MOOS Behaviors Required by SSDS

Introduction

This document describes behaviors of the MOOS system that must be provided for SSDS to meet known MOOS requirements. The document was produced in response to a request from the MOOS Project Manager and ISI Project Manager. The intent is to represent MOOS and ISI software requirements in an implementation-neutral way, so as to provide maximum flexibility for ISI and other MOOS components to meet those requirements.

Note that this list does not capture all the MOOS software requirements, as many of those do not directly impact SSDS. It also only addresses those SSDS requirements that rely on MOOS functionality to perform correctly.

As MOOS system responsibilities continue to be partitioned and its software architecture developed, this list may need to be modified.

Data Sources

The sources from which each requirement is derived are presented in the Origins column, according to the following key:

MFR:	MOOS System Functional Requirements, 3/22/2000 <http://mww.mbari.org/moos/Systems_Engineering/FR_rev.pdf>
MDR:	MOOS System Design Requirements, 11/29/2000 <http://mww.mbari.org/moos/Systems_Engineering/moos_system_mrc.htm>
MDRR	MOOS Design Requirement Recommendations <http://mww.mbari.org/moos/Systems_Engineering/moos_design_requirements_recomme.htm>
MDMW:	MOOS Data Management Workshop Report, 6/21/2000 <http://mww.mbari.org/moos/Data_Management/MOOSDMReport.PDF>
MWR:	MOOS Workshop Report (The Future of MBARI Ocean Observing Systems), 3/10/2000 <http://mww.mbari.org/moos/MOOS_meeting_1999/MOOSfinal.pdf>
UC-ISI+	Addition ISI Use Cases—Draft, 1/24/2001 <http://mww.mbari.org/moos/ISI/Use Cases/ISI_UseCases1.pdf>
UC-CTD	CTD-ISI Use Cases, Draft 1 Revision 1, 6/22/2001 < http://mww.mbari.org/moos/ISI/Use Cases/ISI_CTD_USECASE_1.htm>
UC-ADCP	ADCP-ISI Use Cases, Revision 3, 6/22/2001 < http://mww.mbari.org/moos/ISI/Use%20Cases/ISI_ADCP_USECASE_1.htm>

DMRR:	SSDS Data Management Requirements Review Meetings, 9/18/2001 and 10/2/2001
SRS-DS	Software Requirements Specification for the data system (SSDS)
Meta	Metadata meetings (minutes in *metadata_mtgs_020321-27.doc* dated 4/4/02); '#n' gives row number in table "List of SSDS Functions Facilitated by Metadata"
NDR	NEPTUNE Data Management and Archiving System Design Requirements (DRAFT), 1/21/2002

Numbers after the document identifier are typically section numbers. These may be followed by '-' and any subsection, bullet, or item numbers. (MDMW item numbers in section 4 renumber the second set of bullets from 1-4 to 9-12.) The order in which the requirements documents are listed above is their precedence order. If I found a source requirement in my top-down search through those documents, I did not attempt to find redundant relevant requirements in subsequent documents, although in most cases more references can be found.

Requirements that are engineering derivations are preceded by **[Derived]**, while those which follow directly from Level A requirements (i.e., clearly traceable to stated requirements) don't have any marker.

SSDS-required behaviors that seem likely to be acknowledged/accepted are in black. Those expectations that seem likely to be debated are in blue.

Terms that are defined in the MOOS SSDS requirements (section 1.4) are presented in *italics* when first used.

Required Behaviors Table

Required Behaviors	Origins	Comments
[Derived] A standard, flexible format must be used to describe all *data records* (whether science or operational data, events, or *metadata*) received by SSDS.	**MFR**#8,9	Required to support scaling MOOS to many sensors and moorings (and important to minimize programming)
All received data records and data stores must be accompanied by: 1. unique *data source ID* (including versions, for SW) 2. time of data ingest into ISI/MOOS 3. reference to (i.e., a way to access) the unique *record content description metadata*	**MDMW#4-4**	Provision of this information must be automatic and guaranteed if data management is to have any hope of scaling up.

Process must be supplied that guarantees unique data source IDs identify different hardware components. (The data source ID for hardware is referring to the same thing as the ISI Device ID, by definition and assumption.)	**MDMW#4-4**	Needs to be embedded in overall MOOS architecture, for same reasons as above (must be automatic and guaranteed if data management is to have any hope of scaling up).
[Derived] Process must be supplied that guarantees unique data source IDs identify different software components and versions (software changes on ISI must change data source ID)	**MDMW#1-7, 4-10, 4-11**	Needs to be embedded in ISI's SW architecture somehow, or else anyone can just put another SW component out there with an errant software ID. (Is a different version of software a different ID? Would be nice to have ID and version both.)
[Derived] All received data records and data stores must be accompanied by 1. *data record type* (for multiple *data sets* from one *data source*) as indicated by originator (data source or its proxy, the driver)	**MDMW#4-4**	Scientist (instrument) needs to be able to specify the relationships of its data records, e.g.,, define that a new set has started (typically SSDS will define buckets (where we put data) based on metadata, this is an additional rule)

For every type of received data set, provide content description metadata containing
1. data contact person
2. comments about the data set
3. basic format of data item(s) (the *use metadata*: type, size, units, unit reference, and descriptive name for each described item)
4. characterization of the content description itself
    a. *record content description metadata ID*
    b. record content description metadata version
    c. a timestamp indicating creation time of this version of the description
5. if applicable, identification of any predecessor data sets from which this data set was created
	MDMW#4-4	The characterization information is needed so that updated descriptions can be recognized and utilized effectively (knowing which is most recent, and to which record types it applies).
		
		Note that a single record of a given origin and type defines a data set of that type, so this requirement is also associated with every type of received data record.
		
		Who will define the form of this description (SSDS?), and how will it get into the MOOS system so that it eventually comes back to SSDS (puck via instrument developer?)?

[Derived] Users/operators must be able to correct errors in data's	**MDMW#4-4**	If a bug is incorporated into the "on-board"

metadata/accompanying information that are associated with data records or data sets		(instrument-associated) description, the architecture must at least support its suppression (correction would be better)
Provide necessary information to automatically determine where data was collected	**MDMW#1-8, 4-4**	For some data, this changes with every record; for other data, the entire data set is at the same location.
[Derived] Provide notifications of observatory physical configuration changes at and above instrument level (installation/removal of instruments, nodes, and other data sources)	**MFR#8, MDMW#1-7**	Need to know configuration of systems (location of instruments relative to mooring(s)), both current and historical, to answer the question: "Give me all the data of type X from (location Z at) mooring Y"
[Derived] Provide notifications of physical changes to configuration of instruments (installation/removal/reconfiguration of sensors)	**MFR#8** **MDMW#4-11** **UC-ISI+#4** **UC-CTD#3** Meta#17	Can instrument sensors be changed in situ? This is a common occurrence presently (see UC-CTD introduction), and is documented in numerous sources over the past several years.
[Derived] Provide notifications of software configuration changes to instruments	**MDRR#8-a** **UC-ADCP#0-8** **UC-ADCP#9** Meta#19	When ADCP is commanded its configuration changes. When instrument loses its mind, previous software configuration changes are lost. In either case, changed configuration must be detectable in the context of the data stream.
Raw data formats (i.e., formats produced by instrument) must be recoverable from information in delivered data records	**MFR#11**	SSDS must be able to provide all data in its raw format
Provide notification of asynchronous events	**MFR#5-d, 10-2a, 10-2b** **MDMW#1-2**	MOOS will include science event detection, SSDS must be able to save and present all raw science data (which a newly detected event is)
Be capable of providing a data set for an instrument, at worst case through a non-ISI communication channel, such as through physical recovery	**MFR#10.1**	Saving all raw data is only meaningful if it can then be retrieved/accessed. (How is this done mechanically? What is on-shore interface to SSDS?)
[Derived] Provide information within the data stream that can be used to determine when data is missing	**MDMW#4-1**	Must be able to save all raw data (interpreted here as "all transmitted raw data," since some systems won't transmit all their data to shore) –

		the SSDS must know when we haven't seen all the data. (This behavior is also implied by the fundamental nature of a "connection.")
Provide deterministic access to data that was identified as missing.	**MDMW#4-1**	The SSDS must request any missing data in order to be able to save all the raw data. Note that this gets tricky if the data subscription and retrieval models are not deterministic (i.e., if the models do not allow unambiguous specification of the data known to be missing, a particular request might not return all needed data without that being detected).
Either guarantee that data is in order, OR provide information to detect out of sequence data and resequence it.	**MFR#11** **MDMW#4-1**	(Note this does not include issues of missing data, only out of order data.) SSDS must be able to provide data in its raw format, which means original byte stream order, so it must have a way to ensure its data is in the correct order.
[Derived] SSDS must be able to subscribe to a stream of information from MOOS data sources (both by data source, and by all sources) in such a way that it can be sure no records are missed from the time the data source began transmitting data.	**MFR#9,** **MDMW#4-1**	Polling won't scale to the multiple platforms and instruments that MOOS will encompass, so a subscription ("notify me when new data arrives") mechanism is required. (Note data will be arriving from more than just moorings.) Also, the mechanism must be capable of picking up the very first data record and supplying all subsequent records, or, SSDS would often miss critical configuration information (because it gets sent to portal before SSDS has subscribed to it).
[Derived] SSDS must be able to get list of current data sources, and request notification of new data sources	**MDMW#1-7,** **4-1**	This is generally important to know if a data source of interest is on-line, but it is especially important to SSDS if no general means to subscribe to all data sources is possible.
[Derived] MOOS must allow for integrated, single-pass data	**MFR#9**	Requiring multiple passes, or excessive

entry of system configuration and log info by operators	**MWR#5-1e MDMW#1-7, 4-6**	interaction in a single pass, will be unreliable, will discourage users from adopting MOOS processes, and prevent the MOOS and SSDS designs from scaling up
[Derived] Provide a protocol for requesting historical data from devices which can apply equally well to historical data request from SSDS	—	SSDS should be able to implement the portal's "give me the data" request format, so that requests can go to SSDS instead (saving unnecessary accesses across low-bandwidth media).
[Derived] Defined mechanism for cleanly handling the loss of multiple data sources (e.g., given node or platform failure, some software should report or generate corresponding events at the system level, so that all processes using data from those components handle the failure gracefully)	**MFR#8**	If it's just a communication failure, this should not produce 'disconnect' messages for the devices. But if a node has died and/or is physically removed, system must recognize that logical device connections are gone.
[Derived] Data timestamps must be accurate to within 1 ms (absolute accuracy).	**MDR#9.2.2**	This is a fundamental MOOS requirement, but the SSDS depends on it for optimal integration of disparate data sets. It also forms the basis of operational and other metadata the SSDS uses to answer science queries. (Data that arrives with a significantly wrong timestamp may be grouped with the wrong data sets.) Note that if timestamps are sufficiently precise, they may serve to order data records. (Although, since 2 records could be submitted within the same millisecond for some fast data submitters, this may mean sub-millisecond timestamping)