INTERNATIONAL STANDARD

ISO/IEC 21000-8

Second edition
2008-03-01

AMENDMENT 2
2011-11-01

Information technology — Multimedia framework (MPEG-21) —

Part 8: Reference software

AMENDMENT 2: Reference software for media value chain ontology (MVCO)

Technologies de l'information — Cadre multimédia (MPEG-21) —

Partie 8: Logiciel de référence

AMENDEMENT 2: Logiciel de référence pour l'ontologie de la chaîne de valeur du support (MVCO)

Reference number
ISO/IEC 21000-8:2008/Amd.2:2011(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

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.

Amendment 2 to ISO/IEC 21000-8:2008 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This Amendment describes extra reference software for ISO/IEC 21000-19:2010.

Information technology — Multimedia framework (MPEG-21) —

Part 8:
Reference software

AMENDMENT 2: Reference software for media value chain ontology (MVCO)

In 3.2 Abbreviated terms, insert the following acronym after MPEG:

MVCO Media Value Chain Ontology

At the end of 4.2 Overview of ISO/IEC 21000 reference software, add:

— ISO/IEC 21000-19:2010, Information technology — Multimedia framework (MPEG-21) — Part 19: Media Value Chain Ontology (MVCO): The corresponding reference software modules provide an environment to test MVCO based applications. A reference to the reference software for ISO/IEC 21000-19:2010 is described in 5.22

In Clause 5 Reference software for the ISO/IEC 21000 parts, insert 5.22:

5.22 ISO/IEC 21000-19:2010

5.22.1 Introduction

This Clause describes the reference software for the ISO/IEC 21000-19 Media Value Chain Ontology (MVCO).

The MVCO Reference Software consists of a series of independent applications demonstrating some of the most prominent features of the MVCO. These applications work together in a simple architecture shown in Figure 17. This reference software is a superset of the API described in Annex A.1 in the MVCO standard. The MVCO Reference Software architecture comprises four Devices and one application for the management of one of them.

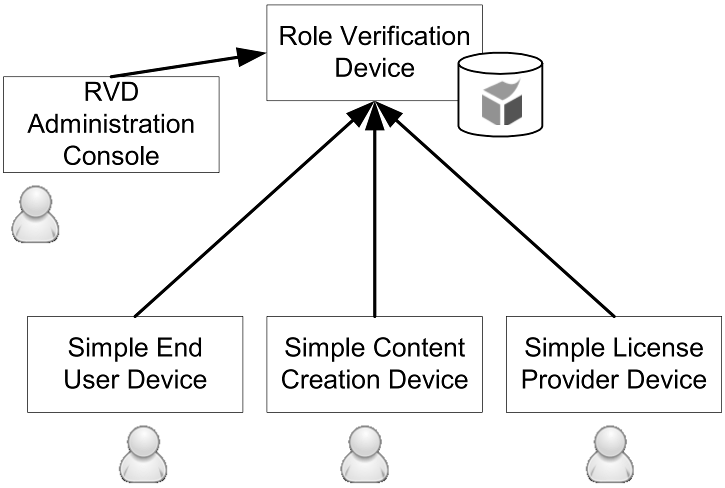

Role Verification Device
RVD Administration Console
Simple End User Device
Simple Content Creation Device
Simple License Provider Device

Figure 17 — ISO/IEC21000-19 Reference Software architecture

NOTE The software model for this architecture is that of MPEG-M, whose terminology is used here. However, MXM Devices are out of scope of the ISO/IEC 23006 standards and therefore its adoption is not mandatory.

The Devices present in the architecture are the following:

- The Role Verification Device (RVD). A server application hosting the MVCO ontology and instances attending the rest of the modules queries and information messages. It implements the API as described in ISO/IEC 21000-19 Annex A.1 to offer the services described in the next Clause.

- A Simple Content Creation Device (SCCD). A simple Content Creation Device, to be eventually replaced by the CCD as informatively described in ISO/IEC 23006-1

- A Simple License Provider Device (SLPD). A simple License Provider Device, to be eventually replaced by the LPD as informatively described in ISO/IEC 23006-1

- A Simple End User Device (SEUD). A simple End User Device, to be eventually replaced by the EUD as informatively described in ISO/IEC 23006-1

- An RVD administrator console (RVDAC), able to supervise the operations and determine the state of the MVCO ontology.

The first component (RVD) acts as a server application (Web Service or any other technology considered convenient). The simple Devices (SCCD, SLPD and SEUD) are console applications, able to execute or simulate a very much reduced set of operations and communicate with the RVD. These applications do not pretend to have any other objective than demonstrate the use of the RVD. The RVDAC is also a console application, able to display state information and at the same time to receive commands from the RVD Administrator Console.

5.22.2 Role Verification Device (RVD)

Module Name: /21000-19 MVCO/MVCO-RVD-A-1.zip

Description

It renders MVCO-based services, handling an archive of MVCO instances.

Input

- A SOAP message requesting a RVD service, given as an array of Strings
- The following parameters can be changed by editing a text file next to the jar file named *rvd.properties:*
  - port This is the port number where the web service server will be listening. Example (and default value): port = 8087
  - debug This parameter defines which level of logs will be created. Logs are given in the rvdlog.txt file. Valid values are OFF, ERROR, DEBUG. Example (and default value): debug = ERROR
  - mvcodir File where the instances will be installed. Example (and default value): mvcodir = rvd.owl
- The following parameters can be changed by editing a text file next to the jar file named *mvco.properties*.
  - local Path to the location of the MVCO ontology (it can be a local file for faster loading and not depending on a internet connection). Example (and default value): local = file:mvco.owl
  - debug This parameter defines which level of logs will be created (related to the MVCO). Logs are given in the mvcoapilog.txt file. Valid values are OFF, ERROR, DEBUG. Example (and default value): debug = ERROR
  - reasoner This parameter defines which reasoner will be used. Valid values are PELLET, JENA. Example (and default value): reasoner = PELLET

Output

- A SOAP message with the response, including as an array of Strings

Programming Language(s)

Java

Platform(s)

Any platform supported by the Java 1.5 or higher.

Dependencies

- Java version "1.5" or later
- Jena and derived components

- Pellet and derived components
- Axis2 and derived componetns
- Log4Java logging library

Details

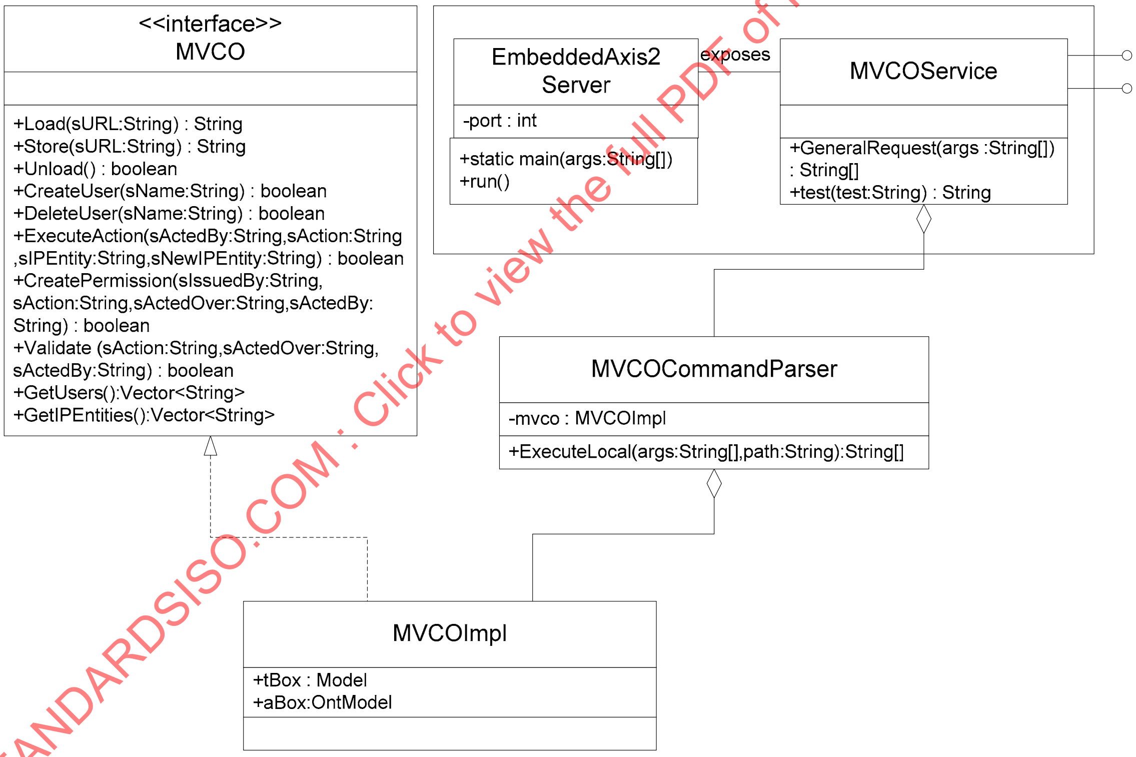
The Role Verification Device (RVD) is a Web Service provider, giving access to a MVCO ontology with instances.

The RVD is entirely developed in Java and it does not require to be deployed in any Web Server, given that it embeds already one. It makes use and demonstrates the MVCO API given in Annex A in ISO/IEC 21000-19.

The RVD consists of two packages:

- `org.iso.mpeg.mpeg21.mvco.mvcoapi`
- `org.iso.mpeg.mpeg21.mvco.rvd`

Class diagram in Figure 18 shows the UML2 class diagram for the RVD main classes. The region within the slashed line corresponds to the package `rvd`, belonging the rest to the `mvcoapi` package.



<<interface>>
MVCO
+Load(sURL:String) : String
+Store(sURL:String) : String
+Unload() : boolean
+CreateUser(sName:String) : boolean
+DeleteUser(sName:String) : boolean
+ExecuteAction(sActedBy:String,sAction:String ,sIPEntity:String,sNewIPEntity:String) : boolean
+CreatePermission(sIssuedBy:String, sAction:String,sActedOver:String,sActedBy: String) : boolean
+Validate (sAction:String,sActedOver:String, sActedBy:String) : boolean
+GetUsers():Vector<String>
+GetIPEntities():Vector<String>
EmbeddedAxis2 Server
-port : int
+static main(args:String[])
+run()
exposes
MVCOService
+GeneralRequest(args :String[]) : String[]
+test(test:String) : String
MVCOCommandParser
-mvco : MVCOImpl
+ExecuteLocal(args:String[],path:String):String[]
MVCOImpl
+tBox : Model
+aBox:OntModel

Figure 18 — RVD class diagram

Symbols used in Figure 18 are those standard in a UML2 class diagram, excepting the offered interface by the MVCOService, more proper for a components diagram but justified here given that the MVCOService is a POJO (*Plan Old Java Object*) class entirely exposed as web service.

Methods in the MVCO API interface are fully described in ISO/IEC 21000-19. Methods in the `MVCOImpl` class merely implement the MVCO API interface.

Both the MVCOService and MVCOCommandParser accept a String array as a parameter (the latter additionaly permits specifying the file where the instances will be read from/stored to). That String array contains a set of strings conforming a command and parameters. The accepted commands are shown in Table 2.

Table 2 — Commands to be executed on the ontology

Table 1 — command	Semantic	Parameters	Return
u	Unloads the ontology		(String) “ok” on success
lu	Lists the users in the ontology		(String []) Users in the ontology
lip	Lists the IP Entities in the ontology		(String []) IP Entities in the ontology
cu USERNAME	Creates a user	USERNAME unique name of the user name to be created	(String) “ok” on success
du USERNAME	Deletes a user	USERNAME unique name of the user name to be deleted	(String) “ok” on success
cip ACTEDBY ACTION ACTEDOVER NEWIPENTITY	Executes an action	ACTEDBY unique name of the user which acts the action ACTION to be acted ACTEDOVER IPEntity over which the action is acted, if any, or “none” NEWIPENTITY name of the created IPEntity, if any, or “none”.	(String) “ok” on success “error” on error “not permitted” if an authorisation is needed
cp ISSUER ACTION ACTEDOVER ACTEDBY	Creates a permission	ISSUER user issuing the permission ACTION permitted action (obtained from a create action command) ACTEDOVER resource over which the action is acted ACTEDBY user which is entitled to act the action	(String) name of the created permission
v ACTION ACTEDOVER ACTEDBY	Validates an action	ACTION action to be validated ACTEDOVER IP Entity over which the action is acted ACTEDBY user which executed the action	(String) ok on success

Add A.12:

A.12 ISO/IEC 21000-19:2010

Annex A.12 describes the utility software for ISO/IEC 21000-19.

A.12.1 Role Verification Device Administration Console (RVDAC)

Module Name: `/21000-19 MVCO/MVCO-RVDAC-A-1.zip`

Description

It is a command-line interpreter of operations, which can be acted either locally or remotely on a remote RVD

Input

A command line command, as described in Table 2.

Also, The following parameters can be changed by editing a text file next to the jar file called *rvdac.properties.* Also, those parameters from the MVCO API can be modified in the *mvco.properties* file as described in 5.22.2

- remoteip This is the IP address of the remote RVD. For example: `remoteip = 147.83.34.226`
- port This is the port number where the web service client will try to connect. Example (and default value): `port = 8087`
- debug This parameter defines which level of logs will be created. Logs are given in the rvdaclog.txt file. Valid values are OFF, ERROR, DEBUG. Example (and default value): `debug = ERROR`
- mvcodir File where the instances will be installed, in case local operations are invoked. Example (and default value): `mvcodir = rvdac.owl`

Output

The result of the operation, acted in an ontology file (either local or remote) , plus a summary on the screen.

Programming Language(s)

Java

Platform(s)

Any platform supporting Java 1.5

Dependencies

- Java version "1.5" or later
- Axis2 and derived components
- Log4java

Details

The RVDAC permits invoking operations on the ontology either locally or remotely.

To act on the ontology locally, the RVDAC makes use of the MVCO method `ExecuteLocal`. To act on the ontology remotely, the `ExecuteRemote` RVDClient method is invoked. This method is Web

Service client of the services exposed by MVCOClient.

Command line parameters in the RVDAC application are processed. If preceded by the option `-r`, they are executed remotely. If not, they will be executed locally. For the rest, parameters described in Table 2, fully apply here. Class diagram:

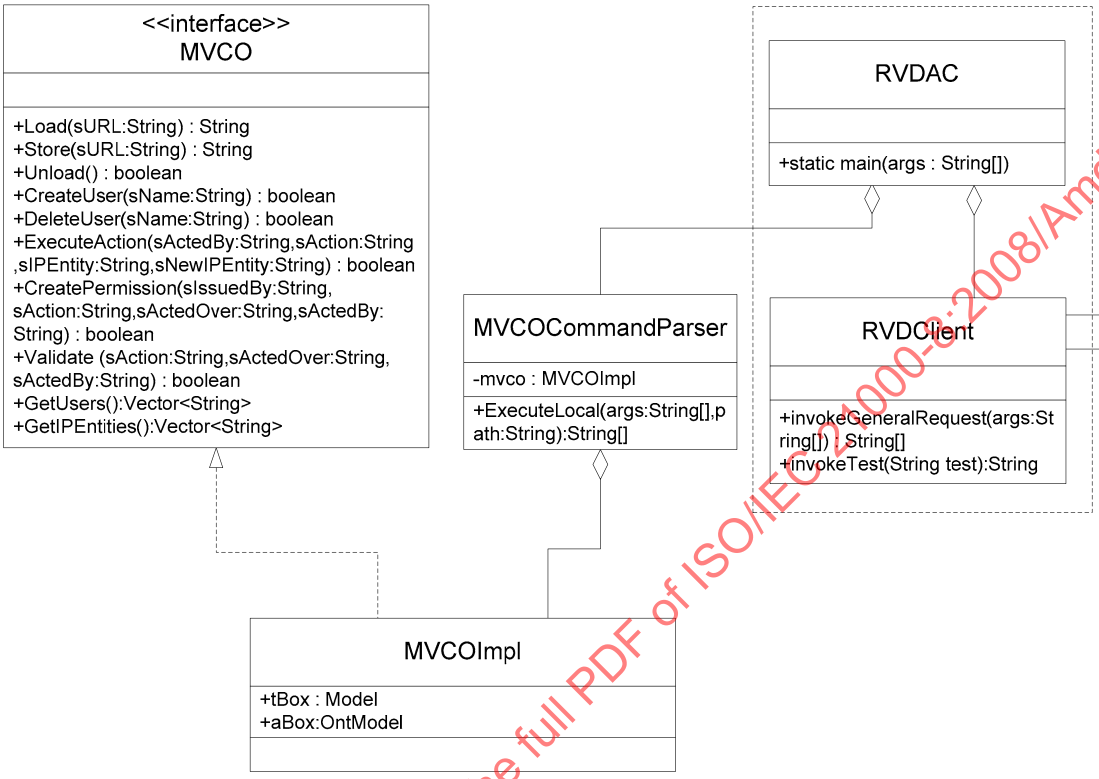

<<interface>>
MVCO
+Load(sURL:String) : String
+Store(sURL:String) : String
+Unload() : boolean
+CreateUser(sName:String) : boolean
+DeleteUser(sName:String) : boolean
+ExecuteAction(sActedBy:String,sAction:String ,sIPEntity:String,sNewIPEntity:String) : boolean
+CreatePermission(sIssuedBy:String, sAction:String,sActedOver:String,sActedBy: String) : boolean
+Validate (sAction:String,sActedOver:String, sActedBy:String) : boolean
+GetUsers():Vector<String>
+GetIPEntities():Vector<String>
RVDAC
+static main(args : String[])
MVCOCommandParser
-mvco : MVCOImpl
+ExecuteLocal(args:String[],path:String):String[]
RVDClient
+invokeGeneralRequest(args:String[]) : String[]
+invokeTest(String test):String
MVCOImpl
+tBox : Model
+aBox:OntModel

Figure A.10 — RVDAC class diagram

A.12.2 Simple Content Creation Device (SCCD)

Module Name: /21000-19 MVCO/MVCO-SCCD-A-1.zip

Description

The SCCD is a command-line interpreter of operations which can register a new Digital Item in the Role Verification Device, following the complete value chain
It can also be a Graphical User Interface, if the application is invoked with no parameters.

Input

— A command line command, with two parameters.

  — Name of the user using the application. This user must exist in the RVD.

  — Path to a JPG file, no bigger than 1 Mb.

— It can also be Graphical User Interface, if the application is invoked with no parameters.