INTERNATIONAL STANDARD

ISO/IEC 23001-4

First edition
2009-12-15

Information technology — MPEG systems technologies —

Part 4:
Codec configuration representation

Technologies de l'information — Technologies des systèmes MPEG —

Partie 4: Représentation de configuration codec

iTeh STANDARD PREVIEW

(standards.iteh.ai)

ISO/IEC 23001-4:2009
https://standards.iteh.ai/catalog/standards/sist/771dca06-7543-4c2f-a797-
371ce9572d77/iso-iec-23001-4-2009

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 23001-4:2009
https://standards.iteh.ai/catalog/standards/sist/771dca06-7543-4c2f-a797-
371ce9572d77/iso-iec-23001-4-2009

COPYRIGHT PROTECTED DOCUMENT

Contents

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 23001-4:2009
https://standards.iteh.ai/catalog/standards/sist/771dca06-7543-4c2f-a797-
371ce9572d77/iso-iec-23001-4-2009

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national 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 and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 23001-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 23001 consists of the following parts, under the general title *Information technology — MPEG systems technologies*:

— *Part 1: Binary MPEG format for XML*

— *Part 2: Fragment request units*

— *Part 3: XML IPMP messages*

— *Part 4: Codec configuration representation*

— *Part 5: Bitstream Syntax Description Language (BSDL)*

Introduction

This part of ISO/IEC 23001 defines the methods capable of describing codec configurations in the so-called reconfigurable video-coding (RVC) framework. The objective of RVC is to offer a framework that is capable of configuring and specifying video codecs as a collection of "higher level" modules by using video coding tools. The video coding tools are defined in video tool libraries. This part of ISO/IEC 23001 defines the MPEG video tool library. The RVC framework principle could also support non-MPEG tool libraries, provided that their developers have taken care to obey the appropriate rules of operation.

For the purpose of framework deployment, an appropriate description is needed to describe configurations of decoders composed of or instantiated from a subset of video tools from either one or more libraries. As illustrated in Figure 1, the configuration information consists of

— bitstream syntax description, and

— network of functional units (FUs) description (also referred to as the decoder configuration)

that together constitute the entire decoder description (DD).

Bitstreams of existing MPEG standards are specified by specific syntax structures and decoders are composed of various coding tools. Therefore, RVC includes support for bitstream syntax descriptions as well as video coding tools. As depicted in Figure 1, a typical RVC decoder requires two types of information, namely the decoder description and the encoded media (e.g. video bitstreams) data.
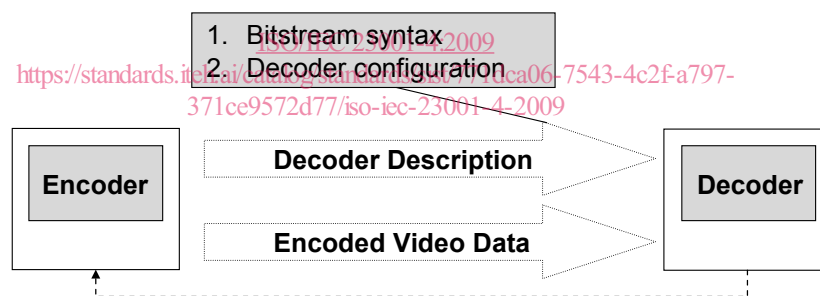
Figure 1 — Conceptual diagram of RVC

Figure 2 illustrates a more detailed description of the RVC decoder.

A more detailed description of the RVC decoder is shown in Figure 2. As shown in Figure 2, the decoder description is required for the configuration of a RVC decoder. The Bitstream Syntax Description (BSD) and FU Network Description (FND) (which compose the Decoder Description) are used to configure or compose an abstract decoder model (ADM) which is instantiated through the selection of FUs from tool libraries optionally with proper parameter assignment. Such ADM constitutes the behavioral reference model used in setting up a decoding solution under the RVC framework. The process of yielding a decoding solution may vary depending on the technologies used for the desired implementations. Examples of the instantiation of an abstract decoder model and generation of proprietary decoding solutions are given in Annex E.

Figure 2 — Graphical representation of the instantiation process or decoder composition mechanism for the RVC normative ADM and for the non-normative proprietary compliant decoder implementation.

Within the RVC framework, the decoder description describes a particular decoder configuration and consists of the FND and the BSD.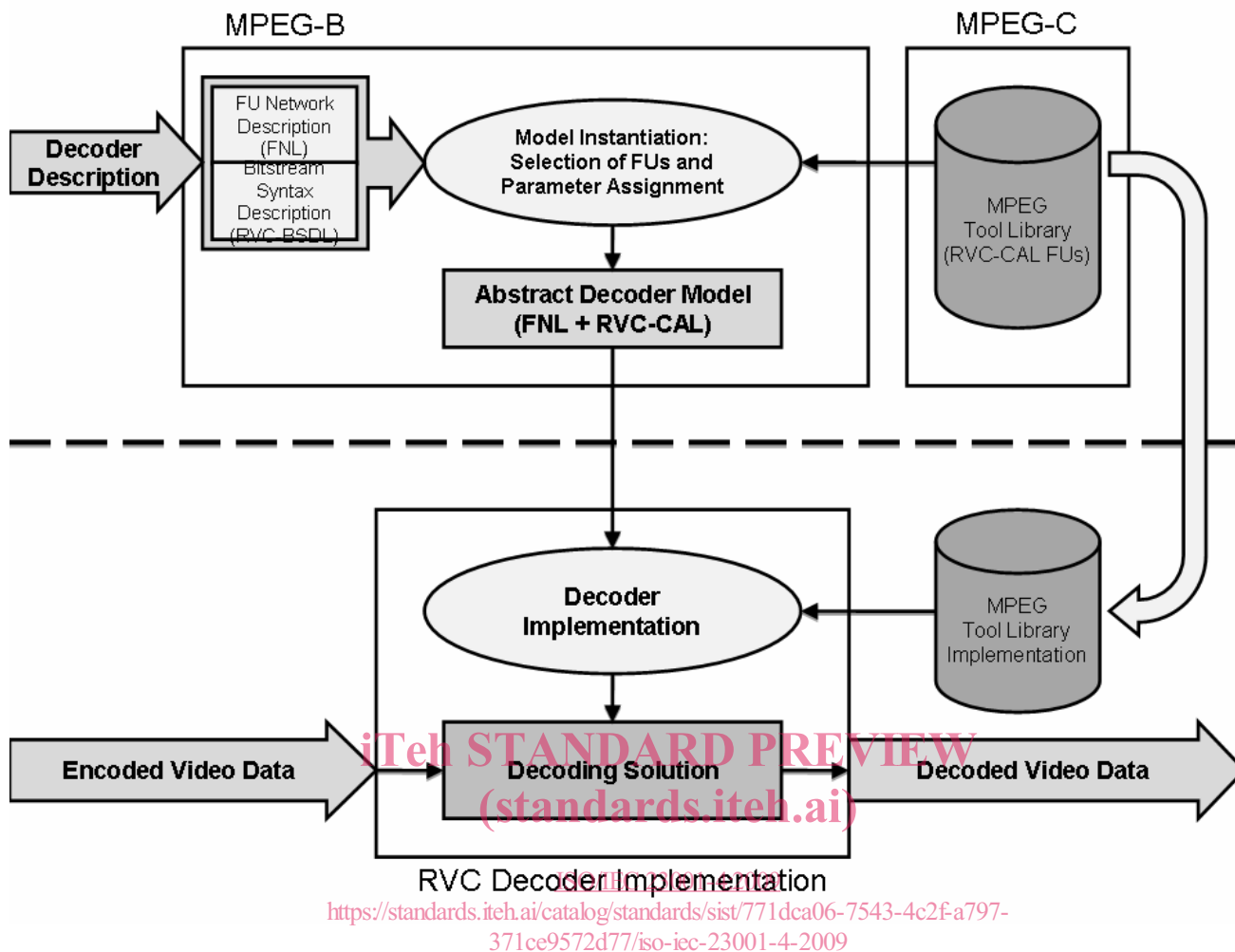 The FND describes the connectivity of the network of FUs used to form a decoder whereas the parsing process for the bitstream syntax is implicitly described by the BSD. These two descriptions are specified using two standard XML-based languages or dialects:

— Functional Unit Network Language (FNL) is a language that describes the FND, known also as "network of FUs". The FNL specified normatively within the scope of the RVC framework is provided in this part of ISO/IEC 23001;

— Bitstream Syntax Description Language (BSDL), standardized in ISO/IEC 23001-5 (MPEG-B Part 5), describes the bitstream syntax and the parsing rules. A pertinent subset of this BSDL named RVC-BSDL is defined within the scope of the current RVC framework. This RVC-BSDL also includes possibilities for further extensions, which are necessary to provide complete description of video bitstreams. RVC-BSDL specified normatively within the scope of the RVC framework is provided in this part of ISO/IEC 23001.

The decoder configuration specified using FNL, together with the specification of the bitstream syntax using RVC-BSDL fully specifies the ADM and provides an "executable" model of the RVC decoder description.

The instantiated ADM includes the information about the selected FUs and how they should be connected. As already mentioned, the FND with the network connection information is expressed by using FNL. Furthermore, the RVC framework specifies and uses a dataflow-oriented language called RVC-CAL for describing FUs' behavior. The normative specification of RVC-CAL is provided in this part of ISO/IEC 23001. The ADM is the behavioral model that should be referred to in order to implement any RVC conformant decoder. Any RVC

compliant decoding solution/implementation can be achieved by using proprietary non-normative tools and mechanisms that yield decoders that behave equivalent to the RVC ADM.

The decoder description, the MPEG tool library, and the associated instantiation of an ADM are normative. More precisely, the ADM is intended to be normative in terms of a behavioral model. In other words what is normative is the input/output behavior of the complete ADM as well as the input/output behavior of all the FUs that are included in the ADM.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 23001-4:2009
https://standards.iteh.ai/catalog/standards/sist/771dca06-7543-4c2f-a797-
371ce9572d77/iso-iec-23001-4-2009

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Information technology — MPEG systems technologies —

Part 4:
Codec configuration representation

1 Scope

This part of ISO/IEC 23001 defines the methods and general principles capable of describing codec configurations in the so-called reconfigurable video-coding (RVC) framework. It primarily addresses reconfigurable video aspects and will only focus on the description of representation for video codec configurations within the RVC framework.

Within the scope of RVC framework, two languages, namely FNL and RVC-BSDL, are specified normatively. FNL is a language that describes the FND, also known as "network of FUs". RVC-BSDL is a pertinent subset of BSDL defined in ISO/IEC 23001-5. This RVC-BSDL also includes possibilities for further extensions, which are necessary to provide complete description of video bitstreams.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

2 Normative references

ISO/IEC 23001-4:2009
https://standards.iteh.ai/catalog/standards/sist/771dea06-7543-4c2f-a707-
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 14496-2:2004, *Information technology — Coding of audio-visual objects — Part 2: Visual*

ISO/IEC 23001-5:2008, *Information technology — MPEG systems technologies — Part 5: Bitstream Syntax Description Language (BSDL)*

ISO/IEC 23002-4, *Information technology — MPEG video technologies — Part 4: Video tool library*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
abstract decoder model
ADM
conceptual model of the instantiation of the **FUs** (3.8) from the **video tool libraries** (3.16) and their connection according to the **FND** (3.9)

3.2
BSD
bitstream syntax description
description containing the bitstream syntax, its implicit parsing rules, and possibly tables [e.g. VLD tables if not already existing in the **RVC** (3.13) toolbox library] to define parser **FU** (3.8)

NOTE   The BSD is expressed using **RVC-BSDL** (3.14).

3.3
BSDL
description of the bitstream syntax and the parsing rules

NOTE   Bitstream Syntax Description Language (BSDL) is standardized by ISO/IEC 23001-5.

3.4
connection
link from an output port to an input port of an **FU** (3.8) that enables token exchange between FUs

3.5
decoder configuration
conceptual configuration of a decoding solution

NOTE 1   Using the **MPEG VTL** (3.12), decoder configuration can be designed as one of the following cases.

— A decoding solution of an existing MPEG standard at a specific profile and level.

— A new decoding solution built from tools of an existing MPEG standard.

— A new decoding solution built from tools of an existing MPEG standard and some new MPEG tools included in the MPEG video tool library.

— A new decoding solution that is composed of new MPEG tools included in the MPEG video tool library.

NOTE 2   In summary, an RVC decoder description essentially consists of a list of **FUs** (3.8) and of the specification of the FU connections [**FND** (3.9) expressed in **FNL** (3.10)] plus the implicit specification of the parser in terms of bitstream syntax description [BSPD expressed in **RVC-BSDL** (3.14)]. In order to be a complete behavioral model [i.e. **ADM** (3.1)] an RVC **decoder description** (3.6) needs to make reference to the behavior of each FU that is provided in terms of I/O behavior by the RVC video toolbox library specified in RVC-CAL.RVC.

3.6
decoder description
DD
description of a particular decoder configuration, which consists of two parts; **FU network description** (3.9) and **bitstream syntax description** (3.2)

3.7
decoding solution
implementation of the **ADM** (3.1)

3.8
FU
functional unit
modular tool which consists of a processing unit characterized by the input/output behavior

3.9
FND
FU network description
FU (3.8) connections used in forming a decoder which are modeled using **FNL** (3.10)

3.10
FNL
FU network language
language that describes the **FND** (3.9), known also as "network of FUs"

3.11
model instantiation
building of the **abstract decoder model** (3.1) from the **decoder description** (3.6) [consisting of the **BSD** (3.2) and the **FND** (3.9)] and from **FUs** (3.8) from the **VTL** (3.16)

NOTE   During the model instantiation, the parser FU is reconfigured according to the BSD or loaded from VTL.

3.12
MPEG video tool library
MPEG VTL
video tool library (3.16) that contains **FUs** (3.8) defined by MPEG (i.e. drawn from existing MPEG International Standards)

3.13
RVC
reconfigurable video coding
framework defined by MPEG to promote coding standards at tool-level while maintaining interoperability between solutions from different implementers

3.14
RVC-BSDL
pertinent subset of **BSDL** (3.3), which is defined within the scope of the current **RVC** (3.13) framework

3.15
token
data entity exchanged between input and output among **FUs** (3.8)

3.16
VTL
video tool library
collection of **FUs** (3.8)

4 Functional unit network description

4.1 Introduction

The FUs in MPEG RVC are specified by:

- The textual description in ISO/IEC 23002-4.

- The RVC-CAL reference software. RVC-CAL is formally specified in Annex D.

The Functional Unit Network Language (FNL) is formally specified in this Clause and is used to describe networks of FUs. FNL is derived from Extensible Markup Language (XML) which was in turn derived from SGML (ISO 8879). The ADM consists of a number of FUs with input and output ports, and the connections between those ports. In addition, the ADM may have input and output ports, which may be connected to the ports of FUs or to each other.

A decoder can be described as a network of one or more FUs even when there is only one FU in the decoder (e.g. Figure 3).
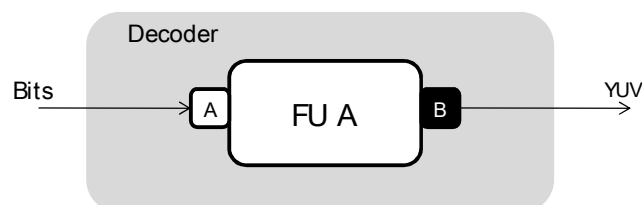

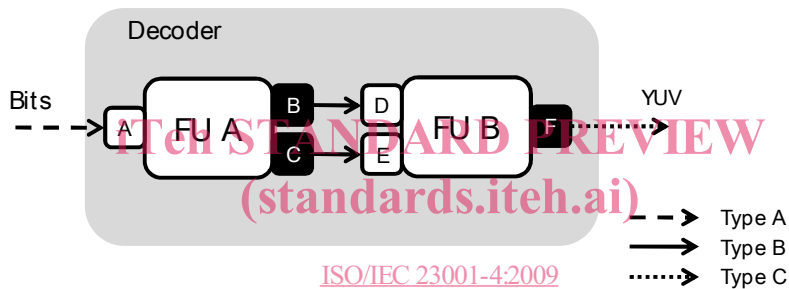**Figure 3 — FU Network of an FU**

A network of FUs is described in FND. A FND includes the list of the selected FUs to form the decoder and the three types of connections that are connections between FUs (type A), connections between decoder inputs and FU inputs (type B), and connections between FU outputs and decoder outputs (type C), which is illustrated in Figure 4.

The list of the selected FUs (Figure 4) is described in FND according to the following table. When selecting FUs from VTL, the ID and name of FUs are used in the FND. The ID and name of the FUs are specified in ISO/IEC 23002-4. The parameter assignment in the listed FU is supported in the FND, but optional.

```
<Instance  id = "FU A">
     <Class name = "Algo_Example1" />
</Instance>
<Instance id = "FU B">
     <Class name = "Algo_Example2" />
</Instance>
```

The connections (type A, type B, and type C of Fig. B) are described in FND as shown in the following table.

Type A	`<Connection src = "FU A" src-port = "B" dst = "FU B" dst-port = "D" />` `<Connection src = "FU A" src-port = "C" dst = "FU B" dst-port = "E" />`
Type B	`<input src = "FU A" src-port = "A" />`
Type C	`<output src = "FU B" src-port = "F" />`

Figure 4 — Three types of connection in an FU network

Another example of FU network with four FUs is illustrated in the Figure 5. The textual description of Figure 5 in FND is described as follows.

```
<XDF name="Decoder">
<Instance id = "Syntax parser">
    <Class name = "syntax parser">
</Instance>
<Instance id = "FU A">
    <Class name = "Algo_ExamFU_A">
</Instance>
<Instance id = "FU B">
    <Class name = " Algo_ExamFU_B">
</Instance>
<Instance id = "FU C">
    <Class name = " Algo_ExamFU_C">
</Instance>
<Input src = "Syntax Parser" src-port = "A" />
<Output src = "FU C" src-port = "R" />
<Connection src = "Syntax Parser" src-port = "B" dst = "FU A" dst-port = "E" />
<Connection src = "Syntax Parser" src-port = "C" dst = "FU A" dst-port = "F" />
<Connection src = "Syntax Parser" src-port = "D" dst = "FU B" dst-port = "K" />
<Connection src = "FU A" src-port = "H" dst = "FU C" dst-port = "O" />
<Connection src = "FU B" src-port = "L" dst = "FU C" dst-port = "P" />
<Connection src = "FU B" src-port = "M" dst = "FU C" dst-port = "Q" />
</XDF>
```
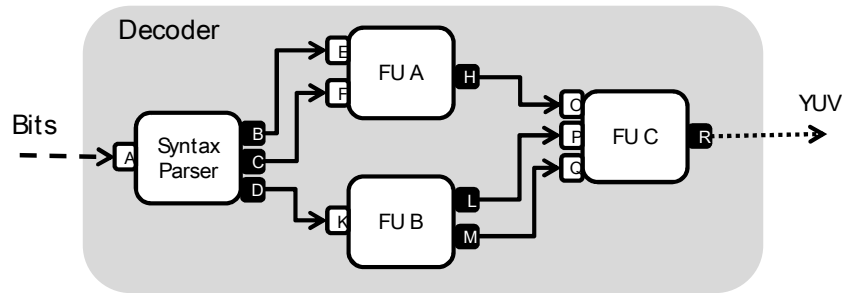
Figure 5 — Another example of FU network

4.2 The specification of a network of FUs

The XML structures with names of elements, such as Decl, Network, Package, Expr, etc. are described in Annex A. In addition, attributes that direct an individual element's features are also introduced. Attribute names will be prefixed with "@". For instance common attribute names are @id, @name, or @kind. In cases where an element name may be qualified by the value of an attribute, square brackets are used. For instance, in order to express the notion of an Expr element whose @kind attribute is the string "literal", Expr[@kind="literal"] is written.

By using the RVC-CAL model, FNL also allows networks of FUs and individual FUs to be parameterized. In particular, it is possible to pass bounded values for specific parameters into FU and/or FU networks. These values are represented by Expr and Decl syntax. Expr and Decl are the syntactical constructs describing a computation, which may, itself, be dependent upon the values of parameters which are either global or local variables.

5 Bitstream syntax description

The MPEG Video Tool Library contains FUs that specify MPEG decoding algorithms. A new decoder configuration implies new bitstream syntax. The description of the bitstream syntax in RVC is provided using BSDL as specified in ISO/IEC 23001-5 and BSDL schema. However, to facilitate the developments of synthesis tools that are able to generate parsers directly from a BSD (i.e. a BSDL schema), the RVC framework standardizes a version of BSDL called RVC-BSDL specified by including new RVC specific extensions and usage restrictions of standard BSDL in ISO/IEC 23001-5. Such extensions and restrictions versus the MPEG standard BSDL are defined in Annex C of this document. RVC-BSDL contains all information necessary to parse any bitstream compliant with such syntax. The procedure to instantiate the parser capable of parsing and decoding any bitstream compliant with the syntax specified by the RVC-BSDL schema is not normative. Examples of such non-normative procedures are provided in Annex D.

6 Model instantiation

This Clause describes the "model instantiation" process which consists of the selection of Functional Units (FUs) from the Video Tool Libraries and instantiation of the FUs with the proper parameter assignments. The instantiation process requires the following information:

- The Video Tool Libraries
- The FU network description
- The bitstream syntax description

The instantiation process consists of attaching source codes corresponding to the FUs identified in the FND in order to build a complete model, which can be simulated. The Video Tool Libraries are repositories of source code for each Functional Units. The FND contains only the references (name of the FUs) to the pieces of code in the VTLs. The process outputs the ADM, which is the result of the gathering of all the pieces of code of all FUs. Figure 6, illustrates the Model Instantiation process.
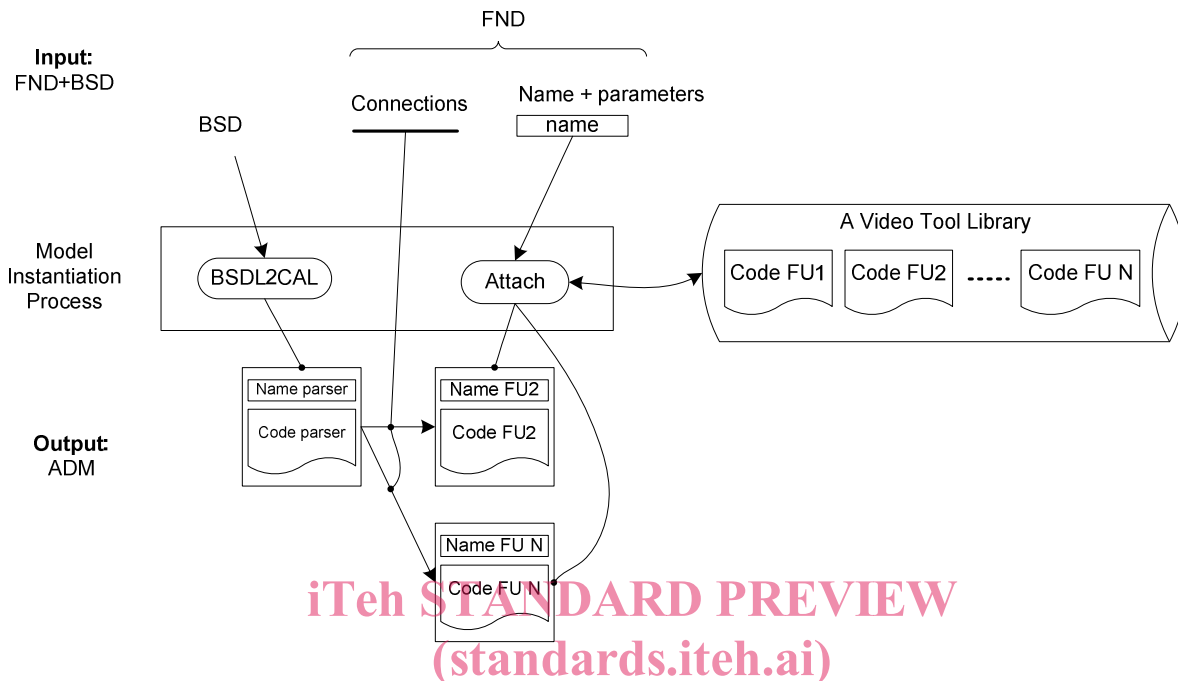

Figure 6 — Description of the model instantiation process

The FU Network Description (FND) provides the structure of the decoder by giving the names of the FUs composing the decoder and their respective connections between each other. The name of the instance of the FU in the ADM is contained in the tag `<instance id="…">`. The tag `<class name="…">` indicates the name of the FU (in the video tool library) from which the FU of the ADM must be instantiated. The tag `<parameter>` provides the values of the parameters, which must be used for the instantiation of the FU in the ADM.

The Bitstream Syntax Description (BSD) provides the structure of the bitstream. The parser is generated automatically from the BSD. Informative examples are provided in Annex E for building the parser. The syntax parser FU of the ADM might use other FUs to parse the bitstream. Thus, a clear link between identifiers inside the BSD and the FND must be established. The tag `<rvc port="…">` indicates the name of the instance of the FU into the ADM to which this element of syntax is sent.

Annex A
(normative)

Functional unit network description

A.1 Elements of a functional unit network

XDF — A FU network is identified by the root element called XDF that marks the beginning and end of the XML description of the network.

- optional attribute: @name, the name of the network. @version, the version number of the current network. Assumed to be "1.0" if absent.

- optional children: Package, Decl, Port, Instance, Connection.

```
<XDF name="mpeg4SP">
   ...
</XDF>
```

Package — This element contains a structured representation of a qualified identifier (QID) (i.e. a list of identifiers that are used to specify a locus in a hierarchical namespace). That QID provides the context for the @name attributed of the enclosing Network element: that name is intended to be valid within the specified namespace, or package.

- required child: QID, the qualified identifier.

```
<Package>
   <QID>
      <ID id="mpeg4"/>
   </QID>
</Package>
```

Decl[@kind="Param"] — Represents the declaration of a parameter of the network.

- required attribute: @name, the name of the parameter.

- optional child: Type, the declared type of the parameter.

```
<Decl kind="Param" name="FOURMV"/>
```

Decl[@kind="Var"] — This element represents a variable declaration. Variables are used within expressions to compute parameter values for actors instantiated within the network and within expressions used to compute the values of other variables.

- required attribute: @name, containing the name of the declared variable.

- required child: Expr, representing the expression defining the value of this variable, possibly referring the values of other variables and parameters.

- optional child: Type, the declared type of the variable.

```
<Decl kind="Variable" name="MOTION">
      <Expr kind="Literal" literal-kind="Integer" value="8"/>
</Decl>
```