# MODELING GENERIC OCEANOGRAPHIC DATA OBJECTS IN XML

*The authors define data objects, or bricks, for the ocean-data community based on commonalities across many data types used in ocean environmental research. They then develop a data exchange structure in an XML environment.*

Many nations have scientific or economic interests in the oceans, making oceanographic research an international endeavor. Such research has a high cost, so a strong incentive exists for collaborative programs, which must be able to exchange data quickly and efficiently. Strong international collaboration is also necessary to ensure that oceanographic observations are preserved and available.

Over the past few years, XML has received considerable attention as a method of data exchange primarily oriented toward dynamic Web page business-application development. However, XML might also have a data exchange role in an oceanographic context. (More information about XML in a computer science context is available elsewhere.[1]) Using data objects we've developed based on commonalities across oceanographic data types, we've implemented a data exchange structure in XML.

1521-9615/05/$20.00 © 2005 IEEE
Copublished by the IEEE CS and the AIP

ANTHONY W. ISENOR
*Defence R&D Canada—Atlantic*
J. ROBERT KEELEY
*Marine Environmental Data Service, Fisheries and Oceans Canada*

## Coordination across the Oceans

Data exchanges occur between researchers and their national archive centers, researchers in different countries, and national and international data centers. Organizations such as the International Council for the Exploration of the Sea (ICES; www.ices.dk) and the International Oceanographic Data and Information Exchange System (IODE; http://ioc.unesco.org/iode/)—a body of the Intergovernmental Oceanographic Commission (IOC)—are deeply involved in coordinating data exchanges. To encourage coordination these organizations form working groups to investigate technologies that have general applicability to member states.

In 2001, ICES and IOC formed the Study Group on the Development of Marine Data Exchange Systems Using XML (or the Study Group on XML [SGXML]), composed of 20 members from 14 nations.[2,3] The group's purpose was to examine the use of XML in marine data exchange, explore how to use XML structures to represent many ocean-data types using common data objects, and test and refine these structures using case studies involving point data, metadata, and biological data.

As a member state, Canada has contributed to the SGXML by developing generic data objects that are common across many types of oceanographic data (see www.meds-sdmm.dfo-mpo.gc.ca/meds/Prog_Int/ICES/ICES_e.htm).[4] These objects represent building blocks (or bricks) that we can use to construct ocean-data structures. The basic premise is that we can assemble a common set (and small number) of bricks in various ways to construct different structures that will suitably represent different types of ocean data.

Ocean-data types are a conceptual data grouping based in part on historical collection methods. Scientists have been collecting certain types—such as *point data*, gathered at a specific point in space and time (for example, a chemical measurement of salt content)—since the first ocean research cruise on *HMS Challenger* in 1872. Other data types include *profile data*, collected via a sensor lowered through the water column that gathers data during its descent (for example, temperature measurements at successive depths); *surface-drifter data*, collected via sensors freely drifting at the ocean surface (such as ocean temperature, air temperature, and air pressure); *profiling float data*, collected via sensors drifting and oscillating through the water column (such as temperature collected via sensors on an Argo float); and *time-series data*, collected over time via sensors fixed in position (such as water elevations at a coastal site).

Traditionally, ocean-data centers have constructed data-processing systems that address the particular data types their national researchers collect. These systems have typically evolved from legacy systems based on unique file structures; in some cases, they include normalized databases, whereas in others, they use other technologies. No matter what the archive technology, though, any international data structure used to exchange data between such systems must be able to store data from all the national systems in the member states.

Here, we develop a small set of bricks that we can use to build a data structure that doesn't conform to a known database. Instead, it assembles information in different patterns to meet the needs of a generalized data type, which is based on the number of independent variables in the spatial–temporal domain. In this exercise, we consider any one-dimensional (1D) data, or data with one independent variable (for example, time for time-series data or depth for profile data) to be similar.

## Conceptual Phase

In construction, bricks are a basic building material; they come in assorted sizes and colors. Although they have only a few basic shapes, builders can combine these few shapes to produce myriad structures, from outdoor fireplaces to apartment buildings.

In the ocean research community, data works in a similar fashion—we collect many different kinds to explore specific science questions, but these data have certain common content. For example, a collected data set has variables defined within it, is collected at a specific location in space and time, and might have a processing history. This common content forms the basis for our ocean-data bricks.

Examining many ocean-data types lets us recognize and describe the common content, so international data centers, which receive several different data, are an ideal place to do so. Once we define these groups (bricks) of common content, we can combine them to create different structures suitable for containing different data sets.

Figure 1 shows a conceptual view of how we can assemble six different bricks to contain data and information for 1D ocean-profile data (Figure 1a) and 3D surface-drifter data (Figure 1b). The different bricks describe a data collection as one or more data sets. Each data set might contain variable definitions, location and processing information, other data sets, and data points.

In the 1D case, the profile illustrates what we might think of as a vertical sampling at a fixed location. An example would be measuring temperature as a function of depth at a single $x$, $y$, $t$ (that is, longitude, latitude, and time) location. In this case, the $x$, $y$, $t$ information is common to all data points, thus we place it in a location brick that the second-level data set contains. We package the $z$ (that is, depth) information with the data in the fourth-level data set.

In contrast, a track might originate from a surface drifter, which moves in $x$, $y$, $t$ space with sensors fixed in the vertical ($z$). Each data point then requires the location to specify the $x$, $y$, $t$ component.

The figure doesn't show all the bricks that we defined; we removed much of the detail to illustrate the essential idea of combining common bricks to replicate two different data types used in ocean research. The bricks' content, and the definitions associated with this content, are independent of the XML (or any other) implementation.

## Definition Phase

We've devoted considerable time and effort to defining content for the bricks, and have attached these definitions to the individual data items that make up the bricks.[4] The details of the data structure have also evolved, with preliminary work oriented toward data with one independent variable. A vertical profile of temperature, for example, has a single independent variable—namely, depth. A time-series record has the same form as a profile, except the independent variable is time. We've also decided how to physically represent data items in the XML environment—that is, should the data

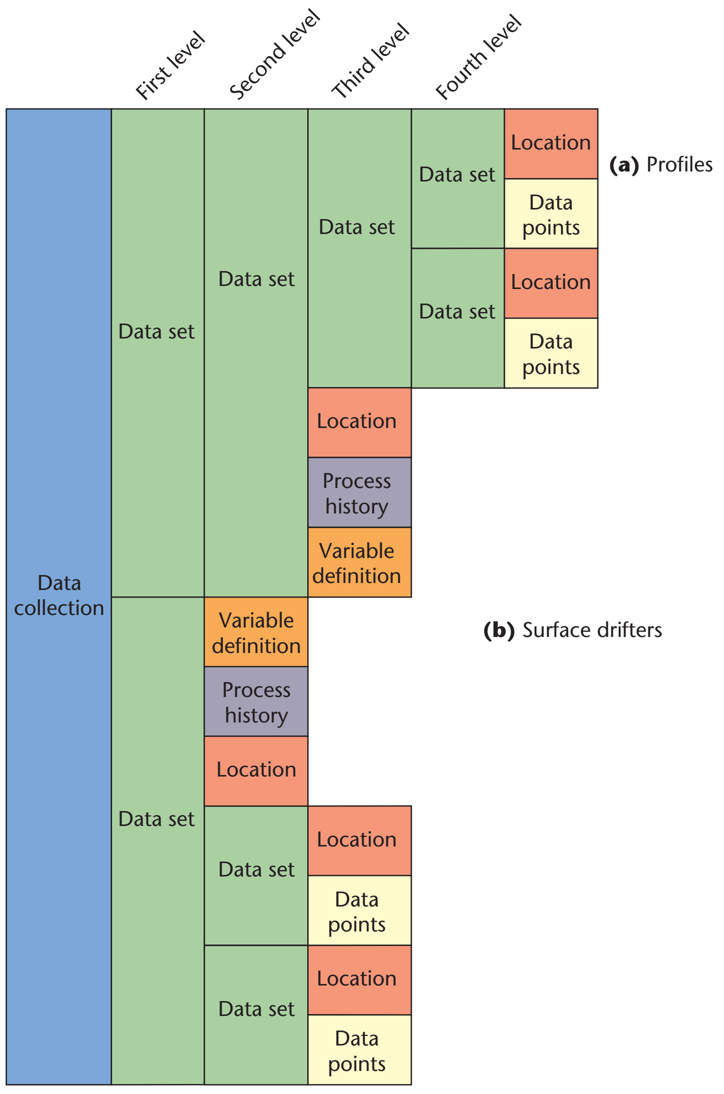

**Figure 1. Combined bricks creating ocean-data structures. Each brick contains the bricks to its right. The data collection (blue) contains two data sets (green) at the first level. (a) The profile structure shows that both the second- and fourth-level data sets contain location bricks; the fourth-level data set also contains data points. (b) The surface-drifter structure shows a first-level data set containing two additional data sets; the first-level data set also contains the variable definitions, history, and location information.**

item appear as an XML element or as an attribute?[1]

The use of elements and attributes is an important issue surrounding any XML implementation—often, we can store the data content equally well in either. For our implementation, we decided that if the content could be described in a list, then we would define the data unit as an attribute of the brick. We felt this was the simplest implementation for list content. We also thought it important to construct such a rule for using elements and attributes to ensure consistency throughout the final structure.

An example of the element–attribute decision is the parameter code. In the ocean-data community, you might describe a data parameter such as temperature to use an abbreviated set of characters that constitute a code—for example, you might use the code `TEMP` to represent temperature. The code will have a formal definition, which could include the instrumentation used to collect the parameter or the parameter units. Particular code definitions in the international community show various levels of detail.

In our XML implementation, we defined the code content using the attribute `pt_code`. Although code lists, or dictionaries, can reach thousands of entries, using the `pt_code` attribute frees the XML element to store the particular data value as the content. This way, we can contain the code and value in a single XML element.

## A Data Collection

The first brick describes a data collection, which is an all-encompassing data unit. In the XML implementation, this translates directly to the document element. A data collection contains various data sets, which could be from a single originator and be described by an encompassing volume in space and time. Free comments might also form part of a data collection; the data contained in the collection should have a primary source for the parameter code definitions.

Implementing the data collection definition in the XML environment means we describe the `<data_collection>` document element as a group of subelements:

```
<data_collection>
       <comment>                  [0,n]
       <data_dictionary>          [0,1]
       <provenance>               [0,1]
       <location_set>             [0,n]
       <data_set>                 [0,n]
</data_collection>
```

Here, we've augmented the XML tags with trailers indicating how often each subelement occurs (for example, `[0,1]` indicates zero or one occurrence). Each subelement might contain attributes or other subelements, as we describe later. We've omitted end tags from subelements for clarity.

Earlier, we mentioned parameter codes, as well as dictionaries that house the code lists. The `<data_dictionary>` element describes the primary dictionary being used within the data collection. The content is a simple text string indicating the dictionary's name. Data dictionary specification is an important concept—currently, no common dictionary exists for international exchange, thus the `<data_dictionary>` element is necessary to accommodate dictionary differences between centers.

## A Data Set

The oceanographic community uses data sets widely, but the concept has few concrete or consistent definitions, so it means different things to different people. With this in mind, our *data set* brick must allow for myriad definitions, so that participating organizations can use the defined structure.

We can identify several aspects or characteristics of a data set: it can contain data values (or data points), for example, and it probably has some type of identifier that the originating organization uses to uniquely identify it. The data set we identified for this investigation included the following characteristics:

- availability for distribution,
- free-format comments,
- data points within the data set,
- a data set identifier,
- the data set's originator,
- the applied-quality test results,
- a description of the applied-quality tests,
- the variables the data set contains,
- spatial–temporal location information,
- processing history, and
- a data set at a finer level of granularity.

As the last bullet shows, a data set can also contain a data set, a structure that exploits the natural hierarchy familiar in oceanographic data collections (for example, many vertical profiles at a single location and many locations sampled during a research cruise).

In the XML implementation, we might represent the data set as follows:

```
<data_set>
       <availability>             [0,1]
       <comment>                  [0,n]
       <data_point>               [0,n]
       <data_set_id>              [0,1]
       <provenance>               [0,1]
       <quality>                  [0,n]
       <quality_testing>          [0,n]
       <variable_set>             [0,n]
       <location_set>             [0,n]
       <history_set>              [0,n]
       <data_set>                 [0,n]
</data_set>
```

A data set's explicit definition uses XML's flexibility by allowing zero, one, or more occurrences of many elements. In the XML structure, for instance, we represent the availability brick as `<availability>`, and it occurs zero or one times in a `<data_set>`.

### Availability

The *availability* brick is composed of information relevant to data set distribution. This brick describes the data set's status and the date it attained this status. In the XML implementation, the `<availability>` element contains a single attribute, `indicator`, which is a flag to describe availability status, such as open, consultation required, or restricted. The element also includes a date, indicating when the indicator attribute was set. The date format is standard ISO 8601 as described in the XML Schema specification[5] (the bold notation indicates mandatory attributes or elements):

```
<availability indicator="">
     <avail_date>                [1]
</availibility>
```

### Comment

We use the *comment* brick to provide free-format text. In XML, the `<comment>` element allows descriptive text intended for the specific data receiver. You shouldn't use this brick if the collected information is intended for automatic software processing:

```
<comment>Some text to describe as-
   pects of the data</comment>
```

### Data Point

We define the *data point* brick via an XML element containing numerous attributes that help describe the data value:

```
<data_point pt_code="" pt_link=""
   statistic=""typing="">2.9</data_
   point>
```

Here, the `pt_code` attribute is used for the parameter code, and the `pt_link` attribute provides a way to link data points to other content in the XML document. For example, data values might be linked to other data values or derived parameters based on that group. Additionally, we might link data values and instrumentation used to collect those data. We could collect temperature values from two different instruments but have the same parameter code (for temperature), for example. In this case, we could link the temperature values to the appropriate instruments using the pt_link attribute.

The statistic attribute lets us indicate if the data point is based on a statistical function or a discrete data-point value. The statistical attribute allows <data_point> to store both discrete values and statistical values (such as mean) using the same parameter code.

Finally, we include the typing attribute for those receiving systems that need information on the content type—strings, integers, doubles, and so on. Although this is a legacy of current technology in the data centers, it's important when the data center is actually the recipient of a data exchange.

## Data Set Identifier

The *data set identifier* brick lets us identify the data set both in a structured hierarchy familiar to those working with ocean-data sets and in a free-text format for the data provider. We describe the hierarchy structure using the level attribute, which provides an enumerated list of allowed content, including the terms cruise, station, profile, record, and related. The free-text aspect is between the <data_set_id> element's start and end tags; it lets the XML document creator specify a name for the data set. Examples could be station names, which often occur when sampling is repeated at a particular location for the generation of long-term time series:

```
 <data_set_id   level="">Station   P
    </data_set_id>
```

## Provenance

The historic source of the data contained in a data set is recorded via the *provenance* brick. Such information is valuable when we need to track information about data received. In the XML implementation, the provenance element contains assorted subelements:

```
<provenance platform_type="">
     <agency>                           [1]
     <country>                        [0,1]
     <data_grouping>                  [0,1]
     <date_created>                     [1]
     <description>                    [0,1]
     <institute_code>                 [0,1]
     <originator>                     [0,1]
     <originator_identifier>          [0,1]
     <platform_name>                  [0,1]
     <project>                        [0,1]
</provenance>
```

The <provenance> element contains two mandatory subelements: the <agency> reporting the data and the <date_created> for the data set. We might include additional information to further describe the data set, including the country and institute providing the data, the platform from which the data were collected, and whether the data are associated with a particular program, such as an international collaborative effort.

## Data Quality

The data's recognized quality is an important aspect of any data set, particularly when centralized sites provide data to clients worldwide. Clients need to make informed decisions regarding what data to include in their analyses, thus they need to know as much as possible about the data's quality.

Typically, scientists perform quality control processing on an entire data set, be it a collection of stations or an entire cruise. We use the *quality* brick to describe the results of any quality-specific testing. The only mandatory information content associated with the quality result is the parameter code, pt_code, in the XML implementation:

```
<quality justification_code=""
    pt_code="" pt_link="" reliabil-
    ity_code="" use_code="">
       <qt_date>                  [0,1]
       <tests_failed>             [0,1]
       <tests_performed>          [0,1]
</quality>
```

## Quality Tests

In contrast to the quality brick, which describes testing results, the *quality-testing* brick describes the type of testing performed. All content in this brick is mandatory, meaning that if we include the brick in the data set, we must fully describe the performed tests. This includes a description of the test, a unique identifier for the test, a test name, and the software version used in the test. Such information helps give the test traceability and repeatability:

```
<quality_testing>
       <test_description>           [1]
       <test_id>                    [1]
       <test_name>                  [1]
       <test_version>               [1]
</quality_testing>
```

## The Variable Set

We implement the *variable set* brick as a simple container for other bricks and indicate this by _set in the XML naming implementation. The variable set brick contains all the information for describing a variable including the mandatory units and variable subelements, which we describe in detail later:

```
<variable_set>
      <analysis_method>          [0,1]
      <calibration>              [0,n]
      <comment>                  [0,n]
      <instrument>               [0,1]
      <sampling>                 [0,1]
      <sensor>                   [0,1]
      <units>                      [1]
      <variable>                   [1]
</variable_set>
```

In terms of a national data center, a variable could be any parameter that's associated with an ocean observation. Typically, data centers have national mandates that emphasize particular data types, but in general they'll accept and archive a broad spectrum of data types. In this regard, the variable set brick must allow for physical, chemical, biological, or geological ocean data. Given this diversity, we can describe variables in different ways. Important information for variables includes the type of analysis, calibrations, sampling techniques, or instrumentation used to collect them.

However, the included variables could also be metadata (data about data) that accompanies the observations. Real-time data transmissions include information about the type of instrument, for example. For other data types, we know whether the sensor collected the data while ascending or descending a water column. We can include all of this in the content of defined variables, so that such information can accompany the observations themselves.

Perhaps not so obvious are the decisions made while constructing the XML hierarchies illustrated by the `<variable_set>`. In the XML implementation, `<instrument>` is a subelement of `<variable_set>`. However, instruments collect data pertaining to a particular variable. In that sense, we could argue that a variable should be a subelement of an instrument. Additionally, scientists typically apply calibrations to instruments or sensors, thus we might reasonably think of a calibration as a subelement of an instrument or sensor.

Defining the hierarchy is a nontrivial task. For this investigation, we believed that a data-centric view was more appropriate. As such, we view the data and associated variable naming as more relevant to the client. If we know the details of instrumentation and sensors, we can include this information in the XML structure, but the structure doesn't require that a variable be a product of an instrument or sensor.

In terms of calibrations, again, the structure takes a data-centric view. Although scientists perform calibrations on instrumentation, the oceanographic community is familiar with data adjustments that manipulate a data value based on other input. These "calibrations" are applied to the data stream, not the originating instrument. The structure reflects this possibility by encapsulating the `<calibration>` element within the `<variable_set>` element.

### The Location Set

We originally conceived the location-related information to specify the data set's *x*, *y*, *z*, *t* location. However, concerns over the XML elements' nonspecific nature resulted in tag-name modifications that produced names focused toward ocean data. The resulting content still describes *x*, *y*, *z*, *t*, but in terms more familiar to ocean-data users—specifically longitude, latitude, depth (or possibly pressure), and date/time, respectively.

The information content in a *location set* brick is critical to an oceanographic data set. If you know a temperature at a particular vertical place in the ocean, but not the horizontal place, the data are useless. By creating specific tag names for this location information, we hope to elevate the content's importance, but this has created problems with processing.

Such problems originate from a metadata–data dilemma. On some occasions, we consider the *x*, *y* position information data rather than metadata. For a surface-drifter track, for example, the longitude/latitude/date information is part of the data stream—in this case, the *x*, *y*, *t* data are independent variables. In a conceptual sense, no difference exists between the *x*, *y*, *t* information and the traditional data points (a temperature value, for instance). We could store the *x*, *y*, *t* information using the `<data_point>` element by specifying an appropriate `pt_code` rather than containing it in a dedicated location set brick:

```
<location_set>
      <comment>                  [0,n]
      <depth_pressure>           [0,1]
      <latitude>                 [0,1]
      <ldate>                    [0,1]
      <longitude>                [0,1]
      <quality>                  [0,1]
</location_set>                  [0,1]
```

### The History Set

The *history set* brick provides information on the

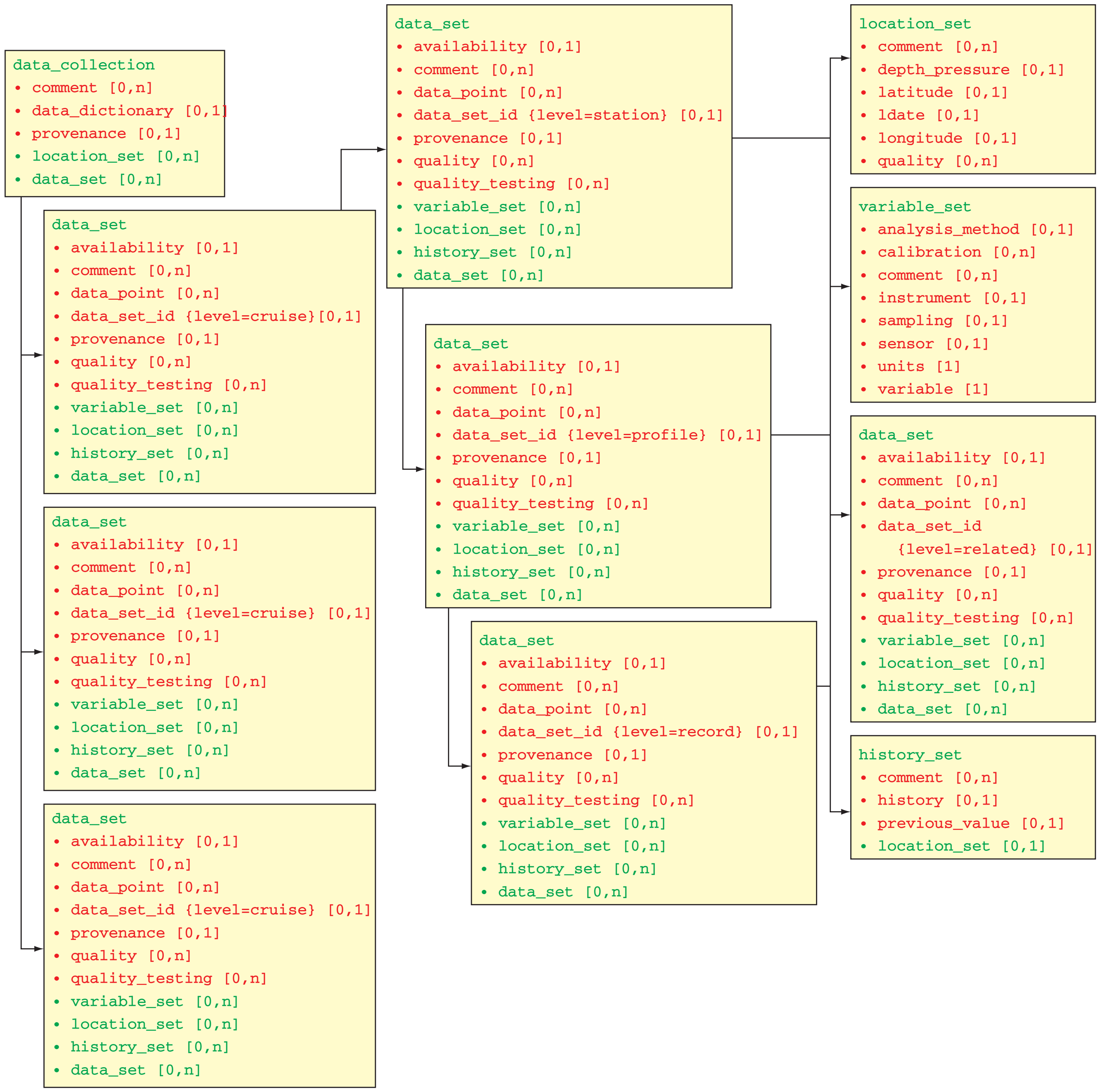

**Figure 2. Bricks representing a 1D profile. The green text indicates simple containers, whereas the red text indicates a brick.**

data set's processing history. From the data center perspective, quality control could have resulted in changes to the data set. A spike value might have been flagged as bad and removed from the provided data set, for example, but the data center system retains the original value, which we can include in the XML document via the `<history_set>` element (for example, as the `<previous_value>` element):

```
<history_set>
       <comment>                     [0,n]
       <history>                     [0,1]
       <previous_value>              [0,1]
       <location_set>                [0,1]
</history_set>
```

### Variable and Units

Two remaining bricks are sufficiently important to warrant further explanation: *variable* and *units*. Both are contained within the variable set.

The variable brick encapsulates information related to the variable. As the following XML implementation shows, variables must include a parameter code (in this case, the mandatory `pt_code` attribute) and the type of variable (for example, `string` or `integer`):

```
<variable duplicate_indicator=""
   kind="" pt_code="" pt_link=""
   typing="">
       <accuracy>                  [0,1]
       <below_detection>           [0,1]
       <decimal_places>            [0,1]
       <maximum_value>             [0,1]
       <minimum_value>             [0,1]
       <null_value>                [0,1]
       <precision>                 [0,1]
       <variable_name>               [1]
</variable>
```

The type is included for legacy purposes. A variable must also have a name, which can be an alias to the parameter code. For instance, `TEMP` might be the parameter code and Temperature the name.

Other information that could accompany the variable definition includes minimum and maximum values for the data set. Inclusion allows internal consistency checking and easily provides legacy systems with the required range. We can also specify detection limits (important for many chemical analyses) in which procedures preclude measuring values below certain thresholds. Additionally, we can include accuracy and precision estimates, as well as null values, which identify missing data in data sets. Many legacy storage systems use null values.

We describe the units brick using mandatory attributes for the parameter code and the stored units. Here, the attribute `stored_units` identifies the units used in the storage system from which we obtained the data used to create the XML document. The attribute `received_units` identifies the variable's original units—for example, scientists might record a temperature and deliver it to the data center in degrees Fahrenheit, but the database system will store it in degrees Celsius. We might also note the conversion required in the `<conversion>` element, and if required, a `<reference>` might provide details. We can also include the variable's name:

```
<units pt_code="" pt_link="" re-
   ceived_units="" stored_units="">
       <conversion>                [0,1]
       <reference>                 [0,1]
       <variable_name>             [0,1]
</units>
```

## An Oceanographic Profile XML Structure

Once defined, we can assemble the bricks, as represented in the XML structure, to correspond to the hierarchy found in ocean-data types. Consider the 1D profile—Figure 2 shows the nesting of the data set as a series of data set bricks. Each data set contains the content described in the XML snippet for `<data_set>`. The `<data_collection>` element acts as the XML document element, and also includes previously described bricks for `<comment>`, `<data_dictionary>`, `<provenance>`, and `<location_set>`.

The `<data_set>` element also indicates the importance of the `level` attribute, which is contained in the `<data_set_id>` element. At the high level, this attribute indicates a `cruise` level data set. Proceeding through the hierarchy, the level attribute indicates `station`, `profile`, and, finally, `record`.

At any of the levels, we might define the `<location_set>`, `<variable_set>`, or `<history_set>` elements. Additionally, a data set of related information (indicated by the `level=related` attribute) might be included. The related information is intended for those data collected in support of the primary data—when collecting a vertical profile of ocean water temperature, you typically collect an estimate of water depth, which would be related information. We can find another example looking at biological data profiles, in which it's sometimes important to collect light intensity or wind velocity. Again, these data support the primary data and would be related.

Our work with data exchange methods is ongoing, and we are now investigating using Geography Markup Language (GML)[6] to construct the brick information.[7] GML provides a markup standard that many commercial mapping packages are using. In this exercise, the bricks provide the foundation content that we must include in the GML description. Again, separating the content definition from the implementation method is a functional advantage.

We're currently considering applying the bricks to other ocean-data types. Our original concept examined data types from the perspective of one-, two-, three-, or four-dimensional data, not from an individual data type. We're now testing the developed profile structure using other 1D data types familiar to the ocean-data community. Additionally, a Canadian standardization project is using the bricks to define common data set content.

Our investigation has also contributed to the realization that parameter codes play a critical role in ocean-data exchange. Currently, countries have no agreement on a single standard or universal set of data codes. Electronic instrumentation for collecting ocean data is prolific, and has resulted in many laboratories defining code systems for their labs, with no inter-lab coordination. These problems exist nationally as well as internationally, with laboratories under the same government department even using differing code systems. This results in serious legacy and exchange issues. Consequently, efforts are underway both within and between countries to map and standardize code dictionaries. Again, our investigation is exploring the use of XML in these mappings.

## References

1. G.K. Thiruvathukal, "XML and Computational Science," *Computing in Science & Eng.*, vol. 6, no. 1, 2004, pp.74–80.
2. *Report of the ICES-IOC Study on the Development of Marine Data Exchange Systems Using XML*, ICES CM 2002/C:12, International Council for the Exploration of the Sea, Apr. 2002; www.ices.dk.
3. *Report of the ICES-IOC Study on the Development of Marine Data Exchange Systems Using XML*, ICES CM 2003/C:12, International Council for the Exploration of the Sea, May 2003; www.ices.dk.
4. A.W. Isenor, J.R. Keeley, and J. Linguanti, *Developing an eXtensible Markup Language (XML) Application for DFO Marine Data Exchange via the Web*, DRDC Atlantic ECR 2003-025, Defence R&D Canada—Atlantic, Mar. 2003; http://pubs.drdc-rddc.gc.ca/.
5. *XML Schema Part 2: Datatypes*, World Wide Web Consortium (W3C) Recommendation, May 2001; www.w3.org/TR/xmlschema-2/.
6. S. Cox et al., *OpenGIS Geography Markup Language (GML) Implementation Specification*, version 3.0, OGC 02-023r4, Open Geospatial Consortium, 29 Jan. 2003; www.opengeospatial.org.
7. A.W. Isenor, *Describing Generic Ocean Environmental Data Objects Using Geography Markup Language*, DRDC Atlantic TM 2004-087, Defence R&D Canada—Atlantic, July 2004; http://pubs.drdc-rddc.gc.ca/.

**Anthony W. Isenor** is a defense scientist at Defence R&D Canada—Atlantic and is cochair of the International Council for the Exploration of the Sea/Intergovernmental Oceanographic Commission's Study Group on the Development of Marine Data Exchange Systems Using XML. His research interests include data exchange structures, metadata, codes, and ontologies. Isenor has a MS in oceanography from Dalhousie University, Nova Scotia. He is a past chair of the ICES Working Group on Marine Data Management and a current member of the Canadian Meteorological and Oceanographic Society. Contact him at anthony.isenor@drdc-rddc.gc.ca.

**J. Robert Keeley** is acting director of the Marine Environmental Data Service, Canada's national ocean-data center. He is a physical scientist in charge of oceanographic systems development and is the originator of the brick concept. His research interests include development of standardized procedures for data exchange. Keeley has a MS in oceanography from Dalhousie University , Nova Scotia. He is a past cochair of the Argo Data Management Team, a member of the Ocean Observation Panel on Climate and a member of the Canadian Meteorological and Oceanographic Society. Contact him at keeley@meds-sdmm.dfo-mpo.gc.ca.