Committee Draft ISO/IEC CD	
Date: **2005-12-28**	Reference number: ISO/JTC 1/SC 32**N1388**
Supersedes document SC 32N1102	

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES.

ISO/IEC JTC 1/SC 32 Data Management and Interchange Secretariat: USA (ANSI)	Circulated to P- and O-members, and to technical committees and organizations in liaison for voting (P-members only) by: **2006-03-27** Please return all votes and comments in electronic form directly to the SC 32 Secretariat by the due date indicated.

ISO/IEC CD 20944-01:200x(E)

Title: Information technology - Metadata Registry Interoperability & Bindings (MDR-IB) Part 1:Framework

Project: 1.32.17.01.01.00

Introductory note: The attached document is hereby submitted for a three-month letter ballot to the National Bodies of ISO/IEC JTC 1/SC 32. The ballot starts 2005-12-28.

Medium: E

No. of pages: 17

Address Reply to: SC 32 Secretary, ISO/IEC JTC 1/SC 32, Farance Inc, Island Box 256, New York, NY 10044-0205, United States of America
Telephone: +1 212 486-4700; E-mail: SC32-Sec@JTC1SC32.org

Reference number of working document: **ISO/IEC JTC1 SC32 N1388**

Date: 2005-12-25

Reference number of document: **ISO/IEC CD2 20944-01**
[Release Sequence #8]

Committee identification: ISO/IEC JTC1 SC32 WG2

SC32 Secretariat: US

Information technology — Metadata Registries Interoperability and Bindings (MDRIB) — Part 01: Framework

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: International standard
Document subtype: if applicable
Document stage: (30) Committee
Document language: E

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 20944-01 was prepared by Technical Committee ISO/IEC JTC1, *Information Technology*, Subcommittee SC32, *Data Management and Interchange*.

ISO/IEC 20944 consists of the following parts, under the general title *Information technology — Metadata Registries Interoperability and Bindings (MDRIB)*:

— *Part 01: Framework*

— *Part 02: Common vocabulary*

— *Part 03: Common provisions for conformance*

— *Part 04: Generic usage*

— *Part 05: Common data structures and services*

— *Part 06: Semi-structured aggregation*

— *Part 20: Common provisions for coding bindings*

— *Part 21: XML coding binding*

— *Part 22: DIVP coding binding*

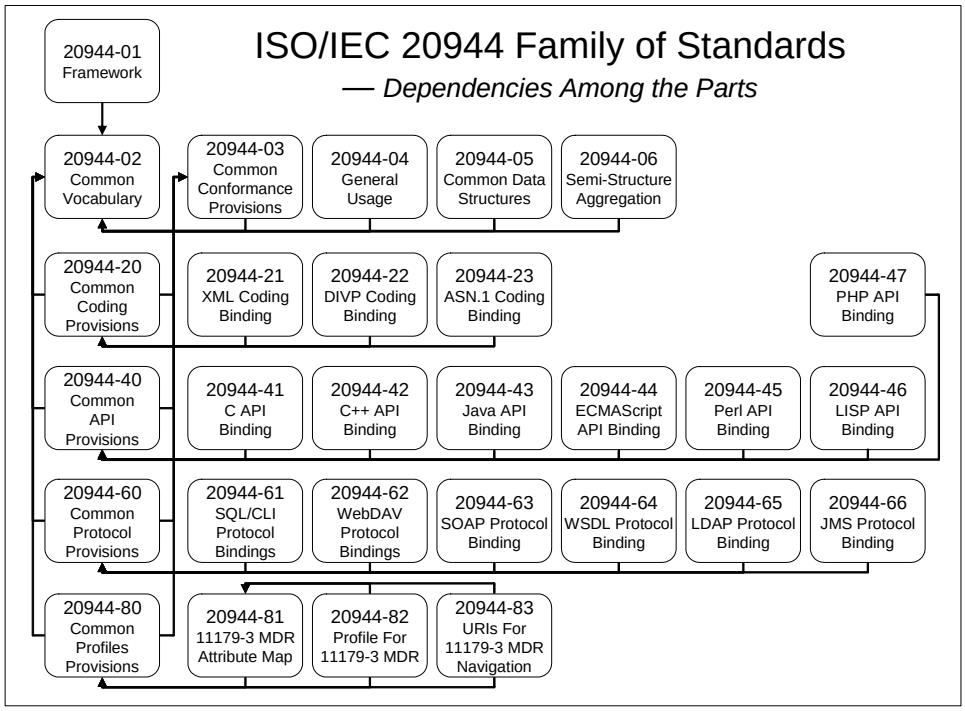
— *Part 23: ASN.1 coding binding*

— *Part 40: Common provisions for application programming interface (API) bindings*

- Part 41: C API binding
- Part 42: C++ API binding
- Part 43: Java API binding
- Part 44: ECMAScript API binding
- Part 45: Perl API binding
- Part 46: LISP API binding
- Part 47: PHP API binding
- Part 60: Common provisions for protocol bindings
- Part 61: SQL/CLI protocol binding
- Part 62: WebDAV protocol binding
- Part 63: SOAP protocol binding
- Part 64: WSDL protocol binding
- Part 65: LDAP protocol binding
- Part 66: JMS protocol binding
- Part 80: Common provisions for profiles
- Part 81: Attribute mapping for 11179-3 MDR metamodel
- Part 82: Profile for 11179-3 MDR metamodel
- Part 83: URI suffixes for 11179-3 MDR metamodel navigation

Introduction

This Part of ISO/IEC 20944 is an overview and framework of the structure and use of the 20944 family of standards. The following diagram shows the organization of the ISO/IEC 20944 family of standards.



Organization of ISO/IEC 20944 family of standards.

This Part provides information about the other parts; this Part does not make requirements upon implementations.

Information technology — Metadata Registries Interoperability and Bindings (MDRIB) — Part 01: Framework

Editor's Note: Each part of 20944 is marked with a common sequence number ("**[Release Sequence #N]**") to indicate they are synchronized and harmonized among themselves. The mark "**[Release Sequence #N]**" does *not* imply that there are a complete set of N-1 prior drafts for any particular Part.

1 Scope

The ISO/IEC 20944 family of standards describe codings, APIs, and protocols for interacting with an ISO/IEC 11179 metadata registry (MDR).

This Part provides the overview and framework of the ISO/IEC 20944 family of standards.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 20944-02:—[1], *Information technology — Metadata Registries Interoperability and Bindings (MDRIB) — Part 02: Common vocabulary*

NOTE The 20944-02 document includes terminology from a collection of several additional standards and specifications.

3 Terms and definitions

For the purposes of this document, the terms and definitions given in Part 2 and the following apply[2].

[1] To be published. The international standards, technical reports, and drafts of the 11179, 20943, and 20944 series are available at

http://metadata-standards.org/11179
http://metadata-standards.org/20943
http://metadata-standards.org/20944

3.1
11179
abbreviation for "the ISO/IEC 11179 family of standards" (Information technology — Metadata Registries (MDR))

NOTE As of the time of publication of this International Standard, ISO/IEC 11179 consists of 6 parts.

3.2
11179-3
abbreviation for "the ISO/IEC 11179-3 standard" (Information technology — Metadata Registries (MDR), Part 3: Registry metamodel and basic attributes)

3.3
20943
abbreviation for "the ISO/IEC 20943 family of standards" (Information technology — Procedures for achieving metadata registry content consistency)

NOTE As of the time of publication of this International Standard, ISO/IEC 20943 consists of 2 parts.

3.4
20944
abbreviation for "the ISO/IEC 20944 family of standards" (Information technology — Metadata Interoperability and Bindings (MDIB))

NOTE See the Foreword and the Introduction for a list of parts in this International Standard.

4 Metadata vs. data

Metadata is descriptive data about objects[3]. An *essential characteristic* of metadata is that it is *descriptive data* and that it is *descriptive about something*. For example, if ***P*** is data and ***P***→***Q*** represents the descriptive relationship such that ***P*** describes ***Q***, then ***P*** is metadata about ***Q***. If there is no relationship to ***Q***, then ***P*** is no longer metadata (i.e., ***P*** is merely data) because metadata is always relative to the object of description. Or stated differently, ***P*** only becomes metadata once its descriptive relationship to ***Q*** is established. Thus, it is *impossible to determine independent of context and relationships* that any piece of data is actually also metadata. The implication is: metadata interchange is no different than data interchange. The 20944 family of standards simply treat everything as data — whether it is *used as metadata* is outside the scope of this International Standard.

NOTE Parts 81 and 82 of this International Standard provide a mapping and a profile such that 20944 bindings may be used to interchange metadata contained in 11179 metadata registries, e.g., an application may connect to, access, read, and use metadata from an 11179 metadata registry. Part 102 explains that the 11179 concept *metadata item attribute* is treated as 20944's concept of a *data element*. The 11179 metamodel also has the concept of a *data element*, but the 11179 data element serves a different in purpose than the 20944 data element.

---

[2] Users and implementers of this International Standard may find it useful to reference additional terms and definitions from 20944-02.

[3] For example, metadata may be descriptive data about other data.

5 Achieving metadata and data interoperability and harmonization

5.1 Concepts

Interoperability with a metadata registry can be achieved in various ways. This family of standard provides a framework within which many approaches can be standardized.

All interoperability requires some kind of interface, and associated bindings, between two or more participating functional units. A binding provides a concrete mapping of a functional unit to an interface.

Three categories of bindings are supported by this family of standards:

— codings, which deal with the formalized representation of information;
— APIs, which specify a binding in programming terms;
— protocols, which specify formalized communications.

5.2 Methodology

As used for many complex engineering tasks, the use of engineering methodologies can simplify the tasks and can reduce the business and engineering risk. In the context of increasing metadata and data interoperability harmonization, three methodologies have been employed to simplify the tasks and to reduce risk.

The first methodology employed is: treating data (and metadata) interoperability as series of layered technical specifications (e.g., standards), from application-independent layers to application-specific layer(s).

The second methodology employed is: the simplification of interoperability specializations, also known as *bindings*. Rather than independently developing each separate method of representation and access (*codings*, *APIs*, *protocols*), a common, harmonized approach is taken where each binding is derived in a consistent two-step process: step #1 is choosing from the categories of coding, API, protocol (or combination)[4], which themselves are derived from a common data model and navigation method[5]; step #2 is to derive the specific binding from its general binding, e.g., the XML coding binding (Part 21) and other (specific) coding bindings are derived from the generic coding binding Part 020; the C API binding (Part 41), the ECMAScript API binding (Part 44), and the other API bindings are derived from the generic API binding Part 40. Because these bindings have a well-defined derivation, the bindings are harmonized, i.e., there is commonality in meaning and interpretation across the bindings. Thus, the complexity of adding and harmonizing a new (coding, API, protocol) binding is greatly simplified.

The third methodology employed is: the use of rule-based bindings to simplify the normative wording of the standards. A rule-based binding is binding that is specified by a general set of rules (in contrast to

---

[4] The identification of the specialization categories (codings, APIs, and protocols) is not intended to be exhaustive.

[5] A *common data model and navigation method* means: IF, for example, a data model has an object **X** that contains three components **J**, **K**, and **L**, and the component **J** itself has two components **P** and **Q**, THEN the corresponding coding binding will include a data object **X**, that contains three components **J**, **K**, **L** and **J**'s subcomponents **P** and **Q**; likewise the API bindings and protocol bindings would have a similar structure: depending upon the binding, there is some **`getvalue`** service that gets the value in a single navigation step (e.g., in one step **`getvalue("X.J.P")`** or in successive steps (e.g., **`putpath("X")`**, **`putpath("J")`**, **`getvalue("P")`**).

application-specific normative wording). For example, the XML coding binding is based upon a set of transformation rules (in contrast to specifying a specific DTD or XML schema).

5.3 Metadata and data interoperability

The successful interchange of data is dependent upon mutual agreement of interchange participants. Some key requirements for successful data interchange include (from lower implementation details to higher level abstractions):

— The *syntax* determines how data is coded (structured) and encoded (represented). Codings include specifications for organizing data structures (e.g., *How are records represented? Is tagging embedded or implied?*). Encodings include specifications for representation of datatypes (e.g., are numbers represented as a string of characters or a string of bits?). Example: *"the temperature is 17"* might be coded/encoded in several ways: (1) in XML, it might be coded as a tagged element **`"<temp>17</temp>"`** that is encoded as 15 characters in UTF-8, the encoding would be the ordering of the bits within the octet (e.g., little endian vs. big endian); (2) in C, it might be coded as a single binary octet **`{ uint8_t temp = 17; }`**.

— The *semantics* define the meaning of the data. Several kinds of descriptive techniques are possible, such as using 11179-3 for describing data. Additional technical specifications, such as standards, may be used in conjunction with the 11179-3 description of data. For example, the statement *"the temperature is 17°"* might not be descriptive enough because (1) it does not convey *units of measure*, e.g., *Celsius* or *Fahrenheit*; and (2) it does not convey what is being measured (e.g., *temperature sensor #289*). Both these features are part of the semantic description that comprises a 11179-3 Data Element.

— Application-specific behavior is determined by the context of the data. Example: The statement *"temperature is 17°C at sensor #289"* may have different meanings depending upon the application. In a telemetry application, the statement *"temperature is 17°C at sensor #289"* might represent data to be recorded and analyzed, such as updating low, average, and high values in a set of time-series data. In contrast, an HVAC (heating, ventilation, and air conditioning) application, the statement *"temperature is 17°C at sensor #289"* might represent a signal that causes heating units to turn on automatically.

Of the three issues above, 20944 concerns itself with the syntax, i.e., the bindings (codings, APIs, and protocols) for data interchange.

Regarding the semantics, 11179 is a primary tool for specifying semantics, via descriptive data, for data interchange.[6] This descriptive data is known as metadata. The descriptive data (metadata) may also be interchanged via 20944. However, in this case 20944 is being used for a different purpose: rather than data interchange, 20944 is being used for descriptive data interchange (i.e., metadata interchange). It is possible to have separate data and metadata interchanges, and to use 20944 independently for each interchange.

Neither 20944 nor 11179 specify application-specific requirements and functionality.

[6] Additional semantics may be described by or supplanted with unstructured descriptive text.

6 Organization of the multi-part standard

6.1 Overview

The ISO/IEC 20944 International Standard is organized into the following major areas. Parts 01-19 are general in nature. Coding bindings are specified Parts 20-39. API bindings are specified in Parts 40-59, and protocol bindings are specified in Parts 60-79. Parts 88-99 are standards profiles (e.g., bundles of standards or specifications). Additional parts and categories may be added in the future.

6.2 General features

Parts 01-19 concern general features that are independent of a particular binding.

- **Part 01, Framework:** a summary of the 20944 family of standards.
- **Part 02, Common vocabulary:** the common terminology for all parts of this International Standard. Part 02 is included, via normative reference, in all other Parts.[7]
- **Part 03, Common provisions for conformance:** the common conformance framework for all bindings. As a standalone document, Part 03 may be used as a reference for provisions on conforming and strictly conforming implementations. Via normative referencing, this conformance wording may be combined with Parts 04, 05, 20, 40, 60, and 80.
- **Part 04, Generic usage:** a template for using the 20944 series of standards for generic data interoperability bindings for applications outside the 11179-3 metamodel. Part 04 is intended for normative referencing by other standards and specifications that want to make use of the 20944 features and services.
- **Part 05, Common data structures and services:** common datatypes, data structures, and services that are referenced and reused in the 20944 family of standards. Part 05 may be used independently of other Parts.
- **Part 06, Semi-structured aggregation:** the data structures for ad hoc combination and aggregation of semi-structured data.

NOTE Parts may be added in the future.

6.3 Coding bindings

Parts 20-39 concern coding bindings, both general (Part 20) and specific (other parts).

- **Part 20, Common provisions for coding bindings:** the framework of data structures that is common across all 20944 coding bindings.

The following parts are specific coding bindings based upon Part 20.

- **Part 21, XML coding binding:** a rule-based mapping of data to/from the syntax of XML (extensible markup language).
- **Part 22, DIVP coding binding:** a rule-based mapping of data to/from the syntax of DIVP (dotted identifier-value pairs).
- **Part 23, ASN.1 coding binding:** a rule-based mapping of data to/from the syntax of ASN.1 (abstract syntax notation one).

---

[7] In other words, Part 02 is included, normatively, in all other parts of this International Standard.

NOTE Parts may be added in the future.

6.4 API bindings

Parts 40-59 concern API bindings, both general (Part 40) and specific (other parts).

— **Part 40, Common provisions for application programming interface (API) bindings:** the framework of access services for 20944 API bindings.

The following parts are specific API bindings based upon Part 40.

— **Part 41, C API binding:** a mapping to the ISO C programming language.
— **Part 42, C++ API binding:** a mapping to the ISO C++ programming language.
— **Part 43, Java API binding:** a mapping to the Java programming language.
— **Part 44, ECMAScript API binding:** a mapping to the ECMAScript (JavaScript) programming language.
— **Part 45, Perl API binding:** a mapping to the Perl programming language.
— **Part 46, LISP API binding:** a mapping to the ISO ISLISP programming language.
— **Part 47, PHP API binding:** a mapping to the PHP programming language.

NOTE Parts may be added in the future.

6.5 Protocol bindings

Parts 60-79 concern protocol bindings, both general (Part 60) and specific (other Parts).

— **Part 60, Common Provisions for Protocol Bindings:** the framework of common protocols and services for 20944 protocol bindings.

The following parts are specific protocol bindings based upon Part 60.

— **Part 61, SQL/CLI protocol binding:** a mapping to the SQL CLI (call level interface) protocol.
— **Part 62, WebDAV protocol binding:** a mapping to WebDAV (Web-based distributed authoring and versioning) protocol and services.
— **Part 63, SOAP protocol binding:** a mapping to the SOAP (simple object access protocol) protocol.
— **Part 64, WSDL protocol binding:** a mapping to the WSDL (web services description language) services.
— **Part 65, LDAP protocol binding:** a mapping to the LDAP (lightweight directory access protocol) protocol and services.
— **Part 66, JMS protocol binding:** a mapping to the JMS (Java message services) services.

NOTE Parts may be added in the future.

6.6 Profiles

Parts 80-99 concern international standardized profiles (ISPs).

— **Part 80, Common provisions for profiles:** this Part provides provisions that are common to International Standardized Profiles, as described by ISO/IEC TR 10000-1.

The following parts are specific profiles.

- **Part 81, Attribute mapping for metadata registries:** the mapping of the ISO/IEC 11179-3 metamodel to specific navigation identifiers.
- **Part 82, Profile for metadata registries:** the bundle of standards for representing metadata or accessing metadata in a 11179-3 metadata registry.
- **Part 83, Uniform resource identifier (URI) suffixes for 11179-3 metadata registry metamodel Navigation:** a profile of IETF RFC 2396 (Uniform Resource Identifier) and 11179-3 metamodel attributes.

NOTE Parts may be added in the future.

6.7 Harmonization of parts

Each of the parts are intended to be harmonized within each category. For example, the XML coding binding is intended to be harmonized with the ASN.1 coding binding; the C++ API binding is intended to be harmonized with the Java API binding; etc..

7 Using the 20944 family as building blocks for metadata and data interchange

7.1 General

An 11179 metadata registry is created and populated with metadata for some application, such as data elements for the transportation industry, terminology for the healthcare industry, and code sets for the aviation industry. For one IT application, programmers read portions of the metadata registry so that they can provide real-time, up-to-date metadata (e.g., a user needs to know the meaning of permissible values of a value domain). For another IT application, data stewards are able to update and clone metadata from other metadata registries.

The following illustrations show possible uses of the 20944 family of standards to support data and metadata interoperability. *Many scenarios are possible, these illustrations show some common uses*.

7.2 Illustration 1: Accessing the content of an 11179 metadata registry

An 11179 metadata registry is created and populated with metadata for some application, such as data elements for the transportation industry, terminology for the healthcare industry, and code sets for the aviation industry. For one IT application, programmers read portions of the metadata registry so that they can provide real-time, up-to-date metadata (e.g., a user needs to know the meaning of permissible values of a value domain). For another IT application, data stewards are able to update and clone metadata from other metadata registries.

In this illustration, both applications make use of one of the 20944 API bindings to access the metadata registry (a Java application would use the Java API binding; a PHP application would use the PHP API binding).

To achieve this scenario, the following parts of 20944 might be referenced:

- Part 041: C API binding
- Part 102: Profile for 11179-3 metadata registry metamodel

In other words, the application *need only reference*[8] *these two parts*. Implicitly, the following parts are normative referenced by this scenario:

- Part 002: Common vocabulary
- Part 003: Common provisions for conformance
- Part 005: Common data structures and services
- Part 040: Common provisions for application programming interface (API) bindings
- Part 100: Common provisions for profiles
- Part 101: Attribute mapping for 11179-3 metadata registry metamodel

Based upon Parts 041 and 102, the following is an excerpt from an application whose purpose is to dynamically generate a pull-down list in an HTML page, based upon the current permissible values of a value domain in a 11179-3 metadata registry:

```
mdrib_handle repository_handle;      // handle to metadata registry
mdrib_handle session_handle;         // handle to value domain
mdrib_handle node_handle;            // handle for cloned session to navigate permissible value
int index;                          // index to array of permissible values
char permissible_value_node[500];   // for navigation within repository
char value_string[500];             // value part of value-meaning pair
char value_meaning_string[50000];   // meaning part of value-meaning pair

// connect to metadata registry
repository_handle = mdib_connect(
                "//nyc.gov/mdr_repository",
                "access_type=readonly"
                );

// establish session starting at value domain
session_handle = mdib_open(
                repository_handle,
                "2.3.56789.0.2", // object identifier for value domain
                ""
                );

// begin HTML select list <select ...>
printf("<select size=1 name==\"%s\">\n",
        "nyc_borough_list"
        );

// initialize counter for indexing (walking) the value-meaning
// pairs of the value domain
index = 0;
for ( ; ; )
{
        // create navigation string to retrieved the Nth (index)
        // value-meaning pair, e.g., the first value-meaning pair
        // is pointed to by "permissible_value/__index_0", the second
        // value-meaning pair is "permissible_value/__index_1", etc.
        sprintf(permissible_value_node,
                "permissible_value/__index_%d",
                index
                );
```

[8] Technical specifications may be referenced for different purposes, such as specifying requirements (e.g., "Applications shall conform to ISO/IEC 20944-041 and ISO/IEC 20944-102") and a supplier's declaration of conformity to requirements (e.g., "Application X claims conformity to ISO/IEC 20944-041 and ISO/IEC 20944-102").

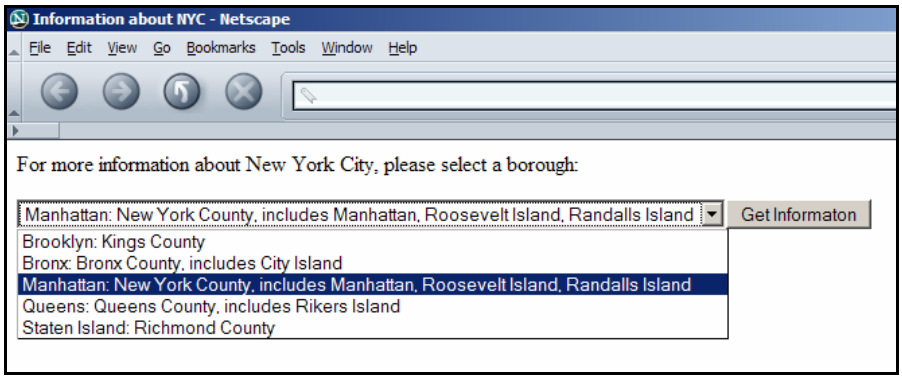
```
        // clone session with starting point at: permissible_value/__index_nnn
        node_handle = mdib_open(
                session_handle,
                permissible_value_node,
                ""
                );
        if ( node_handle == NULL )
        {
                // gone past last permissible_value
                break;
        }

        // get "value" portion of value-meaning pair
        mdib_get_value_as_str8(
                value_string,
                sizeof(value_string),
                node_handle,
                "permissible_value_has_value_relation/"
                        "value_item",
                ""
                );
        // get "meaning" portion of value-meaning pair
        mdib_get_value_as_str8(
                value_meaning_string,
                sizeof(value_meaning_string),
                node_handle,
                "permissible_value_has_value_meaning_relation/"
                        "value_meaning_description",
                ""
                );

        // generate HTML for select item: <option value="...">...</option>
        // Example: <option value="Brooklyn">Brooklyn: Kings County</option>
        printf("<option value=\"%s\">%s: %s</option>",
                value_string,
                value_string,
                value_meaining_string
                );

        // close the cloned session
        mdib_close(
                node_handle
                );
}
```

Assuming the above code extracted a value domain containing the boroughs[9] of New York City, it might product the following HTML excerpt:

```
<select name="nyc_borough_list">
<option value="Brooklyn">Brooklyn: Kings County</option>
<option value="Bronx">Bronx: Bronx County, includes City Island</option>
<option value="Manhattan">Manhattan: New York County,
        includes Manhattan, Roosevelt Island, Randalls Island</option>
<option value="Queens">Queens: Queens County, includes Rikers Island</option>
<option value="Staten Island">Staten Island: Richmond County</option>
</select>
```

This HTML code might produce the following web page:

[9] A *borough* is a geographic subdivision particular to New York City.



7.3 Illustration 2: Providing interoperable XML and ASN.1 bindings for some data model

In this scenario, an application has described its data model via UML (for object-oriented analysis and design) and via ISO/IEC 11404 (for binding-independent datatypes). This application needs to represent its data both in XML and in ASN.1 without designing an application-specific binding for XML and for ASN.1. Assuming that the application's data model is described in "Specification XYZ", for its XML coding binding it might normatively reference:

— Specification XYZ
— Part 021: XML coding binding

while its ASN.1 coding binding might normatively reference:

— Specification XYZ
— Part 023: ASN.1 coding binding

With this kind of specification, it is possible to automatically convert and exchange data to/from the XML coding binding and the ASN.1 coding binding.

[Editor's note: Example to be provided]