SCIENTIFIC DATA ARCHIVING: THE STATE OF THE ART IN INFORMATION, DATA, AND METADATA MANAGEMENT

KALPANA SHANKAR, Ph.D.
CENTER FOR EMBEDDED NETWORKED SENSING (CENS)
UNIVERSITY OF CALIFORNIA, LOS ANGELES

August 6, 2003

Table of Contents

Scope ... 3
Introduction ... 3
Methodology ... 3
The Case for an Integrated Information and Data Management Plan ... 4
Challenges ... 4
Requirements/Components ... 5
General Data Management and Discovery ... 5
Metadata ... 7
Sensors ... 10
Environmental Science/Ecology/Habitat Monitoring ... 11
Seismology ... 13
Oceanography ... 14
Atmospheric Sciences ... 15
Toxic Hydrology (Contaminant Transport) ... 16
Geographical Information Systems (GIS) ... 16
Education ... 17
Conclusions and Recommendations ... 19
Appendix 1: CENS James Reserve Database Data Dictionary ... 21
Appendix 2: CENS James Reserve database – Sensor ML crosswalk ... 27
Appendix 3: CENS James Reserve database – Ecological Metadata Language (EML) crosswalk ... 34

Scope

This paper will focus on introducing current thinking on scientific data management issues and primarily on relevant standards in data description (metadata) and management for scientific archives. Appendices include a rudimentary data dictionary for the current James Reserve database, and sample metadata crosswalks for Ecological Metadata Language (EML) and Sensor ML. This paper will cover current standards, developments, and sources of datasets (where appropriate) for:

- General data management and discovery - tools
- Environmental science/ecology
- Seismology
- Oceanography
- Atmospheric science
- Toxic hydrology
- Geographical Information Systems (GIS)
- Education

Introduction

The UCLA Center for Embedded Networked Sensing (CENS) is one of six Science and Technology Centers (STC's) to be established by the National Science Foundation in 2002. UCLA is the lead organization in a consortium of universities building dense wireless networks of sensors and actuators to observe and analyze biological and physical phenomena. The datasets generated by these networks, as well as associated scientific documents, records, and explanatory material will need to be managed and preserved for the use of both internal CENS researchers and external interested parties.

The purpose of this paper is to describe the state of the art in information and data management in the areas of research that comprise CENS, frameworks for scientific data management, and how they might be integrated. Using this backdrop, I will suggest next steps in developing a concrete information and data management plan.

Methodology

Information for this paper was gathered by searching for relevant government agencies, university projects, and other cooperative projects and discussions with CENS researchers, who suggested relevant standards, protocols, and organizations. These conversations led me to other subject specialists not affiliated with CENS. In a few cases, recent scholarly publications proved to be useful.

The Case for an Integrated Information and Data Management Plan

- The integrity and accessibility of data in CENS is crucial to its research and educational activities and to the multiple research groups within CENS.
- An increased emphasis by funding agencies data sharing and the re-use of expensive data and resources will mandate that large data providers implement plans for managing and disseminating data and other products.
- Appropriate data management and preservations strategies will make data available for longitudinal and multidisciplinary research, as well as for multiple audiences and purposes.
- Data and information must be preserved for risk management. Risks include the threats of data loss, the need to protect the intellectual work of CENS investigators, and the continued usability of data that may be needed to corroborate scientific claims.

Challenges

- Different communities of practice have evolved different systems for managing and describing data that are not mutually compatible. Some scientific disciplines have extensive tools, resources, and data archives for warehousing community datasets, while others do not.
- Sensor networks present technical challenges for data management. These include the volume of data collected, the "lossiness" of compression algorithms, the in-network processing of data (the kind of processing which might normally have been done by human analysis and thus be more easily documented), and differing spatiotemporal scales of data collection suggest potential challenges for data management and discovery. Sensors themselves, if they are to be managed and tasked remotely, need description.
- Tools for scientific data management, use, and discovery, particularly by non-experts, are still in their infancy.
- Describing data to be fully compliant with every discipline-specific standard will require far more resources and overhead than are available

Requirements/Components

An information and data management strategy/plan should incorporate at least the following components.

- Information architecture
    - o overarching framework for integrating data storage, retrieval, and access systems
    - o hardware, software, and middleware deployed in an appropriate configuration to make data storage and retrieval straightforward, configurable, and sustainable
- Metadata
    - o "data about data" – often discipline specific – that make data description, management, and discovery possible and systems interoperable
    - o metadata should help contextualize datasets. Metadata should describe data set format and content, the circumstances of data collection, procedures used to manipulate or model data, custodianship, data quality, preservation information, and discipline-specific description
- Policy and procedures:
    - o who can contribute to the data repository and specifications for data formats and metadata
    - o policy for data use and restrictions
    - o other intellectual property considerations
    - o data citation instructions
    - o procedures for storing, managing, accessing, using, and preserving data
- Staff
    - o necessary skills/competencies and responsibilities need to be identified
    - o funding for staff must be acquired

General Data Management and Discovery

CENS will need to adopt a framework for data management that allows for very high volumes of data, multiple data description standards (often for the same data set, depending upon intended use), and potentially distributed repositories of heterogeneous data. Although models of such data warehouses have been deployed in a number of arenas, there have been few standard technologies for making this an efficient process and as yet little research into the most user-friendly interfaces for such repositories. The following section describes some new systems and technologies for creating interfaces and back-ends for managing and describing scientific data and integrating multiple distributed repositories into an integrated whole.

- Dspace
- Storage Resource Broker (SRB) with MCAT– San Diego Supercomputing Center
- Virtual Seismic Network – Scripps Oceanographic Institute
- Virtual Object Ring Buffer (VORB)

MIT's Dspace (a joint project with Hewlett Packard) is an open-source platform durable digital repository provides another freeware resource for creating, and preserving digital works. It allows for the creation of metadata, supports textual works as well as data sets and learning objects, and is freely downloadable. UCLA's Advanced Technology Service has installed Dspace in its Technology Sandbox.

Other promising efforts in this area are from San Diego Supercomputing Center (SDSC). Their Storage Resource Broker client-server middleware has been designed to work with their Metadata Catalog (MCAT) product to access datasets by attributes rather than names. The latest release of this product was February 2003 (version 2) and is supported on SunOS, SGI, Solaris, Unix, and Oracle back-ends.

The MCAT is a repository of metadata at SDSC to store and query metadata with a uniform interface. System level metadata includes information on data, users, resources, and methods, which can be combined to create more complex data objects. SRB uses a proxy mechanism to act as the "user" so that humans do not need to have accounts on all remote systems. SRB has multiple client-server configurations, as well as interfaces for manipulating datasets, objects, data discovery, and metadata manipulation in the MCAT catalog.

.

Although the SRB technology can serve as usefully as middleware for a federated information system, it alone is not sufficient to allow for collecting and managing data in real-time from wireless networks. The most promising project in this arena appears to be the Virtual Seismic Net concept developed by Frank Vernon (of the Scripps Oceanographic Institute). Vernon's group is using field research with wireless sensor networks to integrate various kinds of datasets (ecological, earthquake, and other) seamlessly. Although the tools have not yet been developed (prototypes exist), they intend to extend the concept of Object Ring Buffers (ORBS) to virtual ORBs (VORBS) for managing multiple connections for multiple users. Many of these interactions will depend upon XML "wrappers" around information objects, with a programmable interface to reconfigure data capture as needed. Vernon's system is currently designed to implement the SDSC's SRB middleware.

Metadata

Metadata, most usefully, can be defined as descriptive information about digital information objects for access, use, preservation, interoperability, and management purposes. Metadata can be digital or non-digital, human generated or derived from the system, be managed separately from an information resource or be integral to it. Metadata can be structured or free-text. Some metadata will be static (creator, date of creation, format) while other may need to be modified as a digital object is used (ex: user and use statistics, preservation/migration).

Metadata will need to cover the following fundamental areas of description. Most existing metadata standards cover most, if not all, of these areas. The categories are drawn from the Getty Research Institute's metadata primer, with modifications to accommodate scientific data. The following section indicates general areas of metadata tagging, with examples.

Administrative: To manage data and other resources

Contact information for use

Rights and responsibilities of use

Descriptive: To describe, identify, or contextualize data

Discipline-specific tags to identify context under which data was created. This would include spatiotemporal information, methodology, protocols, and other scientific descriptors. Can also include metadata that would be useful for teachers, educators, and other non-expert users to employ the resource.

Preservation: To describe any preservation measures that might be used to maintain data

Documentation of condition

Documentation of steps taken to preserve data (migration to other systems, etc)

Technical: To describe how system and metadata behave

Data dictionary of data collection

Hardware/software requirements and use
System information
Network processing information
Security data
Metadata standards and version

Use: To capture how resource is used

Versioning information for datasets and other resources
Use and user information

Most of the discipline-specific metadata standards cover administrative and descriptive metadata, with less emphasis on technical metadata. I have not included general metadata standards (ex: Dublin Core), but have focused on discipline-specific areas.

Application/Research Area	**Metadata and other Data management standards**	**Format**	**Comments**
Sensors, Sensor Nets	(Sensor Model Language) SensorML	Reference model stage	Not standard yet. May be more useful for describing large scale sensors over, but model may be useful for future data discovery in CENS sensor networks
Ecology	Ecological Metadata Language (EML) Content Standards for Digital Geospatial Metadata (CSDGM)	XML	Optimized for ecological data, open source tools/implementation model available. Not optimized for sensor data. CSDGM is important for GIS data and can handle sensor data, but it's bulkiness makes it difficult to implement
Seismology	(Standard for Exchange of Earthquake Data) SEED (National NEESML (Network for Earthquake Engineeering Simulation Metadata Language)	SEED – ASCII NEESML - XML	SEED is an established, robust metadata schema in use for almost 30 years, but its ASCII format makes it unwieldy. NEES has developed a metadata language, but has yet to release specifics.
Oceanography	CSDGM, etc	Varied	Many data sets go back to the 1800's and are in wide variety of formats. Most data

			sets are deposited with federal government repositories, which engages in extensive metadata and data management of these resources
Atmospheric Sciences	CSDGM	XML	Many data sets go back to the 1800's and are in wide variety of formats. Most data sets are deposited with federal government repositories, which engages in extensive metadata and data management of these resources
Toxic hydrology	CSGDM, etc	Varied	Similar to oceanography
Geographical Information Systems (GIS)	CSGDM Geographical Metadata Language (GML)	XML	Metadata for "raw" data, as well as Arc suite of features, etc, possible. Government and other repositories available
Education	GEM SCORM ADN LOM	XML	Standards for describing "static" text-based resources and data discovery in educational learning environments, but no standards for data for educational use.

Sensors

SensorML, (Sensor Markup Language) was developed at the University of Alabama at Huntsville as an XML schema to define sensor attributes for data discovery, archiving sensor descriptions, localizing data, and analysis of performance. It is unusual in that it is a descriptive standard both for data and for hardware. SensorML can be used for static or dynamic sensors that are remote or in-situ. SensorML provides information related to the characteristics of the data being sensed, geometry of data collection, and the sensor's physical properties. A sensor can be described independently of the platform on which it resides, if the researcher wishes. For the purposes of geolocation, SensorML is compatible with and utilizes GML 3.0 (Geographic Markup Language) . SensorML became an OpenGIS standard in late 2002 – as yet, there are no tools for implementation, verification services, or other mechanisms for implementation.

Environmental Science/Ecology/Habitat Monitoring

The ecology community has long recognized that data management and metadata is crucial to their work – the lack of standards, diversity of data formats, and lack of well-documented datasets has slowed cross-institutional and longitudinal research. There has been a great deal of emphasis on standardizing geospatial data, but less on nonspatial data. Nonspatial ecological data is more diverse in format and scope. Because spatiotemporal data management (primarily GIS) is central to several CENS application areas, I will address it separately.

- Knowledge Network for Biocomplexity (KNB)
- Ecological Metadata Language (EML)
- Content Standard for Digital Geospatial Metadata (CSDGM) – Biological Profile
- Long Term Ecological Research network (LTER)
- Ameriflux
- National Center for Ecological Analysis and Synthesis (NCEAS)

The most important recent developments in ecological data management have been developed by several consortia and are currently overseen by Knowledge Network for Biocomplexity.

The KNB system for warehousing data takes advantage of this facility by using a three-layer architecture:

- EML on MetaCat metadata server to describe the dataset
- Quality Assurance and data integration engine (in development)

- Visualization tools for modeling and hypothesis development (in development)

EML, an XML-based data description standard, is a modular exchange standard for communicating metadata, is readily translatable into other metadata standards, and in conjunction with MetaCat, a multipurpose repository system, can be used as a framework for data management. The user can also create metadata over a Web browser that enables Morpho, a Java based, reverse engineerable, metadata entry and validation tool.

The biggest disadvantage of EML for CENS use is its relative lack of support for sensor-based data, although its ready extensibility and XML based natured make it relatively easy to add on relevant data tags. EML can also be ported to the FGDC's CSDGM standard, which, while far less flexible and more monolithic, offers more support for remote sensing data.

Several major ecological networks and projects have agreed to participate in the KNB, use its standards, and deposit their datasets. These include: the University of California Reserve System, many (but not all) of the Long Term Ecological Research Stations (LTERS), and individual ecological institutions.

Available Datasets
Oak Ridge National Laboratory's Mercury System for Metadata allows searching across data repositories for datasets described using the Government Information Locator System (GILS) or the CSDGM. The University of California Natural Reserve System and the Long Term Ecological Research Station Network also house numerous datasets, but their search engines are not user-friendly, making it harder to pinpoint useful datasets based on user-specified criteria. The National Biological Information Infrastructure (NBII) is a portal to a wide assortment of biological data set nodes. The NBII uses the Biological Profile of CSDGM.

The Ameriflux project also provides a number of data resources. The Ameriflux project has been designed to provide long-term, direct measurements of carbon dioxide and water vapor fluxes between terrestrial ecosystems and the atmosphere. CDIAC archives both preliminary Ameriflux datasets and those that are processed and made uniform by CDIAC, which are both available through FTP in text format, but with almost no accompanying metadata. Data is retrievable by a WWW interface by Ameriflux or by year (with combined sites) or by FTP.

Seismology

Seismic data has been collected via digital instruments for over 30 years. For this reason, there are robust and stable standards for describing seismic data across systems and data formats. Waveform analysis requires extensive metadata describing how the underlying data was processed. Recent efforts have focused on disseminating and integrating research data. There are no discipline-specific standards for describing seismic models, even though the development of models is at the core of seismology research.

- SEED (Standard for Exchange of Earthquake Data)
- Network for Earthquake Engineering Simulation (NEES)
- Incorporated Research Institutions for Seismology (IRIS)
- Southern California Earthquake Center (SCEC)

The most common, prevalent standard for data description and exchange of seismological data is SEED (Standard for the Exchange of Earthquake Data, v. 2.1), a stable international standard. SEED is a data format, not a metadata standard, and is optimized for temporal variation, not spatial variation of data and is not designed for unequal time interval data or non-time series data, or for processed datasets. SEED is designed for creation in machine-readable format with ASCII header files to describe the field station, the station network, and the event network. The formatted header files include information about the volume of data, the station channels, and the data itself. The standard has not evolved to take advantage of XML and other flexible, open standards, which makes searching for datasets is difficult.

The George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES) is a new consortium and NSF research initiative which will likely replace existing standards for exchanging earthquake data and engineering models. NEESgrid, the system integration component of NEES, is charged with developing data and metadata standards and policies for NEES. A "strawman" metadata model was released in 2002, as was a metadata service API, a proposed metadata structure for describing models, and a whitepaper detailing a metadata harvesting protocol. They have very recently released NEESML, an XML format for defining metadata schemes for the metadata repository, but have not included a comprehensive reference.

Available Datasets

The IRIS (Incorporated Research Institutions for Seismology) consortium is endeavoring to centralize and disseminate seismic datasets. They have datasets corresponding both to permanent and temporary sensor deployments. Datasets are in numerous formats – near real-time, archived data that has been quality-controlled, event-based datasets, pre-assembled datasets, and miscellaneous datasets. All of these datasets require different retrieval/querying mechanisms: anonymous FTP, email request, and some Web-enabled querying systems.

The Southern California Earthquake Center (SCEC) is also a source of datasets, but these are not searchable, only browseable. They include waveform data, earthquake catalogs, and other kinds of miscellaneous datasets.

Oceanography

- National Oceanographic and Atmospheric Agency (NOAA)
- National Center for Atmospheric Research (NCAR)

Oceanographic datasets share many characteristics with those in the atmospheric sciences – often very large, nonhomogeneous spatiotemporal distribution, collected via a wide variety of mechanisms (field sites, satellite data, among others), and multivariate. The National Oceanographic and Atmospheric Agency (NOAA) and National Center for Atmospheric Research (NCAR) are two of the largest sources of datasets and they have several internal centers that are responsible for archiving oceanographic data (particularly the National Oceanographic Data Center, NODC). Datasets at NOAA and other government agencies tend to be packaged into any number of "standard" scientific data formats. The metadata standard currently in use is the FGDC's CSGDM, but the difficulty of creating FGDC-compliant metadata records makes that one of several choices, including a format called Electronic Data Documentation Form (EDDF), a simple downloadable PDF form that one submits with one's data to the NOAA for inclusion in the NODC system.

Atmospheric Sciences

Atmospheric data comes in many forms, spans spatiotemporal scales, and encompasses field data, satellite readings, and other formats. Because of the importance, quantity, and widespread use of these data sets, NASA in particular has invested heavily in managing and disseminating remote sensing data. There are also a number of other government agencies and consortia involved in creating standards and managing datasets. However, the wide variety of research projects that constitute atmospheric science produce a vast array of data formats, data sets, and models. For this reason, data management in the atmospheric sciences is numerically intensive, widely distributed, and adheres to a wide set of best practices.

- Earth Observation System Data and Information System (EOSDIS)

The **EOSDIS gateway** is a data portal for Landsat and remote sensing data. Nine Distributed Active Archive Centers (DAACS), each of which is responsible for a different kind of data, manage these data sets. The EOS portal allows one request to be sent to all DAAC's, which must then manage the request. Data in the form of processed and unprocessed data sets and images are available. A metadata standard has been implemented in the core system **(ECS Earth Science Data Model)**. The Landsat 7 data retrieval system is part of the EOSDIS.

- NCAR/UCAR (University Corporation for Atmospheric Research)

    The CDIAC data management system offers scientists access to research data in geophysics. One can search for data sets and browse them, then retrieve them by FTP or on CD-ROM. Querying is by project name or spatiotemporal variables. Data varies widely in source, quality, data type, metadata, and spatiotemporal scales.

    The NCAR consortium has also developed the Distributed Oceanographic Data System (DODS) framework and data retrieval protocol o allow researchers to access data in many standard and non-standard data analysis and visualization packages (MatLab, Ferret, etc). Researchers can also make their data available on DODS servers. DODS runs on GNU. The DODS system supports non-standard data formats – users can submit their data using FreeForm data description language.

    CODIAC is UCAR's data management system for accessing and storing geophysical data from many projects and centers. Scientists have access to metadata, can browse data sets, then obtain them via FTP or CD-ROM.

Users can search for data by time period covered, geographical location (longitude/latitude), project name, and other information. Data sets vary also.

The National Oceanic and Atmospheric Administration (NOAA) operates many national data centers which collect, QC, and disseminate data. The centers have specific purposes. The National Climatic Data Center (NCDC) is one of the largest, with high volume, data sets from the mid 1800's, and data in 15 forms. The National Geophysical Data Center (NGDC) has geophysical datasets. The CDIAC (Carbon Dioxide Information Analysis Center) archives high quality, verified datasets of greenhouse and other gases. Satellite data is archived at the National Environmental Satellite, Data, and Information Services (NESDIS).

Toxic Hydrology (Contaminant Transport)

- United States Geological Survey (USGS)

The majority of data collected in toxic hydrology is field data. The United States Geological Survey (USGS) has a vast collection of available datasets, searchable by geographical area, kind of set (real time stream monitoring, water quality. Etc). The metadata is very straightforward and relates to geospatial information, type of data and its quality, well and aquifer identification, and agency that owns the information. However, there seem to be no agreed-upon standards for collecting field data. Most data can probably be accurately managed using GIS and ecological standards, with some modifications for unique information.

Geographical Information Systems (GIS)

- Content Standard for Digital Geospatial Metadata (CSDGM)-
- Geography Markup Language (GML)]

GIS cut across almost all CENS application areas. Metadata and visualization tools for such data are robust and standardized, and optimized for remote sensing. However, the dense nature of network data -- lossiness caused by data compression algorithms, the need to preserve calibration and modeling algorithms with raw data, volume of data, and variation in frequency rate – are difficult to express via existing GIS (and other discipline-specific) metadata standards.

The FGDC (Federal Geographic Data Committee) has developed the Content Standard for Digital Geospatial Metadata (CSDGM) which has become the most widely used standard for GIS data. The metadata captured includes standard identification information as well spatiotemporal coordinates. An extension set for remote sensing metadata is awaiting adoption; the extension for biological information has already been adopted.

The CSDGM has been criticized for its difficulty of implementation, its rigidity, and its inability to coordinate with other metadata standards. EML can be converted to CSDGM, however. Mp is a popular and widely used compiler for formal metadata that will check syntax against the standard and generate Web-enabled textual output. Mp is an open source program. The NBII and the FGDC are appropriate sources of GIS datasets.

The OpenGIS industrial consortium has been developing a vendor-neutral framework and standards to make geographic data exchange interoperable, easily exchanged, and readily discovered. Their Geography Markup Language (GML v. 3.0) is an XML-schema based metadata standard to separate geodata from the programs used to present or manipulate it. GML uses a set of basic geometry tags, a common data model, and a mechanism for creating domain-specific application schemas. GML utilizes the OpenGIS specifications for describing geographic entities/features. OpenGIS has other frameworks for data discovery, including a server architecture (Catalog Services Specification). Although GML will be useful for geospatial data description, there are as yet no tools to automate metadata entry and validation. It is also a component of SensorML.

Education

The educational mandates of CENS require that data and derived products be available and accessible to the educational community and to internal CENS education researchers. Educational materials and learning objects may be CENS datasets, but will also include educational modules, documents, and texts. Metadata for such materials will need to be targeted for the educational community to facilitate resource discovery, management, and evaluation. Current efforts are focused on describing text-based resources, not datasets.

- ADEPT/DLESE/NASA metadata standard (ADN)-
- Alexandria Digital Earth Prototype (ADEPT)-
- Digital Library for Earth Science Education (DLESE)
- Learning Object Metadata (LOM)
- Advanced Distributed Learning Shareable Content Object Reference Model (SCORM)
- GEM (Gateway for Educational Materials)-

The most developed data and metadata management for educational use has been developed by DLESE in conjunction with NASA and ADEPT (Alexandria Digital Earth Prototype) – ADN. The ADN standard works at three levels: required "item level description", robust item level description, and collection level description. The last is still under development. Elements for describing datasets are still forthcoming.

The IEEE has been developing a standard for describing learning objects that is called Learning Object Metadata (LOM). The standard focuses on a minimal set of attributes to describe learning objects (digital and non-digital materials used in technology supported learning). The set of descriptors indicate type of resource (lesson plan, inquiry module, presentation, syllabus, etc), learning objectives, support needed to implement the resource, target audience, ownership and use restrictions, and targeted content standards (international, federal, state, and local).

The Shareable Content Object Reference Model (SCORM 1.2), a framework for developing online learning environments, is emerging as an important standard. Building on the efforts of a number of organizations engaged in Web learning, SCORM is a reference model designed to create a standard among vendors for importing/exporting educational material to/from content repositories and developers of online learning materials. SCORM is being developed in concert with metadata standards such as LOM and IMS. Although SCORM may not be the most appropriate standard for describing datasets, its increasingly widespread use (including potentially by UCLA for describing digital learning object s) may necessitate the use of SCORM for reasons of interoperability with other systems.

The Gateway for Educational Materials (GEM) , a U.S. Department of Education initiative, is both a set of metadata tools for describing educational materials (GEM 2.0) as well as a portal to many uncatalogued educational resources (The Gateway). As a set of metadata standards, GEM 2.0 powers several major educational centers, including MarcoPolo, NASA's Space Science Education Resource Directory, and others. GEM provides tools for cataloguing database resources into GEM compatible formats (GEMCat), GEM Consortia members can include their materials into The Gateway.

Conclusions and Recommendations

Creating an integrated repository for CENS data, resources, and hardware that is fully compliant with all existing disciplinary standards for digital information is unrealistic. It would clearly require more resources than CENS possess, and would likely be overkill. However, the importance of CENS research and its products suggests that CENS should be a leader in the implementation of standards and technologies for multidisciplinary data and sensor network portals.

- Technologies for developing federated multidisciplinary data repositories are in early stages of research and development. Strategic partnerships with appropriate institutions engaged in such research (NCEAS, LTER's, NEON, etc), with CENS as a data testbed, should be developed. Such a partnership would also enable CENS to assess technical and other infrastructure that could be leveraged for data management
- Early efforts should focus on small, reachable goals that provide at least a minimum of documentation. A data dictionary of the data repository and some discipline-specific templates for describing resources for the internal Web site would be useful.
- Existing tools, data formats, and metadata standards should be evaluated by discipline-specific research groups and sensors themselves to ascertain their utility for CENS data. Researchers may choose an important subset for early use, with the possibility for adding on as data gathering activities proceed.
- Even before data is collected, CENS should institute a data use and intellectual property policy and develop a memorandum of understanding for non-CENS researchers who wish to use CENS technologies and research. The Office of Intellectual Property at UCLA can be consulted for guidance, if need be.
- The following steps still need to be conducted if CENS is to develop a robust information and data management system. All of these, except perhaps #5, will require the support and input of the CENS community of researchers.

1. Assess current and projected data management, use, and needs in each of the CENS groups.
2. Develop use scenarios to capture these requirements
3. Determine staffing requirements and sources of funding – a data archivist/librarian, programming staff, and database developers/managers will be needed, at the minimum.
4. Develop an information architecture that captures these use scenarios, as well as related systems
5. Develop a set of repository services, interfaces, and API's that applications can use to access and validate the data repositories
6. Populate the data and metadata repositories and implement other tools.
7. Evaluate the usability of the system and iterate as necessary, adding functionality and making adjustments as needed.

Appendix 1: CENS James Reserve Database Data Dictionary

This document describes the tables and fields of the current iteration of the CENS James Reserve schema (July 7, 2003)

CENS Schema Table	CENS DB	Data Format
CENS_Node: Each specific processing and transmitting node is created as a unique record in this table (ex: Micas, iPAQ's).	CENS_Node.Node_ID (Unique ID for each node)	Integer
	CENS_Node.Node_Type_ID (Refers to ID for node type, references CENS_NODE_TYPE_REF table)	Integer
	CENS_Node.Node_BOD (Born on Date of Node)	Date/Time
	CENS_Node.Node_Name (Name of Node)	Alphanumeric
	CENS_Node.Node_Desc (Description of Node)	Alphanumeric
CENS_Sensor: Each specific sensor is a unique record in this table	CENS_Sensor.Sensor_ID (Unique ID for each sensor)	Integer
	CENS_Sensor.Sensor_Type_ID (Refers to ID for sensor type, references CENS_SENSOR_TYPE_REF table)	Integer
	CENS_Sensor.Sensor_BOD (Born on Date of Sensor)	Date/Time
	CENS_Sensor.Sensor_Desc (Description of Sensor)	Alphanumeric
CENS_Sensor_Data: Records values that specific sensors return. Multichannel sensors enter each value separately This sensor ID is identified when application is configured	CENS_Sensor_Data.Datum_ID (Unique ID for each record of data)	Date/Time
	CENS_Sensor_Data.Sensor_ID (foreign key from CENS_Sensor)	Integer
	CENS_Sensor_Data.Datum_Value (value of data)	Number
	CENS_Sensor_Data.Remote_Time (remote timestamp)	Date/Time
	CENS_Sensor_Data.Time_Type_ID (references CENS_Time_Type_ID table to get type of time)	Integer
	CENS_Sensor_Data.System_Time (system time recorded)	Date/Time
CENS_Sensor_BLOB: Records values that specific sensors return but in opaque form. Multichannel sensors can enter date individually	CENS_Sensor_Blob.Datum_ID (Unique ID for BLOB))	Integer
	CENS_Sensor_Blob.Sensor_ID (Unique ID for sensor that picked up BLOB)	Integer

or in aggregate. Same field definitions as above. This sensor ID is identified when application is configured	CENS_Sensor_Blob.Datum_Blob (BLOB itself)	BLOB
	CENS_Sensor_Blob.Remote_Time (Time comes from sensor)	Date/Time
	CENS_Sensor_Blob.Time_Type_ID (Refers to type of time of Time_Type ID table)	Integer
	CENS_Sensor_Blob.System_Time (System time from database)	Date/Time
CENS_Location: Position for each sensor or node (TIMESTAMPED) CENS_SENSOR_LOC_REL table will associate sensors with location information CENS_NODE_LOC_REL table will associate nodes with location information	CENS_Location.Location_ID (Unique location ID)	Integer
	CENS_Location.X_Pos (Position on X axis; currently not in use)	Number
	CENS_Location.Y_Pos (Position on Y axis; currently not in use)	Number
	CENS_Location.Z_Pos (Position on Z axis; currently not in use)	Number
	CENS_Location.Time_Recorded (Time location was captured)	Number
	CENS_Location.Time_Type_ID (Refers to type of time of Time_Type ID table)	Integer
	CENS_Location.System_Time (database time)	Date/Time
	CENS_Location.Location_Type_ID (References CENS_Location_Type_Ref table and indicates normalized information of sensor or node)	Integer
Telemetry for each sensor (TIMESTAMPED) – CENS_SENSOR_TELEM_REL table will associate sensors with telemetry information CENS_NODE_TELEM_REL table will associate nodes with telemetry information	CENS_Telem.Telem_ID (Unique ID for telemetry)	Integer
	CENS_Telem.Pitch (forward and backward along short axis)	
	CENS_Telem.Roll (movement around long axis)	
	CENS_Telem.Yaw (left and right movement)	
	CENS_Telem.Time_Recorded (sensor time)	Date/Time
	CENS_Telem.Time_Type_ID (ID from time_type_ID table)	Integer
	CENS_Telem.System_Time (database time)	Date/Time
	CENS_Telem.Telem_Type_ID (References CENS_Telem_Type_Ref table and indicates normalized information of sensor or node)	Integer

CENS_Context: users can add annotations at any time	CENS_Context_ID (Unique ID for each annotation)	Integer
	CENS_Context_system_Time (time annotation was added)	Date/Time
	CENS_Context.Conatext_Name (name of annotation)	Alphanumeric
	CENS_ Context.Conatext _Desc (description alphanumeric)	Alphanumeric
CENS_Node_Type_Ref: describes classes of nodes and describes their attributes (expected cardinality with processor type: 1-1)	CENS_Node_Type_Ref.Node_Type_ID (Unique ID for each class of node)	Integer
	CENS_Node_Type_Ref.Node_Name (name of node class)	Alphanumeric
	CENS_Node_Type_Ref.Node_Desc (alphanumeric description of node class)	Alphanumeric
	CENS_Node_Type_Ref.Node_Vendor (vendor of node)	Alphanumeric
	CENS_Node_Type_Ref.Node_Version (version of node)	Alphanumeric
CENS_Sensor_Type_Ref: describes types of sensors being used and their attributes. Also provide information about units and generic meaning of data	CENS_Sensor_Type_Ref.Sensor_Type_ID (unique ID for each sensor type)	Integer
	CENS_Sensor_Type_Ref.Sensor_Channels (number of channels)	Integer
	CENS_Sensor_Type_Ref.Sensor_Units (units in which data will be captured)	Alphanumeric
	CENS_Sensor_Type_Ref.Sensor_Prec (precision of sensor)	Alphanumeric
	CENS_Sensor_Type_Ref.Sensor_Vendor (vendor of sensor)	Alphanumeric
	CENS_Sensor_Type_Ref.Sensor_Name (name of sensor type)	Alphanumeric
	CENS_Sensor_Type_Ref.Sensor_Desc (text description of sensor type)	Alphanumeric
	CENS_Sensor_Type_Ref.Sensor_Version (versoin of censor in use)	Alphanumeric
	CENS_Sensor_Type_Ref.Sensor_Error	Alphanumeric
CENS_Location_Type_Ref: enumerates ways locations can be classified (does not contain location	CENS_Location_Type_Ref.Location_Type_ID (Unique ID for each location type)	Integer

values) ID from this table can be used in other tables to indicate what kind of units are being used to define location		
	CENS_Location_Type_Ref.Loc_Units (units used to describe that kind of location)	Integer
	CENS_Location_Type_Ref.Loc_Name (name of location type)	Alphanumeric
	CENS_Location_Type_Ref.Loc_Desc (alphanumeric description of location type)	Alphanumeric
	CENS_Location_Type_Ref.Loc_Prec (how precise location is)	Alphanumeric
CENS_Telem_Type_Ref: enumerates ways telemetry can be classified (does not contain telemetry values). ID from this table can be used in other tables to indicate what kind of units are being used to define telemtry	CENS_Telem_Type_Ref.Telem_Type_ID (Unique ID of telemetry type)	Integer
	CENS_Telem_Type_Ref.Pitch_Res (resolution of pitch)	Alphanumeric
	CENS_Telem_Type_Ref.Roll_Res (resolution of roll)	Alphanumeric
	CENS_Telem_Type_Ref.Yaw_Res (resolution of yaw)	Alphanumeric
	CENS_Telem_Type_Ref.Pitch_Units (units in which pitch is measured)	Alphanumeric
	CENS_Telem_Type_Ref.Roll_Units (units in which roll is measured)	Alphanumeric
	CENS_Telem_Type_Ref.Yaw_Units (units in which yaw is measured)	Alphanumeric
	CENS_Telem_Type_Ref.Telem_Name (name of telemetry type)	Alphanumeric
	CENS_Telem_Type_Ref.Telem_Desc (Alphanumeric description of telemetry type)	Alphanumeric
CENS_Time_Type_Ref: normalizes meaning of time values in other table	CENS_Time_Type_Ref.Time_Type_ID (Unique ID for time value)	Integer
	CENS_Time_Type_Ref.Time_Res (time resolution)	Alphanumeric
	CENS_Time_Type_Ref.Time_Prec (Precision – how precise the measurement)	Alphanumeric

	CENS_Time_Type_Ref.Time_Units (units time is measured in)	Alphanumeric
	CENS_Time_Type_Ref.Time_Name (name of time value)	Alphanumeric
	CENS_Time_Type_Ref.Time_Desc (alphanumeric description of time value)	Alphanumeric
C CENS_PRJ_REF: Create project and hold high level data about it	CENS_PRJ_Ref.PRJ_ID (Unique project ID)	Integer
	CENS_PRJ_Ref.PRJ_Name (Name of Project)	Alphanumeric
	CENS_PRJ_Ref.PRJ_Start (Start date of project)	Date/Time
	CENS_PRJ_Ref.PRJ_End (End Date of project)	Date/Time
	CENS_PRJ_Ref.PRJ_Desc (Description of project)	Alphanumeric
	CENS_PRJ_Ref.PRJ_Contact (Name of person to contact about project)	Alphanumeric
CENS_Sensor_Reading_Type: Describes general meanings behind sensor readings in SENSOR_DATA or SENSOR_BLOB tables. Can use this table to record higher level data that is not generic and does not fit into other tables	CENS_Sensor_Reading_Type_Ref.Reading_ID (Unique ID for reading type)	Integer
	CENS_Sensor_Reading_Type_Ref.Reading_Name (name of sensor reading)	Alphanumeric
	CENS_Sensor_Reading_Type_Ref.Reading_Desc (alphanumeric description of sensor reading)	Alphanumeric
	CENS_Sensor_Reading_Type_Ref.Reading_Prec (how precise sensor reading is)	Alphanumeric
	CENS_Sensor_Reading_Type_Ref.Reading_Units (units to describe sensor reading)	Alphanumeric
This table will indicate position and telemetry for each node (TIMESTAMPED)	CENS_Node_Sensor_Rel (RELATIONAL TABLE)	
This table will indicate initial position and telemetry for each node (TIMESTAMPED)	CENS_Node_Prj_Rel (RELATIONAL TABLE)	
Relates specific node to a set of locations (1:m expected)	CENS_Node_Loc_Rel (RELATIONAL TABLE)	
Relates specific node to telemetries (1:m)	CENS_Node_Telem_Rel (RELATIONAL TABLE)	

Relates participation of sensor in all projects	CENS_Sensor_Prj_Rel (RELATIONAL TABLE)	
Relates sensor to set of locations	CENS_ Sensor_Loc_ Rel (RELATIONAL TABLE)	
Relates sensor to set of telemetries	CENS_ Sensor_Telem_ Rel (RELATIONAL TABLE)	
Relates a statement of conalphanumeric to a node at any time	CENS_Cont_Node_Rel (RELATIONAL TABLE)	
Relates a statement of conalphanumeric to a sensor at any time	CENS_Cont_Sensor_Rel (RELATIONAL TABLE)	
Relates sensors to others in hierachical fashion	CENS_Sensor_Rel (RELATIONAL TABLE)	

Appendix 2: CENS James Reserve database – Sensor ML crosswalk

This table attempts to map the SensorML schema against the CENS James Reserve database fields - a "crosswalk." SensorML is a modeling language for describing the sensor resources for sensor management and discovery. It does not describe sensor-derived data itself. SensorML is not yet at the level of an XML schema, which makes "translation" to the CENS database difficult. While it has extensive capability for describing moving sensors and platforms, many of the elements of the model relate to describing the platform upon which a sensor rests and geolocation information, which are not fully fleshed out. Platforms are described independently of sensors and nodes, which might be necessary in the future if platforms have characteristics independent of the sensors mounted on them.

CENS Schema Table	CENS DB	SensorML (OGC 02-026r4), 12/20/2
NS_Node: Each specific essing and transmitting e is created as a unique rd in this table (ex: Micas, Q's).	CENS_Node.Node_ID (Unique ID for each node)	AssetDescription:identificationNur (2.2.12)
	CENS_Node.Node_Type_ID (Refers to ID for node type, references CENS_NODE_TYPE_REF table)	Sml: IdentifiedAs (2.2.2).
	CENS_Node.Node_BOD (Born on Date of Node)	
	CENS_Node.Node_Name (Name of Node)	Sml: IdentifiedAs (2.2.2).
	CENS_Node.Node_Desc (Description of Node)	AssetDescription: sml:description (2.2.12)
NS_Sensor: Each specific or is a unique record in table	CENS_Sensor.Sensor_ID (Unique ID for each sensor)	AssetDescription:identificationNur (2.2.12)
	CENS_Sensor.Sensor_Type_ID (Refers to ID for sensor type, references CENS_SENSOR_TYPE_REF table)	Sml: IdentifiedAs (2.2.2).
	CENS_Sensor.Sensor_BOD (Born on Date of Sensor)	
	CENS_Sensor.Sensor_Desc (Description of Sensor)	AssetDescription: sml:description (2.2.12)
NS_Sensor_Data: Records es that specific sensors rn. Multichannel sensors r each value separately	CENS_Sensor_Data.Datum_ID (Unique ID for each record of data)	
	CENS_Sensor_Data.Sensor_ID (foreign key from CENS_Sensor)	
	CENS_Sensor_Data.Datum_Value (value of data)	

sensor ID is identified n application is igured	CENS_Sensor_Data.Remote_Time (remote timestamp)	
	CENS_Sensor_Data.Time_Type_ID (references CENS_Time_Type_ID table to get type of time)	
	CENS_Sensor_Data.System_Time (system time recorded)	
S_Sensor_BLOB: Records es that specific sensors rn but in opaque form. tichannel sensors can r date individually or in regate. Same field nitions as above.	CENS_Sensor_Blob.Datum_ID (Unique ID for Blob))	
	CENS_Sensor_Blob.Sensor_ID (Unique ID for sensor that picked up Blob))	
	CENS_Sensor_Blob.Datum_Blob (Blob itself)	
	CENS_Sensor_Blob.Remote_Time (time comes from sensor)	
sensor ID is identified n application is igured	CENS_Sensor_Blob.Time_Type_ID (Refers to type of time of Time_Type ID table)	
	CENS_Sensor_Blob.System_Time (system time from database)	
S_Location: Position for sensor or node MESTAMPED)	CENS_Location.Location_ID (Unique location ID)	CrsID (2.2.5)
	CENS_Location.X_Pos (position on X axis; currently not in use)	HasCRS (2.2.5), ObjectState (3.3.6)
S_SENSOR_LOC_REL e will associate sensors location information	CENS_Location.Y_Pos (Position on Y axis; currently not in use)	HasCRS (2.2.5) , ObjectState (3.3.6)
	CENS_Location.Z_Pos (Position on Z axis; currently not in use)	HasCRS (2.2.5) , ObjectState (3.3.6)
S_NODE_LOC_REL e will associate nodes location information	CENS_Location.Time_Recorded (Time location was captured)	
	CENS_Location.Time_Type_ID (Refers to type of time of Time_Type ID table)	
	CENS_Location.System_Time (database time)	
	CENS_Location.Location_Type_ID (References CENS_Location_Type_Ref table and indicates normalized information of sensor or node)	
metry for each sensor MESTAMPED) –	CENS_Telem.Telem_ID (Unique ID for telemetry)	
S_SENSOR_TELEM_REL e will associate sensors telemetry information	CENS_Telem.Pitch (forward and backward along short axis)	
	CENS_Telem.Roll (movement around long axis)	
	CENS_Telem.Yaw (left and right movement)	

NS_NODE_TELEM_REL e will associate nodes telemetry information	CENS_Telem.Time_Recorded (sensor time)	
	CENS_Telem.Time_Type_ID (ID from time_type_ID table)	
	CENS_Telem.System_Time (database time)	
	CENS_Telem.Telem_Type_ID (References CENS_Telem_Type_Ref table and indicates normalized information of sensor or node)	
NS_Context: users can add otations at any time	CENS_Context.Context_ID (Unique ID for each annotation)	
	CENS_Context.System_Time (time annotation was added)	
	CENS_Context.Context_Name (name of annotation)	
	CENS_Context.Context_Desc (description text)	AssetEvent 2.2.12)
NS_Node_Type_Ref: ribes classes of nodes and ribtes their attributes ected cardinality with essor type: 1-1)	CENS_Node_Type_Ref.Node_Type_ID (Unique ID for each class of node)	Sml: IdentifiedAs (2.2.2).
	CENS_Node_Type_Ref.Node_Name (name of node class)	
	CENS_Node_Type_Ref.Node_Desc (text description of node class)	
	CENS_Node_Type_Ref.Node_Vendor (vendor of node)	AssetDescription: manufacturedby
	CENS_Node_Type_Ref.Node_Version (version of node)	
NS_Sensor_Type_Ref: ribes types of sensors g used and their butes. Also provide rmation about units and eric meaning of data	CENS_Sensor_Type_Ref.Sensor_Type_ID (unique ID for each sensor type)	Sml:Measurable (2.2.8)
	CENS_Sensor_Type_Ref.Sensor_Channels (number of channels)	
	CENS_Sensor_Type_Ref.Sensor_Units (units in which data will be captured)	
	CENS_Sensor_Type_Ref.Sensor_Prec (precision of sensor)	CharacterizedBy – ows: TypedValu (2.2.8)
	CENS_Sensor_Type_Ref.Sensor_Vendor (vendor of sensor)	
	CENS_Sensor_Type_Ref.Sensor_Name (name of sensor type)	Sml: IdentifiedAs (2.2.2).

	sensor type)	
	CENS_Sensor_Type_Ref.Sensor_Desc (text description of sensor type)	
	CENS_Sensor_Type_Ref.Sensor_Version (versoin of censor in use)	
	CENS_Sensor_Type_Ref.Sensor_Error	
NS_Location_Type_Ref: merates ways locations be classified (does not ain location values) ID n this table can be used in r tables to indicate what l of units are being used efine location	CENS_Location_Type_Ref.Location_Type_ID (Unique ID for each location type)	
	CENS_Location_Type_Ref.Loc_Units (units used to describe that kind of location)	
	CENS_Location_Type_Ref.Loc_Name (name of location type)	
	CENS_Location_Type_Ref.Loc_Desc (text description of location type)	
	CENS_Location_Type_Ref.Loc_Prec (how precise location is)	CharacterizedBy (2.2.8)
NS_Telem_Type_Ref: merates ways telemetry be classified (does not ain telemetry values). ID n this table can be used in r tables to indicate what l of units are being used efine telemetry	CENS_Telem_Type_Ref.Telem_Type_ID (Unique ID of telemetry type)	
	CENS_Telem_Type_Ref.Pitch_Res (resolution of pitch)	
	CENS_Telem_Type_Ref.Roll_Res (resolution of roll)	
	CENS_Telem_Type_Ref.Yaw_Res (resolution of yaw)	
	CENS_Telem_Type_Ref.Pitch_Units (units in which pitch is measured)	
	CENS_Telem_Type_Ref.Roll_Units (units in which roll is measured)	

	CENS_Telem_Type_Ref.Yaw_Units (units in which yaw is measured)	
	CENS_Telem_Type_Ref.Telem_Name (name of telemetry type)	
	CENS_Telem_Type_Ref.Telem_Desc (Text description of telemetry type)	
NS_Time_Type_Ref: nalizes meaning of time es in other table	CENS_Time_Type_Ref.Time_Type_ID (Unique ID for time value)	
	CENS_Time_Type_Ref.Time_Res (time resolution)	
	CENS_Time_Type_Ref.Time_Prec (Precision – how precise the measurement)	CharacterizedBy (2.2.8)
	CENS_Time_Type_Ref.Time_Units (units time is measured in)	
	CENS_Time_Type_Ref.Time_Name (name of time value)	
	CENS_Time_Type_Ref.Time_Desc (text description of time value)	
CENS_PRJ_REF: Create ect and hold high level about it	CENS_PRJ_Ref.PRJ_ID (Unique project ID)	
	CENS_PRJ_Ref.PRJ_Name (Name of Project)	
	CENS_PRJ_Ref.PRJ_Start (Start date of project)	
	CENS_PRJ_Ref.PRJ_End (End Date of project)	
	CENS_PRJ_Ref.PRJ_Desc (Description of project)	
	CENS_PRJ_Ref.PRJ_Contact (Name of person to contact about project)	OperatedBy: sml: ResponsiblePart - contactInfo, IndividualName, organizationName, positionName (2..2.10)
NS_Sensor_Reading_Type: cribes general meanings nd sensor readings in SOR_DATA or SOR_BLOB tables. Can this table to record higher l data that is not generic does not fit into other es	CENS_Sensor_Reading_Type_Ref.Reading_ID (Unique ID for reading type)	

	CENS_Sensor_Reading_Type_Ref.Reading_Name (name of sensor reading)	
	CENS_Sensor_Reading_Type_Ref.Reading_Desc (text description of sensor reading)	
	CENS_Sensor_Reading_Type_Ref.Reading_Prec (how precise sensor reading is)	CharacterizedBy (2.2.8)
	CENS_Sensor_Reading_Type_Ref.Reading_Units (units to describe sensor reading)	
table will indicate tion and telemetry for node (TIMESTAMPED)	CENS_Node_Sensor_Rel (RELATIONAL TABLE)	
table will indicate initial tion and telemetry for node (TIMESTAMPED)	CENS_Node_Prj_Rel (RELATIONAL TABLE)	
tes specific node to a set cations (1:m expected)	CENS_Node_Loc_Rel (RELATIONAL TABLE)	
tes specific node to metries (1:m)	CENS_Node_Telem_Rel (RELATIONAL TABLE)	
tes participation of sensor l projects	CENS_Sensor_Prj_Rel (RELATIONAL TABLE)	
tes sensor to set of tions	CENS_ Sensor_Loc_ Rel (RELATIONAL TABLE)	
tes sensor to set of metries	CENS_ Sensor_Telem_ Rel (RELATIONAL TABLE)	
tes a statement of context node at any time	CENS_Cont_Node_Rel (RELATIONAL TABLE)	
tes a statement of context sensor at any time	CENS_Cont_Sensor_Rel (RELATIONAL TABLE)	
tes sensors to others in achical fashion	CENS_Sensor_Rel (RELATIONAL TABLE)	
		DocumentConstrainedBy (2.2.3) Platform (2.2.4) ObjectState (for moving platforms) 2.2.6 GetCoordinate (2.2.7) – for sensors get coordinates from Web or other service DocumentMetadata (2.2.13)

Appendix 3: CENS James Reserve database – Ecological Metadata Language (EML) crosswalk

This table was derived by matching , as best as possible, fields in the the current CENS database schema in use for the James Reserve (in use as of June 12, 2003) against the most recent iteration of Ecological Metadata Language (EML 2.0). The table suggests that EML is optimized for describing data, not the derivation of that data (i.e., sensor networks).

Highlights:

1) EML describes spatiotemporal variables more thoroughly than the current schema allows for. However, the CENS_Time_Type table in the current CENS database permits the creation of any number of time types.
2) Geographical information may need to be more fleshed out in the schema if research needs demand it.
3) EML is particularly weak on describing sensor data – most ecologists do not use sensor-derived data.

The CENS schema may need to be expanded to include levels of access/security information, if human subjects, environmentally sensitive regions, etc are sensed.

CENS	CENS DB	EML 2.0
NS_Node: Each specific cessing and transmitting node is ted as a unique record in this e (ex: Micas, iPAQ's).	CENS_Node.Node_ID (Unique ID for each node)	
	CENS_Node.Node_Type_ID (Refers to ID for node type, references CENS_NODE_TYPE_REF table)	
	CENS_Node.Node_BOD (Born on Date of Node)	
	CENS_Node.Node_Name (Name of Node)	
	CENS_Node.Node_Desc (Description of Node)	
NS_Sensor: Each specific sensor unique record in this table	CENS_Sensor.Sensor_ID (Unique ID for each sensor)	
	CENS_Sensor.Sensor_Type_ID (Refers to ID for sensor type, references CENS_SENSOR_TYPE_REF table)	
	CENS_Sensor.Sensor_BOD (Born on Date of Sensor)	
	CENS_Sensor.Sensor_Desc (Description of Sensor)	

NS_Sensor_Data: Records ues that specific sensors return. tichannel sensors enter each ue separately s sensor ID is identified when lication is configured	CENS_Sensor_Data.Datum_ID (Unique ID for each record of data)	
	CENS_Sensor_Data.Sensor_ID (foreign key from CENS_Sensor)	
	CENS_Sensor_Data.Datum_Value (value of data)	
	CENS_Sensor_Data.Remote_Time (remote timestamp)	Eml-Coverage-TemporalCoverage (2.4.4)
	CENS_Sensor_Data.Time_Type_ID (references CENS_Time_Type_ID table to get type of time)	Eml-Coverage (2.4.4)
	CENS_Sensor_Data.System_Time (system time recorded)	Eml-Coverage-TemporalCoverage (2.4.4)
NS_Sensor_BLOB: Records ues that specific sensors return in opaque form. Multichannel sors can enter date individually n aggregate. Same field nitions as above. s sensor ID is identified when lication is configured	CENS_Sensor_Blob.Datum_ID (Unique ID for Blob))	
	CENS_Sensor_Blob.Sensor_ID (Unique ID for sensor that picked up Blob))	
	CENS_Sensor_Blob.Datum_Blob (Blob itself)	
	CENS_Sensor_Blob.Remote_Time (time comes from sensor)	Eml-Coverage-TemporalCoverage (2.4.4)
	CENS_Sensor_Blob.Time_Type_ID (Refers to type of time of Time_Type ID table)	Eml-Coverage (2.4.4)
	CENS_Sensor_Blob.System_Time (system time from database)	Eml-Coverage-TemporalCoverage (2.4.4)
NS_Location: Position for each sor or node (TIMESTAMPED) NS_SENSOR_LOC_REL table associate sensors with location rmation NS_NODE_LOC_REL table will ociate nodes with location rmation	CENS_Location.Location_ID (Unique location ID)	Eml-Coverage (2.4.4)
	CENS_Location.X_Pos (position on X axis; currently not in use)	Eml-Coverage-GeographicCoverage (2.4.4)
	CENS_Location.Y_Pos (Position on Y axis; currently not in use)	Eml-Coverage-GeographicCoverage (2.4.4)
	CENS_Location.Z_Pos (Position on Z axis; currently not in use)	Eml-Coverage-GeographicCoverage (2.4.4)
	CENS_Location.Time_Recorded (Time location was captured)	Eml-Coverage-TemporalCoverage (2.4.4)
	CENS_Location.Time_Type_ID (Refers to type of time of Time_Type ID table)	Eml-Coverage (2.4.4)
	CENS_Location.System_Time (database time)	Eml-Coverage – TemporalCoverage (2.4.4)
	CENS_Location.Location_Type_ID (References CENS_Location_Type_Ref table and indicates normalized information of sensor or node)	Eml-Coverage (2.4.4)
emetry for each sensor MESTAMPED) –	CENS_Telem.Telem_ID (Unique ID for telemetry)	

MESTAMPED) – NS_SENSOR_TELEM_REL table associate sensors with metry information NS_NODE_TELEM_REL table associate nodes with telemetry rmation	CENS_Telem.Pitch (forward and backward along short axis)	
	CENS_Telem.Roll (movement around long axis)	
	CENS_Telem.Yaw (left and right movement)	
	CENS_Telem.Time_Recorded (sensor time)	
	CENS_Telem.Time_Type_ID (ID from time_type_ID table)	
	CENS_Telem.System_Time (database time)	
	CENS_Telem.Telem_Type_ID (References CENS_Telem_Type_Ref table and indicates normalized information of sensor or node)	
NS_Context: users can add otations at any time	CENS_Context.Context_ID (Unique ID for each annotation)	
	CENS_Context.System_Time (time annotation was added)	
	CENS_Context.Context_Name (name of annotation)	
	CENS_Context.Context_Desc (description text)	
NS_Node_Type_Ref: describes ses of nodes and describtes r attributes (expected linality with processor type: 1-	CENS_Node_Type_Ref.Node_Type_ID (Unique ID for each class of node)	
	CENS_Node_Type_Ref.Node_Name (name of node class)	
	CENS_Node_Type_Ref.Node_Desc (text description of node class)	
	CENS_Node_Type_Ref.Node_Vendor (vendor of node)	
	CENS_Node_Type_Ref.Node_Version (version of node)	
NS_Sensor_Type_Ref: describes es of sensors being used and r attributes. Also provide rmation about units and eric meaning of data	CENS_Sensor_Type_Ref.Sensor_Type_ID (unique ID for each sensor type)	
	CENS_Sensor_Type_Ref.Sensor_Channels (number of channels)	
	CENS_Sensor_Type_Ref.Sensor_Units (units in which data will be captured)	

	CENS_Sensor_Type_Ref.Sensor_Prec (precision of sensor)	
	CENS_Sensor_Type_Ref.Sensor_Vendor (vendor of sensor)	
	CENS_Sensor_Type_Ref.Sensor_Name (name of sensor type)	
	CENS_Sensor_Type_Ref.Sensor_Desc (text description of sensor type)	
	CENS_Sensor_Type_Ref.Sensor_Version (versoin of censor in use)	
	CENS_Sensor_Type_Ref.Sensor_Error	
NS_Location_Type_Ref: merates ways locations can be sified (does not contain location ues) ID from this table can be d in other tables to indicate at kind of units are being used efine location	CENS_Location_Type_Ref.Location_Type_ID (Unique ID for each location type)	
	CENS_Location_Type_Ref.Loc_Units (units used to describe that kind of location)	
	CENS_Location_Type_Ref.Loc_Name (name of location type)	
	CENS_Location_Type_Ref.Loc_Desc (text description of location type)	
	CENS_Location_Type_Ref.Loc_Prec (how precise location is)	
NS_Telem_Type_Ref: merates ways telemetry can be sified (does not contain metry values). ID from this e can be used in other tables to cate what kind of units are g used to define telemetry	CENS_Telem_Type_Ref.Telem_Type_ID (Unique ID of telemetry type)	
	CENS_Telem_Type_Ref.Pitch_Res (resolution of pitch)	
	CENS_Telem_Type_Ref.Roll_Res (resolution of roll)	
	CENS_Telem_Type_Ref.Yaw_Res (resolution of yaw)	
	CENS_Telem_Type_Ref.Pitch_Units (units in which pitch is measured)	

	CENS_Telem_Type_Ref.Roll_Units (units in which roll is measured)	
	CENS_Telem_Type_Ref.Yaw_Units (units in which yaw is measured)	
	CENS_Telem_Type_Ref.Telem_Name (name of telemetry type)	
	CENS_Telem_Type_Ref.Telem_Desc (Text description of telemetry type)	
NS_Time_Type_Ref: normalizes aning of time values in other e	CENS_Time_Type_Ref.Time_Type_ID (Unique ID for time value)	
	CENS_Time_Type_Ref.Time_Res (time resolution)	
	CENS_Time_Type_Ref.Time_Prec (Precision – how precise the measurement)	
	CENS_Time_Type_Ref.Time_Units (units time is measured in)	
	CENS_Time_Type_Ref.Time_Name (name of time value)	
	CENS_Time_Type_Ref.Time_Desc (text description of time value)	
CENS_PRJ_REF: Create project hold high level data about it	CENS_PRJ_Ref.PRJ_ID (Unique project ID)	Eml-project-id (2.4.5)
	CENS_PRJ_Ref.PRJ_Name (Name of Project)	Eml-project-name (2.4.5)
	CENS_PRJ_Ref.PRJ_Start (Start date of project)	Eml-project (2.4.5)
	CENS_PRJ_Ref.PRJ_End (End Date of project)	Eml-project (2.4.5)
	CENS_PRJ_Ref.PRJ_Desc (Description of project)	Eml-project-description (2.4.5
	CENS_PRJ_Ref.PRJ_Contact (Name of person to contact about project)	Eml-Party-IndividualName (2.4.3)
NS_Sensor_Reading_Type: cribes general meanings behind sor readings in SENSOR_DATA ENSOR_BLOB tables. Can use table to record higher level that is not generic and does fit into other tables	CENS_Sensor_Reading_Type_Ref.Reading_ID (Unique ID for reading type)	
	CENS_Sensor_Reading_Type_Ref.Reading_Name (name of sensor reading)	

	CENS_Sensor_Reading_Type_Ref.Reading_Desc (text description of sensor reading)	
	CENS_Sensor_Reading_Type_Ref.Reading_Prec (how precise sensor reading is)	
	CENS_Sensor_Reading_Type_Ref.Reading_Units (units to describe sensor reading)	
s table will indicate position and metry for each node MESTAMPED)	CENS_Node_Sensor_Rel (RELATIONAL TABLE)	
s table will indicate initial ition and telemetry for each e (TIMESTAMPED)	CENS_Node_Prj_Rel (RELATIONAL TABLE)	
ites specific node to a set of itions (1:m expected)	CENS_Node_Loc_Rel (RELATIONAL TABLE)	
ites specific node to telemetries n)	CENS_Node_Telem_Rel (RELATIONAL TABLE)	
ites participation of sensor in all ects	CENS_Sensor_Prj_Rel (RELATIONAL TABLE)	
ites sensor to set of locations	CENS_ Sensor_Loc_ Rel (RELATIONAL TABLE)	
ites sensor to set of telemetries	CENS_ Sensor_Telem_ Rel (RELATIONAL TABLE)	
ites a statement of context to a e at any time	CENS_Cont_Node_Rel (RELATIONAL TABLE)	
ites a statement of context to a sor at any time	CENS_Cont_Sensor_Rel (RELATIONAL TABLE)	
ites sensors to others in achical fashion	CENS_Sensor_Rel (RELATIONAL TABLE)	