

Minutes

MEETING NAME 09-WG33: WG-33 Data Archive and Management

MEETING PLACE/DIAL IN

DATE & TIME Wednesday, August 5, 2020 | 11:00 am – 12:30 pm US ET

PRESIDING OFFICERS Matthew Bishop, UnityPoint Health
Keith Eklund, Healthcare Tech Solutions

<u>VOTING MEMBERS PRESENT</u>	AAPM	Bevins, Nicholas
	Ambra Health	Ostrow, Daniel
	Argentix Informatics	Silver, Elliot
	Canon/Vital Images, Inc.	Dawson, Tim
	Canon/Vital Images, Inc.	Whitby, Jonathan
	GE Healthcare	Nichols, Steven
	GE Healthcare	Numan, Jouke
	Grafimedia	Georgiadis, Pantelis
	Healthcare Tech Solutions	Eklund, Keith
	Laitek, Inc.	Brown, Barry
	Laitek, Inc.	Costea-Barluti, Razvan
	Laitek, Inc.	Solomon, Harry
	PixelMed Publishing	Clunie, David
	SIIM	Carey, Cheryl
	SIIM	Henson, Kyle
	Varian Medical Systems, Inc.	Schwere, Thomas

<u>OTHERS</u>	CitiusTech Healthcare Technology LTD	Mahalle, Prashant
	DeJarnette Research Systems	Wineke, Steve
	Hyland Software, Inc.	Ullrich, Mike
	Laitek, Inc.	Behlen, Fred

London Health Science Centre Aizawa, Luiz

<u>VOTING</u>	AAPM	Knazik, Shayna
<u>MEMBERS</u>	Canon Medical Research	O'Donnell, Kevin
<u>ABSENT</u>	USA Inc.	
	Change Healthcare	Ho, Kinson
	DesAcc EMEA Ltd	King, Graham
	ESR	Mildenberger, Peter
	Laitek, Inc.	Sluis, Douglas
	Mach7 Technologies	Ulanov, Alexey
	Mayo Clinic	Persons, Kenneth
	Society for Imaging Informatics in Medicine	Bishop, Matthew

DICOM Anna Zawacki, SIIM
SECRETARIAT

1 CALL TO ORDER AND REVIEW OF ANTI-TRUST RULES AND DICOM PATENT POLICY (Co-Chairs, Secretariat)

The meeting was called to order. Guidelines for Conducting NEMA Meetings were read and attendance was recorded.

2 REVIEW AND APPROVE AGENDA (Co-Chairs)

The agenda was reviewed and approved.

3 REVIEW MINUTES (Co-Chairs)

The minutes of the previous meeting were reviewed and approved.

4 TOPIC ITEMS TO BE DISCUSSED (All)

1. Resume reviewing open issues to be addressed in the Supplement at #10

ftp://d9-workgrps@medical.nema.org/MEDICAL/Private/Dicom/WORKGRPS/WG33/2020/2020-08-05/Consensus_Positions_and_Open_Issues-20200805.pptx

10

Open Issue – multiple views of metadata

- Use case – in enterprise archives, different users may desire different views of the metadata (e.g., different Patient IDs)
- DICOM base capabilities
 - Multiple Patient IDs in metadata [Other Patient IDs Sequence (0010,1002)]
 - Multiple AETitles (or DICOMweb service points) per server allows separation of use context
 - No current DICOM specification of user identity / permissions / context, but identity can be passed through DIMSE or DICOMweb over TLS, or OAuth with DICOMweb
- **Option 1: standardized use of user identity to control differential content of inventory**
- **Option 2: use query parameter to control differential content of inventory**
- **Option 3: differential content is out of scope**

Different users might have different views of the metadata, esp. if you are in a multi-institutional setting and especially when it comes to patient ID.

Do we need to address this in the context of our specification?

No way to pass user context using DICOM protocol side of things. Overload AETitle and depending on what AETitle you call our device with, you get a specific view of the data. That should be sufficient for our purposes at least initially.

Slide 11 and 12 issues – closed

I3

Open Issue – access inventory instances

- Use case – user apps need to access inventory instances after their creation
 - Presumption: in DICOM category of Non-Patient Objects, optionally stored in a Part 10 file
- Inventory completion notification should convey Instance UID and available access mechanisms
 - DIMSE, DICOMweb, URI / filesystem access to a Part 10 compliant instance
- C-MOVE / C-GET + C-STORE based services (pull), and corresponding DICOMweb
 - DIMSE Non-Patient Object Storage ([Part 4 Annex GG](#)), IOD-specific Retrieve SOP Class
 - DICOMweb Non-Patient Instance Service and Resources ([Part 18 Section 12](#)) based on DIMSE specs
- **Proposal: Inventory instances shall have defined access mechanisms equivalent to other DICOM Non-Patient Objects**
 - **Separate conformance claim for each mechanism**

How does the user application access these instances?

We have DIMSE and DICOMweb

Only question – given the size of the inventory, do we potentially want to have end run access and be able to provide a direct file URI? See next slide.

I3a

Open issue – DIMSE incremental access mechanism

- DICOMweb and filesystem can access inventory object incrementally, but C-STORE requires transfer and acknowledgment of complete object in single pass
 - Problematic for very large inventory object with millions of Study records
- **Option 1:** access mechanism for random access to blocks of records
 - Cf. frame-based retrieve access for large multi-frame images ([Part 4 Annex Y](#)) - creates new object with selected content
 - Selection key other than record number?
- **Option 2:** SCP partitions total logical inventory into smaller inventory objects
 - Cf. Concatenations for large multi-frame images
 - All constituent chunks identified in inventory completion notification
 - Chunks related by explicit listing in meta-directory, and/or by implicit shared Concatenation UID
 - But see Issue #7

With large objects, do we need some sort of an incremental access mechanism? Option # 3 – is to say, oh well, maybe people just won't use DIMSE C-STORE mechanism for this, they will just use file system access or DICOMweb access. DICOMweb access and file system access require the knowledge of which part to jump into though. Without an index of where things are, you don't know where to jump to. That assumes you want just a piece and not the entire inventory object.

There is potentially a desire to get subparts of the object that have some kind of scope of relevance.

What are the use cases that require you to get a monster inventory as opposed to a lot of moderate-size inventories? And in which of the use cases do we require range-based access mechanisms?

For the migration use case you're going to need the entire inventory.

You'll need a set of inventories, whether one large or multiple smaller ones.

When you anticipate having something really large, you get it in bits by using selective queries.

It is highly likely that every implementation is going to have a max limit on the size of things they are going to create. Things like a wide open query may be rejected. Monster inventory may not be supported feasibly.

Do we want to go back to issue # 7 – and have the server to be able to create a complete inventory but be able to pre-chunk it into smaller inventory objects that cover the necessary subsets to bring the size within a reasonable limit?

So how would it be different from doing a normal query?

There are a couple of differences from the current C-FIND mechanism:

1. When you make a request, it's creating a persistent object with a unique identifier that can presumably be deemed to be a snapshot in time
2. Given that the query follows a certain pattern, it may be something that is pre-generated by the archive in periodic intervals

Once you have created a snapshot it's presumable to be accessible for a certain period of time.

We want to be able to create a snapshot that has a stable set of information that is either predefined or user specified and retrievable in bits and pieces as necessary with a full expectation that you will repeat your snapshot request and try to find delta.

Also propose we do that snapshot can be done at study level.

Disagree, it should be at an instance level. Content of study may vary over time.

Have an interface that can specify - give me all the metadata for this study or this instance.

Basic proposal – you want a snapshot of everything in the archive at a point in time.

Come up with a mechanism to migrate the data, snapshot is one way of getting the data.

Agreed earlier on – in a migration, we still need an inventory of everything.

If we were going to do something like frame-level access – would every study that got returned need something akin to a frame idea that we could ask for and have a knowledge of what that was. Whether each would have a record number. Don't know if it's necessary. You can mimic it with a more constrained query.

2 step operation:

2. Op 1 - make a snapshot and return me the snapshot ID, then
3. Op 2 - make subsequent queries of any granularity you want

Make a new snapshot or return the most recent snapshot

Call to generate the inventory

Call to retrieve the inventory once it's generated

What is our tractable object size? If we're going to need to break it down in any way, why not just allow the server to break it?

You almost have to.

14

Open Issue – Query for inventory objects

- App using inventory may not be same as app invoking its creation
 - Inventories created on periodic (e.g., monthly) basis by PACS automation script
 - Does not receive inventory completion notification with Instance UID and access mechanisms
- Need a general mechanism to list available inventory instances and their scopes
- Each class of DICOM Non-Patient Objects has a Query SOP Class with IOD-specific query and return keys (cf. Color Palette Q/R [Part 4 Annex X](#))
 - Also used as definition for DICOMweb service ([Part 18 Section 12](#))
- **Proposal: Inventory instances shall have defined query mechanisms equivalent to other DICOM Non-Patient Objects**
 - Return keys include inventory scope

At the very least we have to be able to query for whatever the most recent snapshot is

Presume there is no presumption the server needs to keep these around for any defined amount of time

You can also have an explicit snapshot release

2 different systems might want to do 2 different snapshots

Concerning n the retention side especially in the migration use case – does the existence of the snapshot of the inventory need to persist on the source system for the duration of the migration such that it can be referenced or once it's retrieved can it be concerned safe to be deleted.

Do we have to make it part of the standard? No, but it's good when we identify such things to explicitly state it is not part of the standard.

Have something similar to the storage commitment for this.

Brings us back to the security argument.

How do you know what the intended requestor is?

Create new issue of the persistence of the inventory object.

15

Open Issue – stored instance folder/directory structure

- Use case – often all objects in Series or in Study need to be retrieved
 - Filesystems support copy/move functionality on set of files in folder/directory
 - Functionally equivalent to C-MOVE at Study or Series level
- Storage of objects in a study / series folder hierarchy would facilitate such access
 - [Alternative – operations on files must always be handled one at a time]
- PACS may also store all objects of a Study or Series in a ZIP or TAR file
- **Proposal: inventory shall support optional link at Study and Series level for a folder or ZIP/TAR that contains all objects of that information entity**

We'll have to constrain what other than Part 10 DICOM on disc representation going is to be?

What about those systems that separate the metadata in the bulk data?

Do we want to support the ability to provide both of those – some assembly will be required on the receiver side.

This issue is expanding on issue 2 by adding study levels and study level folders. Are ZIP and TAR sufficient?

G ZIP TAR are often used

Are we trying to resolve this issue for existing systems or for the future?

How do we imagine compliance on the migrators' side?

If they can't do direct file access, they can always do a C-STORE

What are the compliance requirements here?

Are we allowing an unbounded list of choices?

That's why constraining issue 2 to G ZIP TAR, anticipate list may grow in the future.

DSC approved this WG completely aware that this fully undermines the entire paradigm of DICOM interoperability and opening the door for people to do anything they want with the DICOM files.

Purpose of this WG - resolving migration case of the future – encourage vendors to use standardized ways in which they store data.

Right now we are trying to solve the existing problem, existing migration case.

16

Open Issue – canonical filenames

- For network Retrieve (DIMSE, DICOMweb), objects and sets of objects (series and studies) are identified by UID(s)
 - Given UIDs, only Retrieve Service AETitle or DICOMweb origin server address is needed to get objects
- For filesystem access, objects and sets of objects are identified by pathnames
 - If pathnames are [derived from] UIDs, then only partition mount point is needed to create pathname to object; eliminates redundancy of UID and pathname in inventory
 - E.g., <mountpoint>/<studyUID>/<seriesUID>/<instanceUID>.dcm
 - [Alternative – pathnames are not derived from UIDs, and explicit pathname must be included in inventory for each object]
- **Proposal: inventory shall allow use of canonical pathnames for stored objects**

One member totally opposed for 2 reasons:

1. Many PACS have non unique things in them

2. Files are stored the way they are stored and it's barely tolerable to have the archive return access path to files where they are now and over the top to expect them to provide some type of link
3. From policy reason – discourage it – opening the door to going away from DIMSE and DICOMweb

Explicit URI to the file

Need a fully qualified path

Can then direct access URIs be proprietary? – Yes

Vendor can provide the URL but it's up to the migrator to say yes, I can make use of it.

What protocol would one use to access those? Are there protocols used to access things in the previous generation object stores?

Either you go through the front door which is DICOM or DIMSEweb or you go through the back door, which is -we give you a URL and good luck with it.

Premise of this WG is to provide file-based access rather than going through DICOMweb or DIMSE for efficiency reasons. If they are actually files on a file system, they may be accessible via SMB or NFS which are if not standard than at least relatively widely used such as everybody can support them and then as an intermediate category would require less well know or vendor specific protocol. Can these direct access URLs be proprietary?

Yes

Are we going to enumerate in the supplement those that we recognize as well known? Yes

Communication of identity and access control are enabled but not necessarily within the scope of our document. Not standardizing any of the authorization controls.

Provide a schema?

Identify the version of the API you are using.

Define schemes for the most common ones in the field today.

17

Open issue – Deprecated objects (IOCM)

- Some SOP Instances may be identified as not for clinical use
 - Including images rejected for quality or patient safety reasons through IHE Imaging Object Change Management (KO objects with specific title concept codes)
 - But SOP Instances may be required to be archived for regulatory reasons (e.g., patient X-ray exposure)
- Such SOP Instances might not appear in normal C-FIND responses, but should be included in inventory
- **Proposal : inventory should include SOP Instances identified as not for clinical use, with specific flag attribute; inventory should include IOCM KO objects with title concept codes**
 - Up to the receiving app (migration manager) to determine what to do about them – different organizations may have different policies

This is something that should be parameterized.

The proposed would work as long as that's what the PACS actually stored. In many instances PACS delete bad stuff. It wouldn't be able to satisfy this requirement. Have to accept the range of different archive behaviors. The inventory is what the PACS have, if it's deleted, then it doesn't have it. It also imposes a high burden on the recipient to process it correctly. How it processes it's up to the policy of the institution.

IOCM has 4 different modes: one is on delete and other three are hiding – and only 1 of those 3 has the behavior to get only the hidden instances. Not only a delete request but also a hide request.

We need to really consider what the primary use case is here? Do people really expect to migrate an audit trail and all the changes?
It's customer specific. Sometimes they want a history of changes, but other times happy with a snapshot.
How do u migrate currently the notion of changes made through the management interface into the new PACS?
Migrate the current state of the data which includes any metadata overrides, and then the audit trail is handled separately, either imported into the new system as a separate entity or stored in a separate query-able place.

Legal requirements impose that you have to maintain stuff (images), but not that you keep the record of changes.

Changes through the user interface or changes through IOCM should manifest themselves in our migration use case in the same way – out of scope and handled in the audit trail.

5 OLD BUSINESS

6 NEW BUSINESS

7 DATE AND TIME OF NEXT MEETINGS (Secretariat)

- Continue T-con meetings bi-weekly for the time being (at least until the New Work Item Proposal is submitted to DSC)
- Next call is August 19, 2020 between 11:00 am and 12:30 pm ET

<u>NEMALINK CODE</u>	09-WG33
<u>SUBMITTED BY</u>	Hull, Carolyn
<u>SUBMITTED ON</u>	8/12/2020
<u>LEGAL REVIEW</u>	Peter Tolsdorf, 8/14/2020
<u>UPLOAD LOCATION</u>	Enter upload location.