Intelligraphics
Incorporated

September 5, 2007

EDCS Usage
by
Dr. Richard F. Puk

Introduction

This report describes the reasons for utilizing the Environmental Data Coding Specification (EDCS) standardized as ISO/IEC 18025:2005 to apply semantic meaning to the content of X3D Earth worlds. It also specifies a technique for using EDCS within X3D by populating metadata nodes.

Background

For many years, the modeling and simulation community has had the requirement to assign semantic meaning to data. Prior to the standardization of the SEDRIS family of standards (of which EDCS is an integral part), the codes used to assign semantic meaning were proprietary and not portable. As part of the SEDRIS standardization effort, a scheme was developed to standardize semantic meanings in a precise manner with the intent of ensuring that definitions of these meaning had the following characteristics:

1. The meanings should be based on widely accepted terms.
2. The meanings shold be derived from the primary sources responsible for the definition of these terms with specific references to these sources cited.
3. The meanings should widely applicable to any application related to the representation of environmental data.
4. Adding additional meanings to the standard should be possible with appropriate oversight.

EDCS is the first attempt to create an International Standard that satisfies these goals.

EDCS

EDCS is divided into several categories of dictionaries of environmental concepts:

7644 Cortina Court
Carlsbad, CA 92009-8206
Phone: (760) 753-9027
FAX: (760) 753-9027

a. *Classifications* that specify the type of environmental objects;

b. *Attributes* that specify the state of environmental objects;

c. *Attribute value characteristics* that specify information concerning the values of attributes;

d. *Attribute enumerants* that specify the allowable values fo the state of an enumerated attribute;

e. *Units* that specify quantitative measures of the state of some environmental objects;

f. *Unit scales* that allow a wide range of numerical values to be stated;

g. *Unit equivalence classes* that specify sets of units that are mutually comparable;

h. *Organization schemas* that are useful for locating classifications and attributes from a common context; and

i. *Groups* into which conepts sharing a common context are collected.

Each of these categories has a dictionary. Within each dictionary are the following entries:

a. Concept definition;

b. Alphanumeric label;

c. Numeric code;

d. Reference type, reference, and supplemental references; and

e. Other dictionary-dependent information

It is intended that the numeric codes be used to identify concepts with the alphanumeric label used to mnemonically specify the numeric code. Where possible the definitions use terms already defined. In many cases, the meaning of a term is derived directly from an English language dictionary. In the EDCS, the Shorter Oxford English Dictionary is used for this purpose.

The dictionary are extended by registering new concepts (along with their definitions, labels, codes, references, and other information) with the International Registry of Items. A formal group has been established for reviewing registration proposals. Registration proposals can be submitted by SC24 national bodies, SC24 working groups, and liaison organizations having a formal relationship with SC24. This includes the Web3D Consortium. Thus, Web3D Consortium members can submit proposals either through the Web3D Consortium or through their respective national bodies.

A complete description of the EDCS concepts as well as a listing of the various dictionaries may be found in ISO/IEC 18025 available at:

http://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_IEC_18025_Ed1.html

To download a copy select the “Downloadable html version” link for ISO/IEC 18025 at:

http://standards.iso.org/ittf/PubliclyAvailableStandards/

Relationship of EDCS to X3D Earth

The X3D Earth project is involved in the access of environmental information usually in the form of geographic data. As part of this effort, the environmental information should be annotated with information describing the semantic of that information. EDCS provides the means whereby the semantic information associated with the data can be accurately specified. Since X3D Earth data is intended to be access from within X3D worlds, one means of providing this semantic information is by judicious use of the already existing X3D metadata nodes. The functionality of the X3D metadata nodes is such that the data so specified becomes part of the executing X3D world and thus is accessible during the running of the X3D world both by routing information between the fields of a metadata node and other fields in the X3D world and by reference using the Scene Access Interface (SAI). While other specifications for specifying semantic data (e.g., those provided by OGC facilities), only EDCS has been specified as an International Standard and thus will be widely supported. However, should some data use other specifications for semantic data (e.g., OGC, DGIWIG, etc.), which specification is being used can be easily discerned by the value of the reference field of the metadata node.

Summary of X3D Metadata Node Capabilities

There are five metadata nodes in X3D:

a. MetadataDouble

b. MetadataFloat

c. MetadataInteger

d. Metadata Set

e. Metadata String

Each of the metadata nodes is derived from the X3DMetadataObject abstract interface. Thus, each metadata node consists of the following four fields:

a. *metadata*—this SFNode field may reference a metadata node providing information about the metadata node in which the metadata field resides. Thus, even metadata nodes can have metadata.

b. *name*—this SFString field specifies the name of the metadata information.

c. *reference*—this SFString field specifies the name of the specification that defines the meaning of the name.

d. *value*—this field specifies the zero or more values for the metadata. Its data type depends on the specific metadata node. For example, the *value* field of a MetadataDouble node is of data type MFDouble. The MetadataSet node is unusual in that it allows multiple value of differing data types (rather equivalent to a record data type).

All nodes in X3D have a metadata field (including the metadata nodes). Thus, metadata can be specified at any level with an X3D world.

Possible Bindings of EDCS to X3D Metadata Nodes

The following are two different ways of binding the functionality of EDCS to the X3D metadata nodes. Each of these two bindings use the current X3D metadata nodes. It would be possible to extend the set of X3D metadata nodes to provide for a special MFInt32 selection field that remove be used to specify the extra codes required by EDCS for the various forms of attributes. It is expected that whichever binding is selected will be approved as an X3D Earth Recommended Practice.

Common approach for non-attribute dictionaries

The binding of EDCS dictionary entries to X3D metadata nodes is straightforward and applies for each fo the three attribute approaches and applies to the Classification, Attribute Value Characteristic, Unit Equiavlence Class, Organization Schema, and General Group dictionaries. For these dictionary entries, the following binding to a MetadataInteger node is proposed:

X3D Field	EDCS mapping	Example
metadata	As needed	NULL
name	Dictionary name	“Classification”
reference	Reference to EDCS	“ISO/IEC 18025:2005”
value	Code from dictionary	121 (code for a balloon)

The complication arises when specifying attributes since the values are not simply numbers or text strings. The following are three possible approaches to binding EDCS attributes to X3D metadata nodes.

Single node approach for EDCS attributes

In this approach, the EDCS attribute value is bound as much as possible to a single X3D metadata node.

Attributes with a numeric attribute value can be either a single value or one of eight types of interval values. Therefore, a numeric attribute value always has a numeric value type that indicates whether the attribute has a single value or is one of the various interval values. Then, depending on the numeric value type the attribute value will either be also have a single number if the the numeric value type is a single value or is one of the semi-open intervals (i.e., unlimited on one end of the interval) or two numbers if the numeric value type is a closed interval (i.e., the bounds are specified for each end of the interval).

The following is an example for a HEIGHT attribute which has a real value specified using a MetadataDouble node (all floating point values in EDCS are double-precision):

X3D Field	Number of entries	EDCS mapping	Example
metadata	1	As needed	NULL
name	1	Dictionary name	"Attribute"
reference	1	Reference to EDCS	"ISO/IEC 18025:2005"
value	1st entry in MFDouble array	Code from dictionary	443.0 (code for HEIGHT converted to real)
	2nd entry in MFDouble array	Numeric value type	1.0 (code for SINGLE_VALUE converted to real)
	3rd entry in MFDouble array	Unit code	142.0 (code for METRE converted to real)
	4th entry in MFDouble array	Unit scale code	11.0 (code for UNI converted to real)
	5th entry in MFDouble array	value	27.5 (specifying that the height is 27.5 metres.)

For attributes whose data type is INTEGER, INDEX, or COUNT, a MetadataInteger node would be used. The following is an example for the EDCS attribute HOSPITAL_BED_CAPACITY which has data type COUNT and can handle 1 to 240 patients:

X3D Field	Number of entries	EDCS mapping	Example
metadata	1	As needed	NULL
name	1	Dictionary name	“Attribute”
reference	1	Reference to EDCS	“ISO/IEC 18025:2005”
value	1st entry in MFInt32 array	Code from dictionary	469 (code for HOSPITAL_BED_CAPACITY)
	2nd entry in MFInt32 array	Numeric value type	5 (code for CLOSED_INTERVAL)
	3rd entry in MFInt32 array	Minimum value	1
	4th entry in MFInt32 array	Maximum value	240

For attributes whose data type is ENUMERATED, a MetadataInteger node would be used. The following is an example for the EDCS attribute THUNDERSTORM_INTENSITY indicating a thunderstorm intensity of MODERATE:

X3D Field	Number of entries	EDCS mapping	Example
metadata	1	As needed	NULL
name	1	Dictionary name	“Attribute”
reference	1	Reference to EDCS	“ISO/IEC 18025:2005”
value	1st entry in MFInt32 array	Code from dictionary	1433 (code for THUNDERSTORM_INTENSITY)
	2nd entry in MFInt32 array	value	2 (code for MODERATE)

Attributes with a STRING, KEY, or CONSTRAINED_STRING attribute value consists of a locale specification and an accompanying string. The locale specification consists of a two-character language code and a three-character country code. A MetadataString node would be used. The following is an example of the EDCS attribute ROUTE_DESIGNATION for route A25:

X3D Field	Number of entries	EDCS mapping	Example

metadata	1	As needed	NULL
name	1	Dictionary name	"Attribute"
reference	1	Reference to EDCS	"ISO/IEC 18025:2005"
value	1st entry in MFString array	Code from dictionary	"1145" (code for ROUTE_DESIGNATION converted to text)
	2nd entry in MFString array	locale	"ENUSA" (two character code for English followed by three character code for United States of America)
	3rd entry in MFString array	value	"A25"

For all EDCS attributes, if an entry from the EDCS Value Characteristics dictionary is required, those entries will be bound inserting a MetadataInteger node in the metadata field of whatever metadata node is being used to specify the attribute value. That additional metadata node will have the form described in *Common approach for non-attribute dictionaries*.

Structured metadata approach

In this approach, an EDCS Attribute is mapped to a MetadataSet node. The value field of a MetadataSet node is a set of X3D metadata nodes.

Attributes with a numeric attribute value can be either a single value or one of eight types of interval values. Therefore, a numeric attribute value always has a numeric value type that indicates whether the attribute has a single value or is one of the various interval values. Then, depending on the numeric value type the attribute value will either be also have a single number if the the numeric value type is a single value or is one of the semi-open intervals (i.e., unlimited on one end of the interval) or two numbers if the numeric value type is a closed interval (i.e., the bounds are specified for each end of the interval).

The following is an example for a HEIGHT attribute which has a real value:

X3D Field	Number of entries	EDCS mapping	Example
metadata	1	As needed	NULL
name	1	Dictionary name	"Attribute"

reference	1	Reference to EDCS	"ISO/IEC 18025:2005"
value	1st entry of value field 1st metadata node (a MetadataInteger node) in MFNode array	Code from dictionary	443 (code for HEIGHT)
	2nd entry of value field of 1st metadata node (a MetadataInteger node) in MFNode array	Numeric value type	1 (code for SINGLE_VALUE)
	value field of 2nd metadata node (a MetadataDouble node) in MFNode array metadata field of 2nd metadata node is a MetadataInteger node specifying the unit and unit scale for the value	value Unit code and unit scale code	27.5 (specifying that the height is 27.5 metres.) 142 (code for METRE) in 1st element of MFInt32 array and 11 (code for UNI) in 2nd element of MFInt32 array.

For attributes whose data type is INTEGER, INDEX, or COUNT, a MetadataInteger node would be used. The following is an example for the EDCS attribute HOSPITAL_BED_CAPACITY which has data type COUNT and can handle 1 to 240 patients:

X3D Field	Number of entries	EDCS mapping	Example
metadata	1	As needed	NULL
name	1	Dictionary name	"Attribute"
reference	1	Reference to EDCS	"ISO/IEC 18025:2005"

value	1st entry of value field of 1st metadata node (a MetadataInteger node) in MFNode array	Code from dictionary	469 (code for HOSPITAL_BED_CAPACITY)
	2nd entry of value field of 1st metadata node (a MetadataInteger node) in MFNode array	Numeric value type	5 (code for CLOSED_INTERVAL)
	1st entry of value field of 2nd metadata node (a MetadataInteger node) in MFNode array	Minimum value	1
	2nd entry of value field of 2nd metadata node (a MetadataInteger node) in MFNode array	Maximum value	240

For attributes whose data type is ENUMERATED, two MetadataInteger nodes would be used to populate the MFNode field of the MetadataSet node. The following is an example for the EDCS attribute THUNDERSTORM_INTENSITY indicating a thunderstorm intensity of MODERATE:

X3D Field	Number of entries	EDCS mapping	Example
metadata	1	As needed	NULL
name	1	Dictionary name	“Attribute”
reference	1	Reference to EDCS	“ISO/IEC 18025:2005”

value	1st entry of value field of 1st metadata node (a MetadataInteger node) in MFNode array	Code from dictionary	1433 (code for THUNDERSTORM_INTENSITY)
	1st entry of value field of 2nd metadata node (a MetadataInteger node) in MFNode array	value	2 (code for MODERATE)

Attributes with a STRING, KEY, or CONSTRAINED_STRING attribute value consists of a locale specification and an accompanying string. The locale specification consists of a two-character language code and a three-character country code. A MetadataString node would be used. The following is an example of the EDCS attribute ROUTE_DESIGNATION for route A25:

X3D Field	Number of entries	EDCS mapping	Example
metadata	1	As needed	NULL
name	1	Dictionary name	“Attribute”
reference	1	Reference to EDCS	“ISO/IEC 18025:2005”
value	1st entry of value field of 1st metadata node (a MetadataInteger node) in MFNode array	Code from dictionary	1145 (code for ROUTE_DESIGNATION)
	1st entry of value field of 2nd metadata node (a MetadataString node) in MFNode array	locale	“ENUSA” (two character code for English followed by three character code for United States of America)

	2nd entry of value field of 2nd metadata node (a MetadataString node) in MFNode array	value	“A25”

For all EDCS attributes, if an entry from the EDCS Value Characteristics dictionary is required, those entries will be bound inserting a MetadataInteger node in the metadata field of the MetadataSet node. That additional metadata node will have the form described in *Common approach for non-attribute dictionaries*.

X3D selected metadata approach

This approach would require adding a *selection* field of data type MFInt32 to each of the existing metadata nodes. If this field is not specified, the various X3D metadata nodes would have the current definition. Of course, the interpretation of this additional field would have values that depend on the particular metadata specification identified in the *reference* field. When use for specifying EDCS attributes, the 1st entry in this *selection* field would be used to specify the attribute code (and, for real valued attributes, the numeric value type).

This approach has the distinct advantage of being a single level metadata approach but would have the consistency of the structured metadata approach. The one open question is whether the Unit and Unit Scale codes for the real valued attributes are specified in the metadata field of the MetadataDouble node or are specified as the 3rd and 4th entries of the *selection* field. Specifing these values in the *metadata* field would require that an attribute value characteristic value also be allowed in the MetadataInteger node that would populate the *metadata* field. Specifying these values in the *selection* field might be considered confusing since they have nothing to do with selection and really are metadata about the metadata.

The following is an example for a HEIGHT attribute which has a real value specified using a MetadataDouble node (all floating point values in EDCS are double-precision):

X3D Field	Number of entries	EDCS mapping	Example
metadata	1	As needed	NULL
name	1	Dictionary name	“Attribute”
reference	1	Reference to EDCS	“ISO/IEC 18025:2005”
selection	1st entry in MFInt32 array	Code from dictionary	443 (code for HEIGHT)

	2nd entry in MFInt32 array	Numeric value type	1 (code for SINGLE_VALUE)
	3rd entry in MFInt32 array	Unit code	142 (code for METRE)
	4th entry in MFInt32 array	Unit scale code	11 (code for UNI)
value	1st entry in MFDouble array	value	27.5 (specifying that the height is 27.5 metres.)

For attributes whose data type is INTEGER, INDEX, or COUNT, a MetadataInteger node would be used. The following is an example for the EDCS attribute HOSPITAL_BED_CAPACITY which has data type COUNT and can handle 1 to 240 patients:

X3D Field	Number of entries	EDCS mapping	Example
metadata	1	As needed	NULL
name	1	Dictionary name	“Attribute”
reference	1	Reference to EDCS	“ISO/IEC 18025:2005”
selection	1st entry in MFInt32 array	Code from dictionary	469 (code for HOSPITAL_BED_CAPACITY)
	2nd entry in MFInt32 array	Numeric value type	5 (code for CLOSED_INTERVAL)
value	1st entry in MFInt32 array	Minimum value	1
	2nd entry in MFInt32 array	Maximum value	240

For attributes whose data type is ENUMERATED, a MetadataInteger node would be used. The following is an example for the EDCS attribute THUNDERSTORM_INTENSITY indicating a thunderstorm intensity of MODERATE:

X3D Field	Number of entries	EDCS mapping	Example
metadata	1	As needed	NULL

name	1	Dictionary name	“Attribute”
reference	1	Reference to EDCS	“ISO/IEC 18025:2005”
selection	1st entry in MFInt32 array	Code from dictionary	1433 (code for THUNDERSTORM_INTENSITY)
value	1st entry in MFInt32 array	value	2 (code for MODERATE)

Attributes with a STRING, KEY, or CONSTRAINED_STRING attribute value consists of a locale specification and an accompanying string. The locale specification consists of a two-character language code and a three-character country code. A MetadataString node would be used. The following is an example of the EDCS attribute ROUTE_DESIGNATION for route A25:

X3D Field	Number of entries	EDCS mapping	Example
metadata	1	As needed	NULL
name	1	Dictionary name	“Attribute”
reference	1	Reference to EDCS	“ISO/IEC 18025:2005”
selection	1st entry in MFInt32 array	Code from dictionary	1145 (code for ROUTE_DESIGNATION)
value	1st entry in MFString array	locale	“ENUSA” (two character code for English followed by three character code for United States of America)
	2nd entry in MFString array	value	“A25”

Summary

Each of the approaches cited above has advantages and disadvantages. However this should provide sufficient context for further discussion. Note that if the proposal to add metadata nodes for each of the field types is adopted, only the EDCS Boolean attribute value type would be affected but, in each approach, the modification would be straightforward.

THIS PAGE INTENTIONALLY LEFT BLANK