it is necessary for the preparation technician to manually enter information to show where an instrument is. Unfortunately the use of bar codes and electronic ID tags is ruled out by the corrosion and pressure suffered during deployments.

Some of Joe's initial configuration activities are forced by observatory policy that requires that any instrument installed on the observatory be configured or described in certain ways. For example, OOIota assumes every instrument has an owner, a maintainer and a data processor, and these are stored in the metadata record. They will be changed as needed through the instrument life cycle.

3. Deployment

The Zeta operations team needs to deploy 3 of the Alpha instruments as soon as possible on two of its OOIota moorings to take advantage of new science observing opportunities. One mooring will get two instruments, with one backing up the other.

First, the team must discover what Alpha instruments are available; for this they search the observatory instrument database for appropriate ones that are not deployed. They find the iRegister'ed instruments that became visible to all system operators as soon as they were discovered by iRegister, and see that Joe Theta is in the middle of the test, configuration, and calibration sequences.

The operations team communicates with Joe and agrees to help him complete the sequences for three of the instruments. Three members of the operations team each select one insrument for calibration and test, and enter their own identification in the appropriate software so that it is clear who performed what action on which component. This provides key quality control information, as later the results of the different team members in preparing systems can be compared, using the logged information about calibration data, test outcomes, and who performed what steps of the process.

Discussion of Deployment Metadata: The metadata encoded in an instrument before physical deployment depends on the architecture of the particular observatory, and the degree of desire for complete metadata. If deployment-level metadata are installed, mechanisms must exist to ensure that the deployment is 'closed' before the instrument is deployed elsewhere, or the metadata will become corrupted. A sophisticated "plug-and-play" observatory (which is what is described in this section) should typically 'know' instrument location, but sometimes benefits from the 'best estimated deployment location' that can be inserted into the instrument just before deployment. Because of the operational challenges of correlating deployment metadata with deployment activities, this Concept of Operations assumes deployment metadata are not written to the instrument.

Discussion of Device Proxies: in this part of the document, the notion of an Instrument Agent has been introduced. The Instrument Agent is the component that represents the instrument to the networked system. In some instruments, the Instrument Agent may be packaged with the instrument, while in others it will need to be provided as an external attachment or service that bridges the instrument to the network (physically and functionally, or possibly just functionally) while providing necessary additional services. For convenience, the functions of the core instrument that can occur independently of the Instrument Agent (e.g., powering on the instrument may cause it to silently start taking and logging data) are distinguished, but sometimes it is easier to refer to activities of the Instrument Agent as belonging to the instrument. Treating the two as commingled should be serviceable for most of the architectural discussions at the level of the COI, but there is a very specific detailed component architecture that must be developed further before these concepts can be precisely discussed.

3.1 Installation, Network/System Connection and Registration

When the Zeta team physically installs the Alpha instruments on a "plug-and-play" observing platform, they are detected by the platform software. Detection is automatic on mooring Kappa, but on the older mooring Lambda, it must be manually initiated by an operator command. Note there are electrical engineering challenges to these processes, but happily they are beyond the scope of this document. Just as in the instrument acquisition case described above, software on the platform knows how to identify conformant instruments, with different processes used to communicate with them according to their physical connection/networking protocol (e.g., Ethernet vs RS-232 devices).

Following the acquisition sequence, the observing platform will register the instrument it detects and send key metadata back to the observatory information repository. Following OOIota observatory policy, the platform authenticates the instrument's observatory-issued identity before further transactions occur. In the case of observatory OOIota, at this time the platform also evaluates the instrument's authorizations to confirm that its configuration and capabilities are consistent with the observatory observing policies for the platform. Alternately, this may be performed by systems on shore, or may occur (instead or additionally) during command and control operations.

Several observatory actions may also take place once the registration and evaluation process is performed. In particular, at this time the platform and observatory could:

1. Synchronize time on the instrument, the instrument agent, and the platform
2. Identify the best estimate position of the instrument on the platform, based on installation port (or an additional metadata source, e.g., a query to shore)

3. Evaluate nominal instrument performance before providing application-level services
4. Notify interested parties of the instrument's connection and status
5. Update information about available measurements

Most observatories are capable of performing steps 4 and 5 using software provided by the OOI CI team, but OOIota has just remotely installed the surprisingly sophisticated components necessary to perform step 1, time synchronization.

3.2 Command and Control

Upon being powered on, an instrument may simply perform core operations (i.e., operations that are independent of the Instrument Agent representation -- in the previous section, the Instrument Agent connected to the system and uploaded metadata) according to its configuration; may await commands to perform operations or change their configuration, or may combine the two operational modes. *(This document ignores the detail of whether commands to a device may be cached or ignored while a device completes a set of operations.)* In the case where no command or control signals can be issued to the instrument itself, the only activities of interest may be data generation and output. However, note that the Instrument Agent will always have to represent the instrument in a way that accepts and responds to certain commands from the connected system.

When a controllable instrument is registered on a platform, potential users will want to command it. Multiple agents want to be able to mediate commands to the device:

1. The observatory operator
2. The platform controller for the platform on which the device is deployed
3. The device owner
4. Other interested users
5. Other interested parties (e.g., national security authorities)

Each of these may have simultaneous and competing interests in operation of the instrument, so the observatory will have to have policies and procedures for mediating conflicts.

The instrument lead scientist, Zhang Mu, having been given an opportunity to issue commands to the instrument, will want to be confident of the following:

1. The identity of the instrument being commanded
2. The ability to enter all the commands of interest
3. The ability to preclude interruption from other interested users

In this case, Zhang will send a command to the intrument to start taking data every hour on the hour.

When Zhang sends a command to the instrument, it verifies the identity of the issuer and ensures that it is an authorized issuer of commands. (Since the instrument can't be expected to know all potential command issuers, what it really does is verify the identity of the observatory, and accept the assurance of the observatory that this is an acceptable, authorized command.) The observatory or its designee(s) (e.g., the platform) logs all interactions with the instrument for later use in diagnostic analysis of the instrument, observatory, and above all the science data.

Having received Zhang's command, the Alpha instrument acknowledges its receipt, with an intermediate response for a command that takes time to execute, and with the final results of the command when they are available. These responses are linked to the original command by a unique identification, allowing each command's interactive record to be maintained. For example, if multiple users have commanded the instrument to take data every hour, the interactive records allow potential duplication to be spotted and traced to the original sources of the commands.

3.3 Data Generation

An instrument will eventually take data according to its configurations and commands. The instrument or instrument agent may be capable of storing data for a period of time or choosing not to relay it at all, but for the purposes of this document it is assumed that the instrument agent is the equivalent of the device, and hence publishes the data as they become available from it. *(While this assumption may not be valid for any particular instrument, it is valid for the system as a whole, since latency inside the instrument and inside the agent can be combined for architectural design purposes.)*

In a similar vein, instruments may be configured to output data as they are generated, store data internally until they are requested, or do both, keeping an internal copy or version of the data in an internal buffer indefinitely. The first case has been assumed so far. In the instance where data from the instrument must be requested, the instrument agent will render that step invisible to the observatory to the maximum extent possible, consistent with power and other resource constraints. Finally, if an internal version of the data is maintained, they must be downloaded when available and possibly compared to the existing data of record to highlight any errors in the original transmission.

Note that data generation (i.e., the taking of sensor readings) occurs some finite length of time before data publication. While the latency differs from one instrument (and agent) to the next, the maximum latency can be characterized for any given unit. Furthermore, the instrument and agent can only provide the current time to a limited accuracy. Thus, the actual time that any measurement was generated cannot be known precisely, but can only be known to within a time interval based on instrument, agent, and timekeeping network characteristics. The timestamp used to label each data record or other instrument output -- the timestamp that Zhang sees in the displayed data from the instrument -- is the observatory's best estimate of the average time of the sensor measurement(s), based on all these parameters.

As the lead for the instrument, Zhang gets to see the data that result from his command. However, the wider publication of instrument data must be mediated by the rest of the observing system components that together determine when and how the data records are released to interested parties. For example, on some platforms data records will be kept on the platform until a cost-effective means of off-loading them is available, whether that is a passing platform with suitable wireless functions and data download applications, or a ship recovering the entire platform.

Zhang would like to see all of the data from the new Alpha instruments. Any potential viewer of the data must meet appropriate authorization criteria; the list of mediated users includes any entity that is a participant in the system with the role of data user or data handler. As the owner of the Alpha instruments has not stipulated any review period for the output, all data are immediately available to any member of the general public who is registered in the OOIota observatory portal. As a registered member, Zhang has no trouble accessing the Alpha instrument data streams and the corresponding archived data sets.

3.4 Failure Detection, Diagnosis and Repair

Detection of instrument failures can occur through multiple channels. The participant or role that detects and reports each type of failure is noted in parentheses.

1. Communication with the instrument can fail catastrophically (Platform)
2. An instrument owner or lead can notice problems during interaction with an instrument (Owner/Lead)
3. Automatic quality control software can detect data anomalies (QC Software)
4. Device-specific quality control/processing software can detect problems. (Instrument-Specific QC/Processing Software)
5. Users of the data from the instrument can notice problems. (Data Users)

In each case, a communication process exists to log the problem in a way that references the affected instrument.

In fact, Zhang notices that the Nu sensor on Alpha instrument #2623 (the primary Alpha on the Kappa mooring) is not reporting any useful data. Zhang notifies the observatory operations team using the handy "report a problem" button in the portal page where he is viewing data. This button captures the instrument number, the URL accessed, and information about the user, and puts that information into a problem reporting record. Zhang adds to that record the problem he is seeing in the data on the page, and presses the Submit button. The report is sent directly to the observatory operations team, and is automatically assigned to Eta Garbo by that team's problem reporting system.

Faced with few diagnostic options for this particular instrument, Eta is limited to checking the log records for possible physical problems or configuration problems. Because the instrument went through a thorough test process, Eta can tell the Nu sensor was working before the instrument was deployed. Since the likely problem is a physical one and there is no immediate possibility of repair, Eta confirms that the secondary (backup) unit is working correctly, then swaps the primary and secondary Alpha systems on the Kappa mooring. Now instrument #2623 is merely providing auxiliary verification data, and Alpha instrument #2621 provides the primary stream of Alpha data for that mooring.

(In fact Eta makes the swap effective as of the installation date, so anyone asking for 'primary' Alpha data from the earlier time period will get the entire data set from #2621, rather than #2623. This has some consequences for data consistency and integrity for similar requests issued at different times, before Eta's command and after it. Unfortunately, those consequences are too complex to address in this scenario.)

Although the mooring is notified of the change, it does not carry out any processing using the data from either Alpha, so no immediate changes result in mooring calculations. But the post-processing software that is responsible for maintaining the Kappa mooring time series will notice and respond to the change because requests for 'primary Nu data' from the mooring will receive data from a different instrument. The post-processing software must track these data source changes so that the metadata record for the post-processed results will accurately reflect what post-processed data sets came from which instruments. When the post-processing is rerun as a batch process at the end of the month, the earlier results using instrument #2623 will be replaced by results using the Nu sensor (and other sensors) in instrument #2621. The previous set of results will be maintained to enable accurate record-keeping in case anyone used the previous data sets in their own analyses.

4. Recovery

As it turns out, a problem with the Alphas necessitate their replacement with the Omegas, the next generation instrument. Thus it becomes necessary to remove the Alpha instruments from the platforms. (Because the Omegas have Nu sensors as well, and the Omega instrument is well described by metadata, the systems that do post-processing on the Nu sensor data will be able to perform the same functions as they did before, with no software modifications required even though the instrument's data stream format is significantly different. The long-term time series is likewise maintained without additional intervention, with the instrument change noted in the metadata to help explain any apparent differences in the data that appear at that time.)

4.1 Turn-Off and Removal

Before the instruments are powered off and physically removed, several steps occur in the observatory. The observatory operators begin to disable the instruments, which actually incorporates several important steps: notifying the instrument users, including any post-processing or quality control software, of the change and its rationale; commanding the instruments to stop collecting and publishing data; notifying the platform that the instruments are being removed; and putting the instrument into a stable and safe state for turning off its power. At that point, operators can turn off power to the instrument and send out a marine operations team to physically remove the units. Once removed, the new location for the instruments is logged, so they can be found if needed later.

4.2 Decommissioning

Once the instrument is physically removed, operators can change its status to match circumstances. In this case, because the instruments are not considered suitable for further use by the observatory, the status for all of them are updated to "Unavailable", with a suitable explanatory comment. This will prevent users from trying to obtain those instruments and install them elsewhere, as only observatory operators (or an explicit search for that instrument type or ID) will be able to find the instruments, and their status will be immediately apparent.

4.3 Disposal

Eventually the OOIota Observatory decides to get rid of those lousy Alpha instruments (whose idea were they, anyway?). The instrument status for all them are updated to "Destroyed", and a large sculpture is created by a local artist using their salvaged remains.