Integrating the Healthcare Enterprise

IHE IT Infrastructure (ITI)

Technical Framework Supplement

Cross-Enterprise Document Workflow (XDW)

Trial Implementation

Date: September 20, 2013

Author: IHE ITI Technical Committee

Email: iti@ihe.net

Foreword

This is a supplement to the IHE IT Infrastructure Technical Framework V10.0. Each supplement undergoes a process of public comment and trial implementation before being incorporated into the volumes of the Technical Frameworks.

This supplement is published on September 20, 2013 for Trial Implementation and may be available for testing at subsequent IHE Connectathons. The supplement may be amended based on the results of testing. Following successful testing it will be incorporated into the IT Infrastructure Technical Framework. Comments are invited and may be submitted at http://www.ihe.net/ITI_Public_Comments.

This supplement describes changes to the existing technical framework documents.

"Boxed" instructions like the sample below indicate to the Volume Editor how to integrate the relevant section(s) into the relevant Technical Framework volume.

Amend section X.X by the following:

Where the amendment adds text, make the added text **bold underline**. Where the amendment removes text, make the removed text **~~bold strikethrough~~**. When entire new sections are added, introduce with editor's instructions to "add new text" or similar, which for readability are not bolded or underlined.

General information about IHE can be found at: http://www.ihe.net.

Information about the IHE IT Infrastructure domain can be found at: http://www.ihe.net/IHE_Domains.

Information about the structure of IHE Technical Frameworks and Supplements can be found at: http://www.ihe.net/IHE_Process and http://www.ihe.net/Profiles.

The current version of the IHE Technical Framework can be found at: http://www.ihe.net/Technical_Frameworks.

CONTENTS

Introduction

The Cross-Enterprise Document Workflow (XDW) profile enables participants in a multi-organizational environment to manage and track the tasks related to patient-centric workflows as the systems hosting workflow management applications coordinate their activities for the health professionals and patients they support. XDW builds upon the sharing of health documents provided by other IHE profiles such as XDS, adding the means to associate documents conveying clinical facts to a patient-specific workflow. XDW provides a common interoperability infrastructure upon which a wide range of specific workflow definitions may be supported. It is designed to support the complexity of health services delivery with flexibility to adapt as workflows evolve.

This profile defines an instrument, called a "Workflow Document", to manage and track a shared workflow. It records the creation of tasks and maintains a historical record of tasks as they move through the associated workflow. The Workflow Document also maintains the references to health information input and output associated with each task. Such shared workflow status information allows the various participating systems to coordinate their actions by:

- being aware of the history of a workflow for a patient;
- obtaining and reading the workflow's incomplete tasks;
- updating this shared document as the workflow tasks are performed according to a referenced Workflow Definition.

XDW provides to offer a common, workflow-independent interoperability infrastructure that:

- provides a platform upon which a wide range of specific workflows can be defined with minimal specification and application implementation efforts on the workflow definition (e.g., Medical Referrals Workflow, Prescriptions Workflow, Home Care Workflow);
- benefits many clinical and non-clinical domains by avoiding different competing approaches to workflow management;
- increases the consistency of workflow interoperability, and enables the development of interoperable workflow management applications where workflow-specific customization is minimized;
- facilitates the integration of multi-organizational workflows with the variety of existing workflow management systems used within the participating organizations;
- offers the necessary flexibility to support a large variety of different healthcare workflows by not being overly constrained.

More specifically XDW supports workflows that are:

- patient-centric;

- based on business/clinical needs that are defined externally to the XDW Profile. Such workflow definitions have to be known only by the applications within the participating systems, not by the XDW infrastructure (flexibility);
- executed in loosely connected, distributed environments, where centralized workflow management systems are not desired, or in many instances, possible.

The XDW Workflow Architecture illustration (figure 30-1) shows how the sharing of XDW Documents between "edge" applications using Document Sharing infrastructure supports the management of Workflow according to Workflow Definitions established between participating applications.

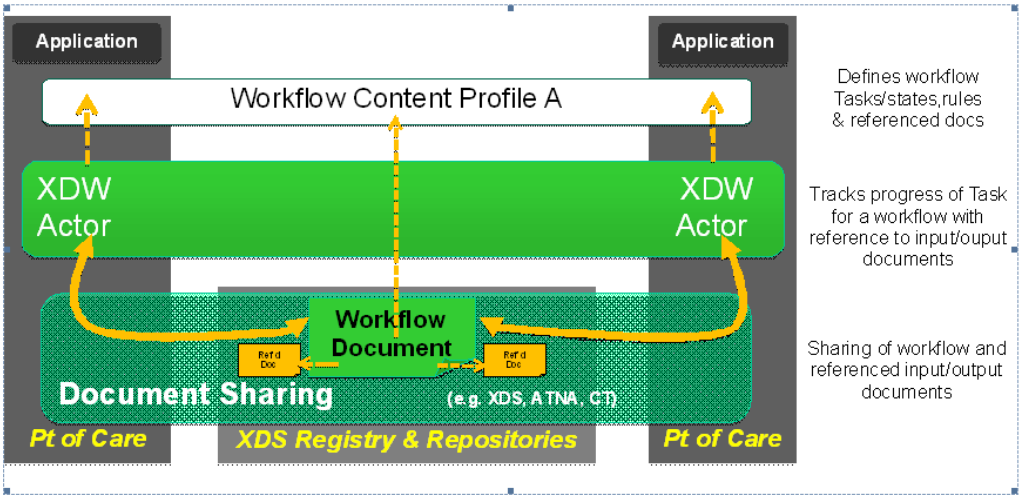

Figure 1-1: XDW Workflow Architecture

Open Issues and Questions

1. *The data structures used to describe tasks in XDW have been based on the OASIS Human Task standard. XDW has been designed to support workflow definitions that are specified externally to the XDW profile (layer above). Workflow definitions may be documented using informal forms such as textual definitions, as well as formal computable forms. It is expected that BPEL and/or BPMN will be useful standards supporting workflow definitions are introduced. It is recognized that OMG with BPMN 2.0 has recently introduced data structure for "user tasks". The trial implementation version of the XDW profile has not chosen the data structure of BPMN 2.0 user task, as it found out that it is a functional subset of Human Task. As the experience with XDW grows and the use of Human Task and BPMN2.0 matures in the general IT market, it should be expected that IHE will refine the requirements for computable Workflow Definitions in health, and consider the need for evolutions of XDW Workflow Tasks in its use of the OASIS or OMG family of workflow standards.*

Closed Issues

1. *Should there be any clinical information inside the Workflow Document? No, XDW as specified requires that all the Clinical information be included by reference to external documents. (figure 1). This ensures that a generic Workflow Document structure is defined, with only a few attributes customized for the workflow specifics (e.g., workflow definition IDs, workflow description, task codes, task description, possible referenced documents specification, etc.).*

2. *It was decided to include the management of the status of the task that is "task in the future". Completed Tasks are a record of past up the most recent activity that has been performed. Future activities could be introduced inside the Workflow document as created, or in-progress tasks, as well as supporting documents referenced as outputs (e.g., orders, requests, treatment plans, etc.). This reflects the reality that in a multi-organizational environment future tasks are an "expectation" shared by one professional with others that will rely upon their medical judgment and the latest information to perform or not such expected activities within the constraints of the workflow description.*

3. *The same document can be referenced in the input or output of one or more task*

4. *This profile specifies no rules for controlling the succession of tasks except through the reference to a workflow definition as specified by the Workflow Document typecode and description.*

5. *IHE-ITI specifies a classCode for XDW Workflow Document.*

6. *This profile is specified to be supported with XDS, XDM and XDR; it is left to the future to specify application of the profile in XCA (Cross-Community).*

7. *Should we be more specific in term of kinds of queries have to be supported or implemented by a Document Consumer in XDW? Show all workflow for patient, show me all open workflow, I've a Workflow Document Id give me the approved one (walking through replace associations or with folder or how?)*

   A section has been added to describe query strategies in Volume 3.

8. *A step may refer to another Workflow Document as input or output. If referred as output, this implies that new workflow document for a different workflow was created. However, to have a task refer to another specific task in the same WD as inputs or outputs has not been included.*

9. *XDW does not explain how to define a notification system about the change (replacement). Should an appendix be developed to discuss?*

   Because XDW relies on the XDS family of profiles, introducing notification mechanism such as DSUB or NAV or a matter of combining existing profiles. The possibility is mentioned as a note on the volume 1 use case.

10. *In a Workflow Document, the reference to an input or output document is a homecommunity ID and a documentID. It seems sufficient.*

11. *Appropriateness of the use of FolderID for referencing other Workflow Documents. (To reference a Workflow Document that is inside another Workflow Document benefits from the use of the FolderID of the Workflow Document. This avoids referencing a specific Workflow Document that may have been deprecated due to further tasks been added.)*

    Closed. The use of the folder has be adopted and explained in Volume 3.

12. *Appropriateness of the use of Folders for managing back links from documents to Workflow Documents. Should this back link be mandatory or optional? (A performance improvement mechanism to find the Workflow Documents referencing a specific clinical document has not been included. An approach to have any referenced documents placed in the same folder as the Workflow Documents has been considered but not included. This Folder mechanism could have offered a simpler back-link within an XDS affinity domain. However it does not scale up to document that may be referenced in a multi-community environment (use of XCA) as it is expected that XDW would be extended in the future to multi-communities.)*

    The use of the backlink has been left as an implementation choice; the use of the folder has been suggested only for the use cases based on an XDS infrastructure.

13. *Definition of the information that will be in the header: open/close, creator, uniqueID,*

    Closed and defined in Volume 3.

14. *Management of the closing of a workflow to avoid inactive Workflow Document being returned by queries. This introduces some form of state of the workflow. However, as XDW does not want to define a mechanisms to define overall workflow status (this may be done through the definition of specific tasks by the workflow specific specifications built on top of XDW. To make that point clear, it is proposed to only introduce a "workflow active flag", which may be valued either as: "open" or "close" to make clear that such a mechanism is quite limited. It is proposed to place this "workflow active flag" in the document metadata "event code list". It is proposed to not duplicate this workflow active flag in the workflow document header, but to simply introduce a copy of the flag indication in the workflow task data that resulted in modifying the "workflow active flag".*

    Agreed.

15. *A framework for specifying XDW-Based Workflow Definition Profiles is proposed in the form of an example. The XDW Document Content specification is sufficiently generic to support without customizations in its structure (no new data elements may be added). It is sufficient to constrain already defined data elements in the XDW Workflow Document such as defining: (1) a set of Task Name codes and Display names allowed, (2) succession rules/constraints between these steps, (3) the referenced input and output document content specifications (e.g., IHE PCC, Pharmacy, Laboratory, QRPH Content*

profiles), (3) the ability to span "sub-workflows" by creating new Workflow Documents, etc. This approach enables the development of more generic XDW Workflow management applications.

A workflow definition for a basic unstructured workflow has been proposed in an appendix. Later a guide for documenting workflow definition profiles will be developed as handbook.

16. *This proposal uses the CDA as formal structure of the Workflow Document. Is it appropriate? If not propose a detailed alternative.*

    No. The use of Human task was preferred.

17. *This proposal does not yet specify templates, value set if any, codes, etc. these will be defined, please comment.*

    A workflow definition for a basic unstructured workflow has been proposed in an appendix. Later a guide for documenting workflow definition profiles will be developed as handbook.

Volume 1 – Integration Profiles

1.7 History of Annual Changes

Add the following bullet to the end of the bullet list in section 1.7

30. **Cross-Enterprise Document Workflow (XDW)** – enables participants in a multi-organization environment to track the tasks related to patient-centric workflows as activities are coordinated.

For the information of reviewers, the following permission is already part of the framework.

1.10 Copyright Permission

Modify section 1.10 adding the copyright rights about OASIS

WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/who/trademark.php for above guidance.

Add Section 2.2.30

2.2.30 Cross-Enterprise Document Workflow Content Profile

The Cross-Enterprise Document Workflow (XDW) profile enables participants in a multi-organization environment to manage and track the tasks related to patient-centric workflows as the systems hosting workflow management applications coordinate their activities for the health professionals and patients they support. XDW builds upon the sharing of health documents provided by other IHE profiles such as XDS, adding the means to associate documents conveying clinical facts to a patient-specific workflow. XDW provides a common

interoperability infrastructure upon which a wide range of specific workflow definitions may be supported. It is designed to support the complexity of health services delivery with much flexibility to adapt as workflows evolve.

Add Section 30

30 Cross-Enterprise Document Workflow Content Profile

The Cross-Enterprise Document Workflow (XDW) profile enables participants in a multi-organizational environment to manage and track the tasks related to patient-centric workflows as the systems hosting workflow management applications coordinate their activities for the health professionals and patients they support. XDW builds upon the sharing of health documents provided by other IHE profiles such as XDS, adding the means to associate documents conveying clinical facts to a patient-specific workflow. XDW provides a common interoperability infrastructure upon which a wide range of specific workflow definitions may be supported. It is designed to support the complexity of health services delivery with flexibility to adapt as workflows evolve.

This profile defines an instrument, called a "Workflow Document", to manage and track a shared workflow. It records the creation of tasks and maintains a historical record of tasks as they move through the associated workflow. The Workflow Document also maintains the references to health information input and output associated with each task. Such shared workflow status information allows the various participating systems to coordinate by:

- being aware of the history of a workflow for a patient;
- obtaining and reading the workflow's incomplete tasks;
- updating this shared document as the workflow tasks are performed according to a referenced workflow definition.

XDW provides to offer a common, workflow-independent interoperability infrastructure that:

- provides a platform upon which a wide range of specific workflows can be defined with minimal specification and applications implementation efforts on the workflow definition (e.g., Medical Referrals Workflow, Prescriptions Workflow, Home Care Workflow);
- benefits many clinical and non-clinical domains by avoiding different competing approaches to workflow management;
- increases the consistency of workflow interoperability, and enables the development of interoperable workflow management applications where workflow-specific customization is minimized;
- facilitates the integration of multi-organizational workflows with the variety of existing workflow management systems used within the participating organizations;

- offers the necessary flexibility to support a large variety of different healthcare workflows by not being overly constrained.

More specifically XDW supports workflows that are:

- patient-centric;
- based on business/clinical needs which are defined externally to the XDW Profile. Such workflow definitions have to be known only by the applications within the participating systems, not by the infrastructure systems;
- executed in loosely connected, distributed environments, where centralized workflow management systems are not desired, or in many instances, possible.

The XDW Workflow Architecture illustration (figure 30-1) shows how the sharing of XDW Documents between "edge" applications using Document Sharing infrastructure supports the management of Workflow according to Workflow Definitions established between participating applications.

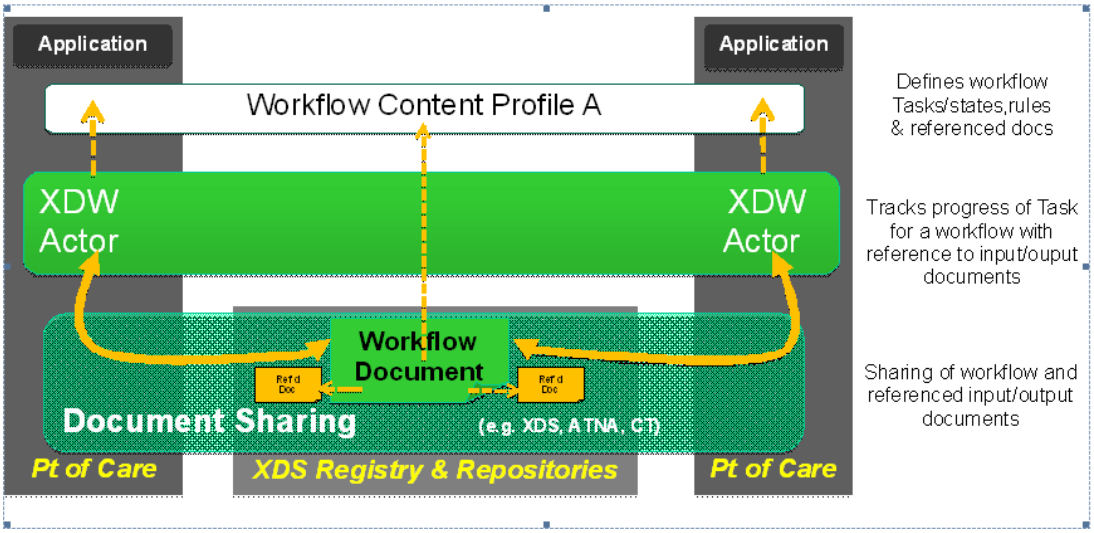

Figure 30-1: XDW Workflow Architecture

30.1 Actors/ Transactions

The XDW Content Profile is based on three actors, the Content Creator, the Content Consumer and the Content Updater. Content is created by a Content Creator or a Content Updater and is to be consumed by a Content Consumer or a Content Updater. The sharing or transmission of content or updates from one actor to the other is addressed by the use of IHE Integration profiles such as XDS, XDM or XDR (see PCC-TF-1: 2.1 for a detailed explanation of the use of "Content Profiles" with "Integration Profiles").

Figure 30.1-1 shows the actors directly involved in the XDW Profile and the direction that the content is exchanged.

A product implementation using this profile must group actors from this profile with actors from a workflow or transport profile to be functional. See Section 30.3 “XDW Actor Groupings and Profile Interactions”.

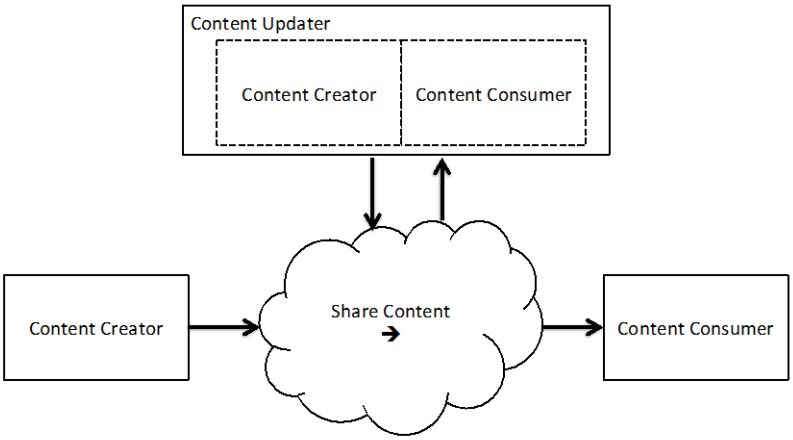

Figure 30.1-1: XDW Actor Diagram

Table 30.1-1 lists the content module(s) defined in the XDW Profile. To claim support of this profile, an actor shall support all required content modules (labeled “R”) and may support optional content modules (labeled “O”).

Table 30.1-1: XDW Profile - Actors and Content Modules

Actors	Content Modules	Optionality	Reference
XDW Content Creator	XDW Workflow Content Module (see Note 1)	R	ITI TF-3: 5.4
XDW Content Consumer	XDW Workflow Content Module (see Note 1)	R	ITI TF-3: 5.4
XDW Content Updater	XDW Workflow Content Module (see Note 1)	R	ITI TF-3: 5.4

Note 1: The XDW Workflow Content Module defines how to create an agnostic unstructured Workflow Document. Implementations may also choose to support Content Modules for specific workflows defined by IHE in workflow definition profiles (e.g., profiles in the PCC domain: Cross-Enterprise eReferral Workflow Definition (XBeR-WD), Cross-Enterprise TeleHomeMonitoring Workflow Definition (XTHM-WD), Cross-Enterprise Tumor Board Workflow Definition (XTB-WD, etc.).

30.1.1 XDW Content Creator

The Content Creator is responsible for creating content that will be shared or exchanged between other IHE Actors. It is required to be grouped with other Actors that perform the actual sharing or exchanging of information (see Section 30.3). The XDW Content Creator shall be able to create new workflows by creating a new XDW Workflow Document as defined in ITI TF-3:5.4. This actor is workflow agnostic and it is responsible only for the creation of the first version of the XDW Workflow Document.

30.1.2 XDW Content Consumer

The Content Consumer is responsible for accessing XDW Workflow Documents that have been shared or exchanged between other IHE Actors. It is required to be grouped with other Actors that perform the actual sharing or exchanging of information (see Section 30.3. The XDW Content Consumer actor may only obtain and read the last version of a specific XDW Workflow Document. The XDW Workflow Document consumed can belong to any kind of clinical workflow.

30.1.3 XDW Content Updater

A Content Updater shall be able to contribute to existing workflows by consuming an existing XDW Workflow Document and replacing it with an updated Workflow Document. It is required to be grouped with other Actors that perform the actual sharing or exchanging of information (see Section 30.3). This actor shall be able to consume and read the most recent version of a specific XDW Workflow Document. The XDW Content Updater shall be able to update the XDW Workflow Document, acting on the content in many different ways (tracking a new task initiated or performed, changing the status of tasks, adding documents reference in some tasks, changing the status of the whole workflow, etc.). After the update, the XDW Content Updater shall be able to replace the previous version of the XDW Workflow Document with the new one. This actor shall be able to solve “race condition” events (see section ITI TF-3:5.4.5.1).

30.2 Cross-Enterprise Document Workflow Profile Options

Options that may be selected for this Profile are listed in the table 30.2-1 along with the Actors to which they apply.

Table 30.2-1: XDW - Actors and Options

Actor	Options	Vol. & Section
Content Creator	*No options defined*	- -
Content Consumer	*View Option (Note 1)*	ITI TF-1: 30.2.1
	Document Import Option (Note 1)	ITI TF-1: 30.2.2
Content Updater	*View Option (Note 1)*	ITI TF-1: 30.2.1
	Document Import Option (Note 1)	ITI TF-1: 30.2.2

Note 1: The Actor shall support at least one of these options

30.2.1 View Option

A Content Consumer or a Content Updater that supports the View Option shall be able to:

- use the appropriate XD* transactions to obtain the Workflow Document along with associated necessary metadata;
- interpret the content of the Workflow Document and display its required content elements in a way which shows the tasks that are not complete and the completed task in a chronological way. The required elements to display are identified in the "View" column in ITI TF-3: table 5.4.3-8 and table 5.4.3-9.
- For each task, it shall list the documents referenced inside the Workflow Document and may optionally support the retrieve and the rendering of the documents referenced inside the Workflow Document.
- Any additional display capabilities that are specific to the referenced Workflow Definition profile may be provided.

30.2.2 Document Import Option

A Content Consumer or a Content Updater that supports the Document Import Option shall support the storage of the entire Workflow Document (as provided by the XD* sharing framework) along with applicable metadata to ensure its later processing. Documents referenced in the Workflow Document may also be stored. This Option requires the proper tracking of the relation between the Documents referenced and the content of the Workflow Document origin. Once a document has been imported, the Content Consumer or the Content Updater shall offer a means to use the document without the need to retrieve it again from the XD* sharing framework. When viewed after it was imported, a Content Consumer and/or a Content Updater may choose to access the XD* sharing framework to find out if the related Document viewed has been deprecated or replaced.

Note: For example, when using XDS, a Content Consumer may choose to query the Document Registry about a document previously imported in order to find out if this previously imported document may have been replaced

30.3 XDW Actor Grouping and Profile Interactions

A XDW Content Creator, Content Updater and Content Consumer shall be grouped with appropriate actors from the XDS, XDM and XDR profile to exchange XDW Workflow Documents. The metadata used for document entries in document sharing or interchange has specific relationships or dependencies (which we call bindings, see ITI TF-3:5.4.6) to the content of the clinical document – a XDW Workflow Document.

When XDW is used in conjunction with XDS:

- an XDW Content Creator shall be grouped with
  - an XDS Document Source with the Folder Management Options;

- an XDW Content Updater shall be grouped with
  - an XDS Document Source with both the Document Replacement Option and the Folder Management Options;
  - an XDS Document Consumer;
- an XDW Content Consumer shall be grouped with
  - an XDS Document Consumer.

When XDW is used in conjunction with XDR:

- an XDW Content Creator shall be grouped with
  - an XDR Document Source;
- an XDW Content Updater shall be grouped with
  - an XDR Document Source;
  - an XDR Document Recipient;
- an XDW Content Consumer shall be grouped with
  - an XDR Document Recipient.

When XDW is used in conjunction with XDM:

- an XDW Content Creator shall be grouped with
  - an XDM Portable Media Creator;
- an XDW Content Updater shall be grouped with
  - an XDM Portable Media Creator;
  - an XDM Portable Media Importer;
- an XDW Content Consumer shall be grouped with
  - an XDM Portable Media Importer.

Note: The support of Workflow spanning XDS, XDR and XDM environments is not explicitly addressed.

30.4 XDW Process Flow

30.4.1 XDW Approach to Workflow

XDW is a core component of a common, workflow-independent interoperability infrastructure that provides a platform upon which a wide range of specific workflows can be defined by

"content specialization" with minimal specification and implementation efforts (e.g., Medical Referrals, Prescriptions, Home Care).

This section first describes the overall architecture within which the XDW profile operates. Next, the structure of the XDW workflow document, the primary data structure that is shared among the workflow participants, is described.

30.4.1.1 XDW Workflow Architecture

A **Workflow Definition** is structured as a set of logical or clinical tasks definitions and rules. Each task definition describes an activity or a group of activities that needs to be accomplished by the owner of the task. The rules in the workflow definition ensure that the different participants in a workflow operate jointly to advance within process and to move from one task to another in a consistent way.

Figure 30.4.1.1-1 presents an overview of the Workflow Architecture built around the XDW Profile.

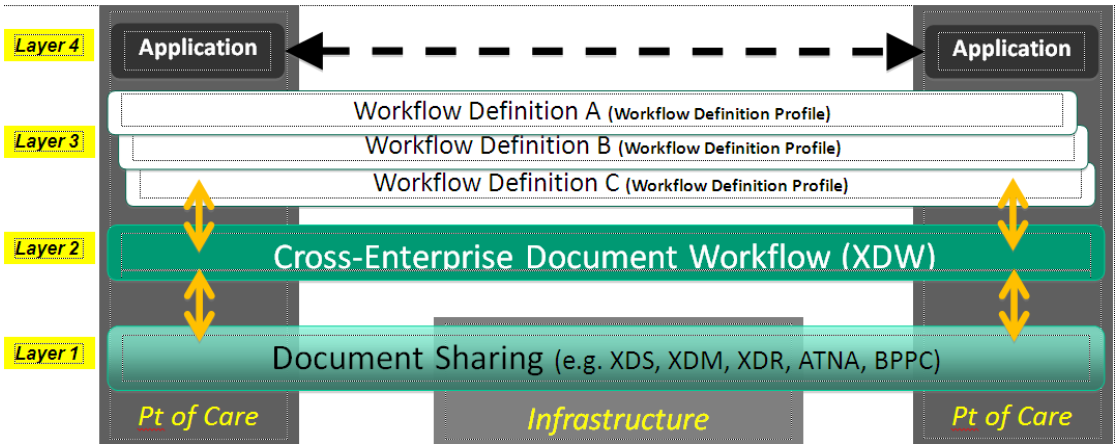

Figure 30.4.1.1-1: XDW Architecture Overview

In this workflow architecture:

- The first layer supports the sharing or exchange of documents. This interoperability foundation is enabled by a set of existing IHE document sharing profiles such as XDS, XDR and XDM along with document content profiles and security/privacy profiles such as ATNA and (optionally) BPPC;
- The second layer defines a generic data structure called a Workflow Document which is shared among the workflow participants by using the first layer of this architecture. Likewise, the clinical and administrative documents that are used as input and produced as output by the tasks of workflows managed by the XDW profile are shared using the same first layer of this architecture;

- The third layer introduces the semantic definition of the workflows that can be understood and executed among the participating systems/applications. The orchestration of specific workflows allows the workflow participants to share a common understanding of the specific tasks, the dependencies between these tasks, and a number of rules that control the workflow execution. Execution details are conveyed through the XDW Workflow Document defined by the second layer of the architecture. The specification of Workflow Definitions at this third layer is not part of the ITI XDW Profile and is currently best handled with a natural language expression (See example of Basic Unstructured Workflow Definition Profile, ITI TF-2x:Appendix X);
- The fourth layer of this architecture contains the applications executed by the participating systems. Such applications bridge between XDW managed workflow and the locally managed workflow. Much of the details of the local workflows managed by each application will be hidden and encapsulated in "higher" granularity tasks exposed through XDW; as such details would not need to be externally exposed. The workflow definitions conveyed by the third layer should only contain higher granularity tasks that require workflow coordination across organizational boundaries.

30.4.1.2 XDW Document Structure

The XDW Profile uses the XDW Workflow Document to manage workflows.

The XDW Workflow Document enables participants in a multi-organization environment to manage and track the execution of patient-centric workflows. The structure of WorkflowDocument is organized into Tasks and TaskEvents.

A **Task** describes an activity, or a group of activities, that need to be accomplished or have been accomplished. A Task is characterized by several attributes:

- the type of task,
- the owner of the task,
- the current status of this task (one of the status value that are valid for this task),
- the references to documents used for input or produced as output
- the history of past **Task Events** for this task, that document the progress of the task up to the present state

When a person or organization has been assigned as owner of a task, the task is placed under execution. (It moves from a "CREATED" or "READY" status to an "IN_PROGRESS" status). When the expected activity(ies) is completed successfully the task moves to the "COMPLETED" status, otherwise to the "FAILED" status (for the state diagram see ITI TF 3: 5.4.2.4).

Task Event is a record of a change (status and/or other attribute) of a task; a Task Event history is the list of Task Events for a specific task.

As shown in the figure X.30.4.1.2-1, the XDW Workflow Document is structured into two parts:

- a first part with general workflow information about the document,
- a second part that collects the different Tasks that are completed or not yet completed in the workflow, as well as for each task, the related Task Events that tracks its progress. Task and Task Event specification leverages a proper subset of the task model and specification from the OASIS Human Task, a standard closely related to well-known workflow standards such as BPEL and BPMN.

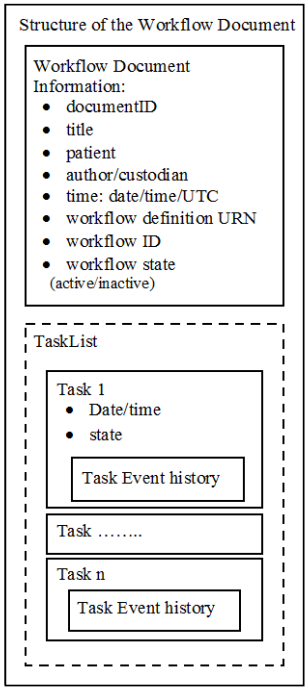

Figure 30.4.1.2-1: Workflow Document Structure

The Task and Task Events include references to clinical or administrative input/output documents to the Task or Task Event:

- The **Input attribute** contains references to documents that are relevant for workflow participants in performing the Task. For example, for a performed examination, this could contain a reference to a referral request. It may also contain references to "parent" workflows to which this workflow is a "child".
- The **Output attribute** contains references to documents that were produced as a result of performing this Task. For example, this could contain a reference to a report written by a specialist. It may also contain references to "child" workflows initiated by this workflow as a parent.

At any time, if a participant chooses to update the workflow for a specific patient, it shall either create one (or more) new task or update an existing task and record a past taskEvent. Each

update to the Workflow document results in a new instance of the Workflow Document which is published as a replacement. The prior version being replaced is then placed in the status "deprecated" (XDSDocumentEntry availabilityStatus) so that only the newest Workflow Document is active. The technical description of the updating process of the Workflow Document is specified in ITI TF-3:5.4.5.4.

When a new Workflow Document is created it is placed in a **Workflow Context Folder**. All subsequent replacement workflow documents are also placed in the same folder so that the Workflow Context Folder uniqueId provides a stable reference to an instance of a workflow, while the Workflow Document uniqueId is different for each version of the workflow document.

Although, it is possible to place inside the same Workflow Context Folder of the Workflow Document all the referenced documents (clinical documents referenced as Input attribute or Output attribute inside the Workflow Document), there is no requirement to do so. It is up to each specific workflow definition to analyze the need and frequency to search for the list of workflows in which a specific document may be referenced. If such a "workflow back-link" is deemed necessary, it may be performed in an XDS environment by various approaches, such as:

- Query for Workflow Documents based on their metadata attributes (workflow active flag, date of service, document class, document type, document format, etc.) and by filtering the retrieved set of (non-deprecated) Workflow Documents for the referenced document Id across all steps within each Workflow Document. Note: This approach may require a larger number of transactions, but is extensible to a cross-community environment;
- By requiring that all documents referenced within a specific Workflow Document be placed within the folder that contains the Workflow Documents (all the documents referenced are part of the same XDS Affinity Domain). Note: this approach is constrained to be used with a single XDS affinity domain.

30.4.2 XDW Use-Cases and Process Flow in an XDS Affinity Domain

A broad range of use cases may be supported by the XDW Content Profile.

The purpose of this section is to describe a typical use of XDW with no intent to present the breadth and flexibility of XDW. The use case described in this section provides the necessary background to the reader in understanding the basic capabilities of XDW.

This use case is not intended as a Workflow Definition Profile specification. Such Profiles are being developed by clinical IHE Domains in order to support their specific workflows.

30.4.2.1 Referral Workflow Use Case

This workflow is a three step process:

A. a physician refers a patient to another healthcare provider for a specialist's consultation;

B. the specialist starts the consultation which may span one or more visits

C. the specialist completes the consultation and produces a report.

Each step will be described both from a clinical and a technical point of view.

The description will rely on two figures:

- Figure 30.4.2.1-1 represents the evolution of the Workflow Document during this Referral workflow. Each one of the three steps A, B, C is depicted in a column.
- Figure 30.4.2.1-2 is a sequence diagram of the transactions between "system actors" in the sharing of the Workflow Document as it is updated, using an infrastructure based on the XDS profile (although not shown here, this use case could be transposed on the XDR or XDM profiles).

30.4.2.1.1 Referral Workflow Use Case - Step by Step

We present below the detailed chronological sequence of steps:

A. A physician refers a patient to another healthcare provider for a specialist's consultation

In this task, the GP examines the patient and reviews the patient's most recent laboratory report. The GP refers the patient to a specialist, creating an eReferral Document and referencing the laboratory report.

The GP's software, as Content Creator, produces the e-Referral Document and one Workflow Document to track the clinical workflow of the eReferral. As shown in column A of figure X.4.2.1-1, at this moment the Workflow Document created has only one task ("Referral Requested") characterized by:

- a task status "COMPLETED"
- as inputs of the task the references to the laboratory report analyzed by the GP
- as outputs of the task the reference to the eReferral document produced.

In order to share the documents that are produced during the task, the GP's Software (as a grouped Content Creator and XDS Document Source) submits the eReferral Document and the Workflow Document to the XDS Document Repository as shown in box A of figure 30.4.2.1-2.

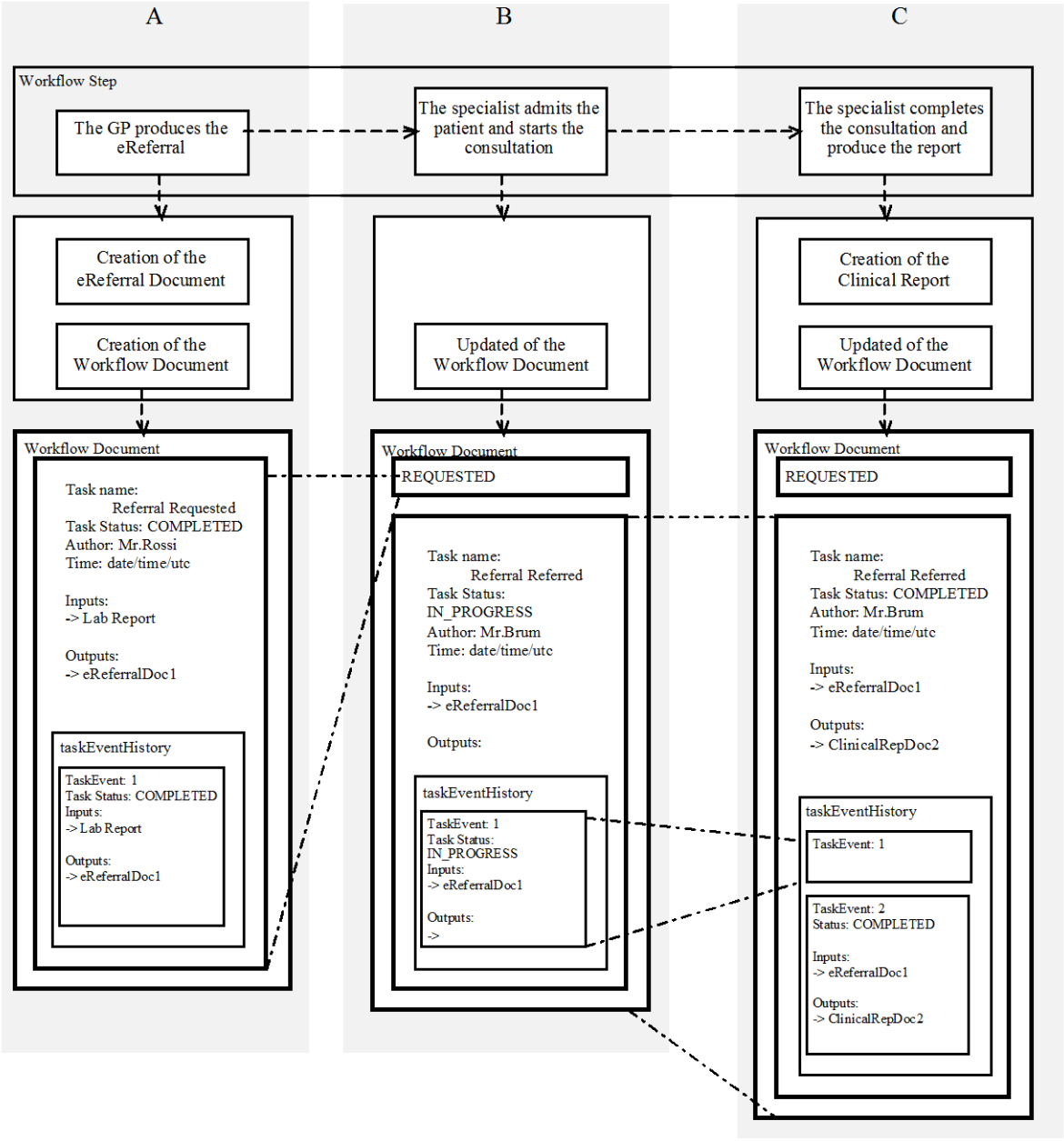

Figure 30.4.2.1.1-1: Management of the Workflow Document

650

B. The specialist starts the consultation which may span one or more visits

In this task, the patient goes to the specialist of his choice (or suggested by his GP).

The specialist consults the eReferral document and the associated Workflow Document to understand the task that needs to be performed.

The specialist accesses the document by using his software, which is a grouping of a Content Updater and an XDS Document Consumer, to query and retrieve the Workflow Document and the eReferral document, as shown in box B of figure 30.4.2.1-2.

If consistent with the Workflow Definition referenced in the Workflow Document, the specialist accepts the patient and updates the Workflow Document so that no other specialist may perform the consultation.

As shown in column B of figure 30.4.2.1-1, at this step of the workflow, the Workflow Document is updated with a new version in which a new task "Referral Referred" is added to the content of the previous version of the Workflow Document. The task "Referral Referred" is characterized by:

- a task status "IN_PROGRESS"
- as inputs of the task the references to the eReferral document produced by the GP

The Specialist's software, as a Content Updater and an XDS Document Source, provides the updated version of Workflow Document to the XDS Document Repository/Registry through a Replace of the previous version of the Workflow Document (see box B in figure 30.4.2.1-2).

C. **The specialist completes the consultation and produces a report**

The specialist ends the consultation and he produces a report of the consultation.

In this task, the software of the specialist, as a Content Updater, updates the Workflow Document changing the status of the "referred" task.

As shown in column C of the figure 30.4.2.1-1 the Workflow Document, the "Referral Referred" task is characterized by:

- a task status "COMPLETED"
- as inputs of the task the references to the eReferral document produced by the GP (the laboratory report was not used by the specialist)
- as output of the task the references to the report of the consultation

The history of the changes of the statuses of the task are tracked inside the task as a list called taskEventHistory.

The Specialist's software, as a Content Updater and Document Source, provides the updated version of Workflow Document to the Document Repository through a replace of the previous version of the Workflow Document (see box C in figure 30.4.2.1-2).

At any time the GP may review the Workflow Document and the new documents produced related to this workflow. This is accomplished through a query and retrieve by the GP's software of the active Workflow Document from the XDS Document Registry and the XDS Document Repository.

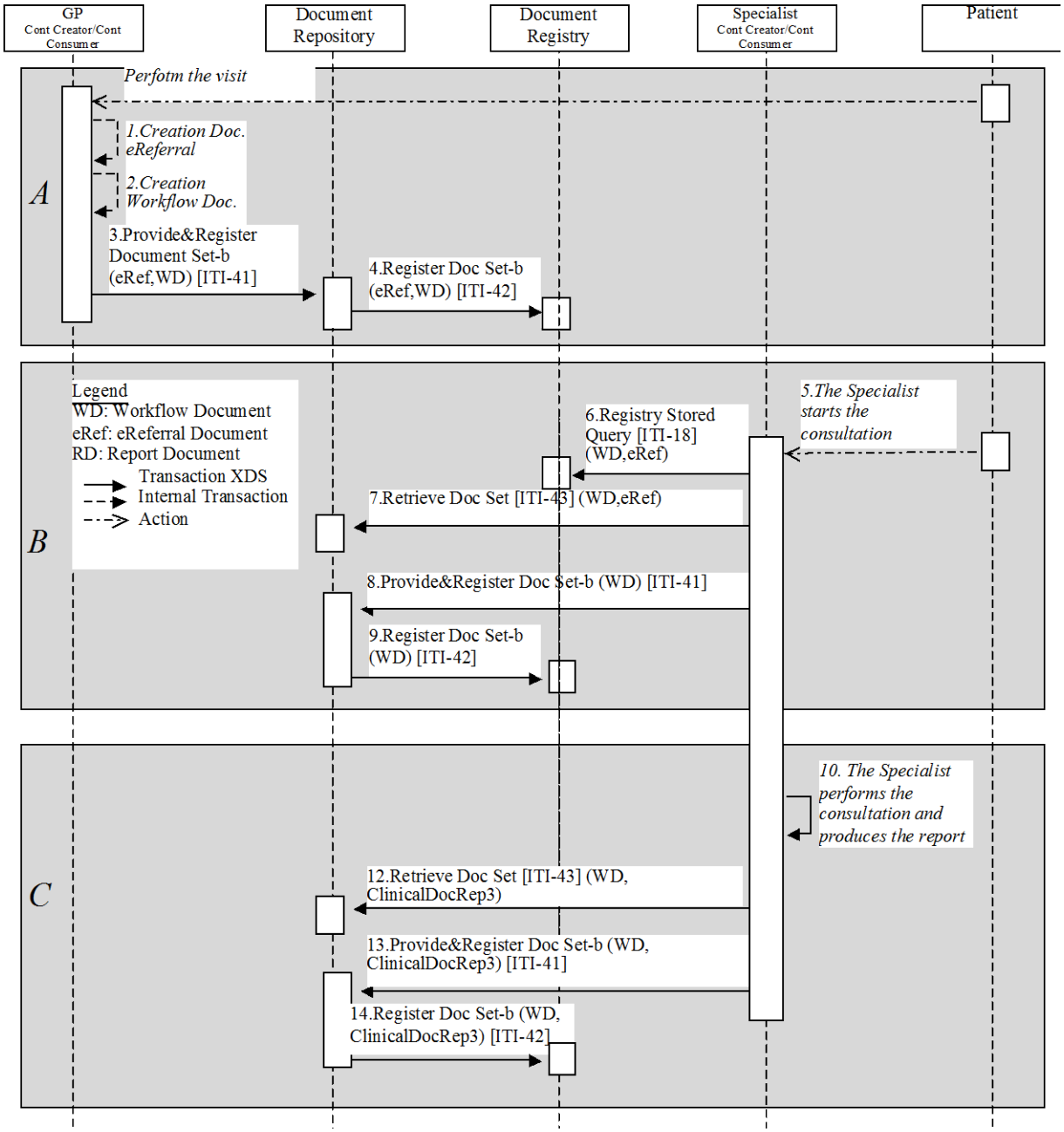

Figure 30.4.2.1.1-2: Basic Process Flow in XDW Profile, Simple Referral use case

Although not shown in this use case, it would also be possible to manage a system of subscription and notification to communicate the progress between the different steps through the use of the Document Metadata Subscription (DSUB) profile or the Notification of Document Availability (NAV) profile.

30.5 XDW Security Considerations

The XDW content profile relies on the security controls in the underlining transport (e.g., XDS). The XDW content is an administrative document that should not include clinical information but administrative information can be just as sensitive as clinical information.

The XDW Workflow Document will be authored by different organizations. As the document is updated the active version will be replaced with a newer version as the workflow progresses. However, with clinical documents it is not expected that organizations will replace documents authored by other organizations, as typically a clinical document comes from only one organization or individual. Therefore in order to adhere to the principle of least privilege organizations want to prevent clinical documents from being replaced by other organizations, while allowing XDW Workflow Documents to be replaced. It is recommended that organizations retain general restrictions on XDS documents, but make an exception for XDW Workflow Documents, based on classCode.

When a Workflow Description Profile is created a risk assessment following the Security Cookbook may result in additional security considerations beyond those for the usual clinical report.

The ITI XDW profile doesn't have a volume 2. It has one appendix in volume 2x which is at the end of this document.

Volume 3 – Cross-Transaction Specifications and Content Specification

Add section 5.4

5.4 XDW Workflow Content Module

This section defines the XDW Workflow Document by providing a schema and explaining its use. This document does not include clinical information about the patient directly. It shall only contain information necessary for organizing and defining work tasks. All clinical information regarding any task shall be provided through separate documents that are referenced from the associated input or output documents.

5.4.1 Referenced Standards

HL7 CDA Release 2.0 (denoted HL7 CDA R2, or just CDA, in subsequent text)

Web Services – Human Task (WS-HumanTask) Specification Version 1.1, OASIS

5.4.2 Discussion of Content Standards

The XDW Workflow Document is a document that incorporates elements from the HL7 CDA document structure and from the WS-HumanTask structure. The Workflow Document exists to coordinate the activities of multiple people in different organizations. They agree to share these documents as a method of exchanging work information. These documents are used by these organizations to feed what is often considered their own internal task management systems and have their own administrative rules for managing activities.

Sharing clinical documents is often accomplished as a normal part of providing healthcare. The XDW workflow allows the work information to be shared in the same way as other patient related clinical information. Integrating the internal workflow management systems of independent organizations with independent administrative rules, and perhaps in different legal and regulatory systems, is avoided.

The XDW Workflow Document does not contain clinical information about the patient. The input, output, and other elements of the task data shall contain references to documents (XDSdocumentId) that contain the clinical information.

XDW Workflow uses the XDS lifecycle management tools to coordinate updates to the Workflow Document instead of requiring an integration of all the different task management systems in the different organizations.

The XDW Workflow Document builds upon two other standards, HL7 CDA and OASIS WS-Human Task.

The XDW Workflow Document shall comply with the XDW XML Schema that includes elements from the CDA and OASIS Human Task standards. Access to the schema files from those standards will be needed.

The figure below represents the main level structure of the Workflow Document with the first level of the elements that composed the structure.

It is possible to divide the structured into four parts:

- Part 1: elements derived from HL7 CDA standard (Type of the element: CDA),
- Part 2: two elements, patient and author, defined in the XDWSchema with the structure derived from HL7 R-MIM standard (Type of the element: tXDWpatient and tXDWauthor),
- Part 3: elements defined by IHE XDW Profile
- Part 4: the element `<TaskList>` in which is defined by elements derived from the OASIS WS-HumanTask standard. In this last section the `<TaskList>` is a list of elements `<XDWTask>` composed of the HumanTask `<taskData>` (all data that define the XDWTask) and the HumanTask `<taskEventHistory>` that contains a list of elements `<taskEvent>`.

All the elements of the figure 5.4.2-1 are described in section 5.4.3.

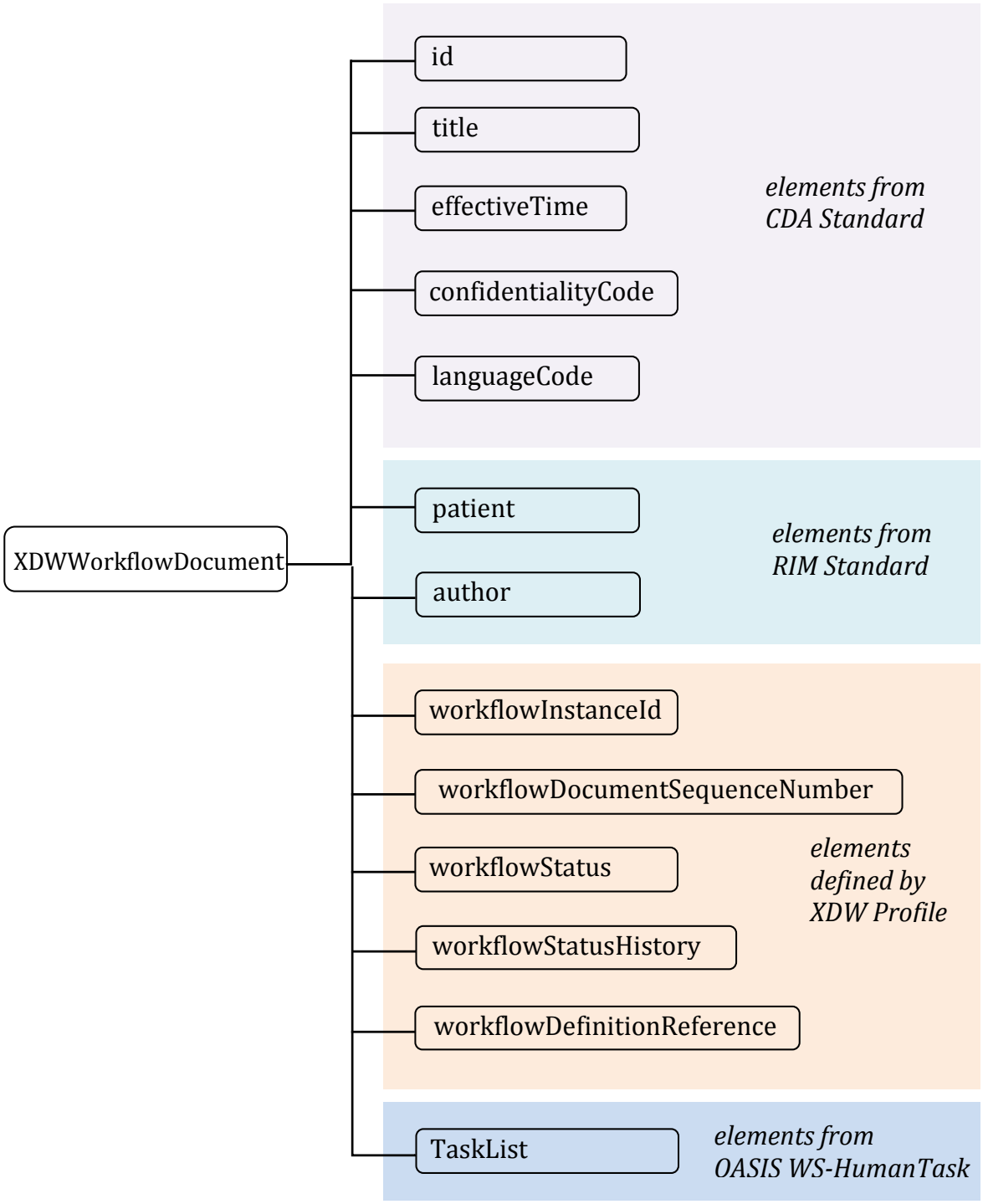

Figure 5.4.2-1: XDW.WorkflowDocument Structure

5.4.2.1 XDW Workflow Document Elements from HL7 CDA Standard

Some elements are incorporated directly from the HL7 CDA standard. This means that the elements, their definitions, and the rules for interpreting them are in the HL7 standard. These are summarized here for convenience.

`<patient>` and `<author>` elements have been defined based upon the HL7 CDA R-MIM. The XDW schema defines these elements using elements from CDA, and was derived by eliminating all elements that are not needed for workflow identification purposes. The R-MIM includes elements that are of clinical value. These have been removed for workflow use.

5.4.2.2 XDW Workflow Document Elements defined by IHE XDW Profile

The XDW Workflow Document also has elements that are defined by IHE (see table 5.4.3-1):

- `<workflowInstanceID>` Every version of the Workflow Document shall have the same workflowInstanceID value. It represents the Folder.uniqueId of the Workflow Content Folder in which the Workflow Document is contained. It shall be globally unique, because it is shared by many organizations.
- `<workflowDocumentSequenceNumber>` This is used to simplify management of the changes to the Workflow Document as the workflow is executed. It shall be created as "1", and be incremented for each update to the Workflow Document.
- `<workflowStatus>` This shall be either

  OPEN– which means that further updates are expected for this Workflow Document. These updates could be modifications to existing tasks or addition of new tasks or update to an existing task. Tasks shall not be deleted.
  CLOSED– which means that further updates to this Workflow Document are not expected. A workflow with a CLOSED workflowStatus may continue to be updated, after which the value of workflowStatus may be transitioned back to OPEN or remain CLOSED. These constraints will be defined by the Workflow Definition referenced.

- `<workflowStatusHistory>` This element represents the history of changes of status of the workflow document. It consists of sub-elements named documentEvent. Each documentEvent describes a change of status of the workflow document. In case that the workflowDefinitionReference describes a type of workflow that can't change its status from CLOSED to OPEN, the workflowStatusHistory contain at most two documentEvent elements, one for the opening of the workflow corresponding to the creation of the workflow document, and one to track the closing of the process related. Instead, if the workflowDefinitionReference permits the change of status from CLOSED to OPEN (e.g., OPEN-->CLOSED-->OPEN…) the element workflowStatusHistory will contain from 1 to N documentEvent elements to track these changes. A documentEvent is described by sub-elements defined in table 5.4.3-5.

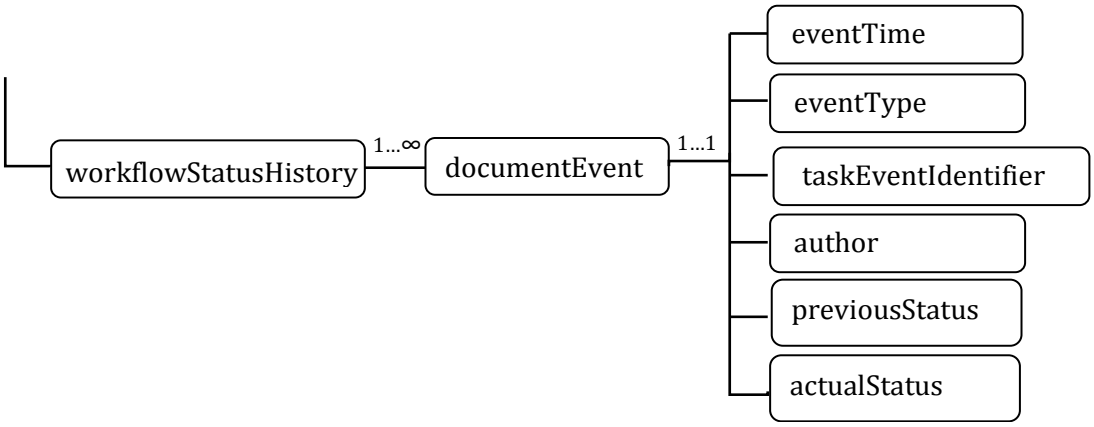

Figure 5.4.2.2-1: workflowStatusHistory Element

- `<workflowDefinitionReference>`. This is the reference to the workflow definition. This is usually contained in policy or procedure document or may be defined by IHE as a specific workflow definition profile. This profile places no restriction on the style used to document such Workflow definition. It is recommended to assign an OID to those. It shall be recorded by the creator of the initial Workflow Document and shall be preserved unchanged in all subsequent versions of the document.

5.4.2.3 XDW Workflow Document Elements from the OASIS Human Task

The descriptions of a task and of `<taskEvent>` are taken from the OASIS Human Task standard. This standard defines a way to describe a human task. It was defined as an extension to the BPEL and BPMN workflow standards. These standards are in use to manage the workflow of automated tasks under the control of an integrated task management system. It was recognized that while these standards do not have the ability to control human task, they needed a way to describe tasks to be performed by humans and other organizations.

The element `<XDWTask>` groups all information about one task in the workflow, the `<XDWTask>` is structured in two sub elements: `<taskData>` and `<taskEventHistory>.`

- `<taskData>` describes a single task. This is a list of details about the task, a description, the inputs to the task (e.g., documents), the outputs from the task (e.g., documents), fault descriptions and comments. The `<taskDetails>` include elements like the task ID, description, state, etc. (see table 5.4.3-8)

- `<taskEventHistory>` contains a list of the `<taskEvent>` elements that describe the changes of the task. For each task, there is one or more `<taskEvent>` that describes the history of the task. There is a list of the `<taskEvent>:` an `<eventType>`, a description, the inputs to the `<taskEvent>` (e.g., documents), the outputs from the `<taskEvent>` (e.g., documents), fault descriptions, comments, and

attachments (other documents that do not represent outputs). The details include elements like the task ID, status, etc. (see table 5.4.3-10)

The definitions and rules such as the state machine that defines status are in the Human Task standard. There are other datatypes and web services also defined in Human Task standard that are not used by XDW.

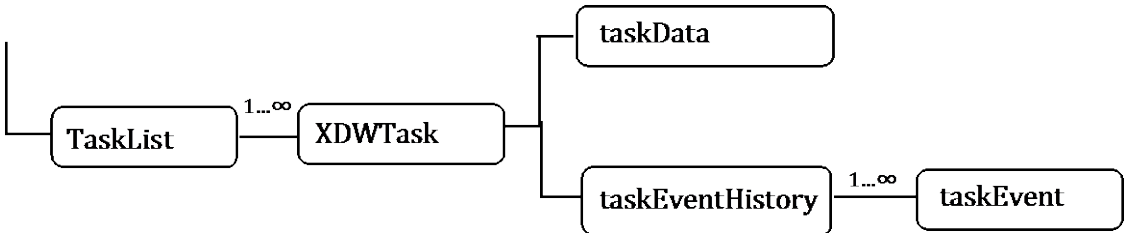

Figure 5.4.2.3-1: XDW Workflow elements derived from OASIS WS-HumanTask

5.4.2.4 Relationship between Task and <taskEvent>

When a Task is generated it has a first `<taskEvent>`. A Task can either have only one `<taskEvent>` if the status of the task is not modifiable and it is born just completed or it can have more status and so more taskEvents. In this case at any time the task changed a new `<taskEvent>` is created.

When a new Task is generated, zero or more references to external documents, associated with the Task, either as input or output, are put in the respective element of the Task. As a Task changes new input or output documents may be added (cumulative list of references). However, for each Task Event, only the input and output document related to the specific task Event shall be included. The inputs documents of a `<taskEvent>` are the documents that have been used as input for performing the Task change. The Output documents are those that have been created as a result of the Task Change. As a consequence, all input and output document references, present one or more times in the task Events list shall be listed (without duplicates) in the Task. Likewise for output document references.

The clinical documents referenced in the input or output data elements of Tasks and task Events shall be accessible in the affinity domain (if XDW is used along with XDS) or Media Interchange (if used along with XDM) or Point-to-point submission set (if used along with XDR). In anticipation of the use of XDW in a cross-community environment, both the document uniqueID and homeCommunityID are permitted to be included.

The XDW Workflow Document defines a task list which is a series of task descriptions. The relationship between the task, the order of the elements in this list and the possible status of a task, all this rules are defined in the Workflow Definition Document.

The XDW profiles define the recommended statuses processable in a Task with the `<taskEvent>`. These statuses are a subset of the HumanTask Standard. There are other task status values possible, but these are not normally used.

Table 5.4.2.4-1: Description of Task Status

Task status	Description
CREATED	The workflow is open, the task is created but not assigned to an owner
READY	The task created is assigned to an owner and is ready to be performed
IN_PROGRESS	The task is started and the owner is performing the task actions
FAILED	The task is completed with fault response (it is not possible conclude the action of the task)
COMPLETED	The task is completed with response

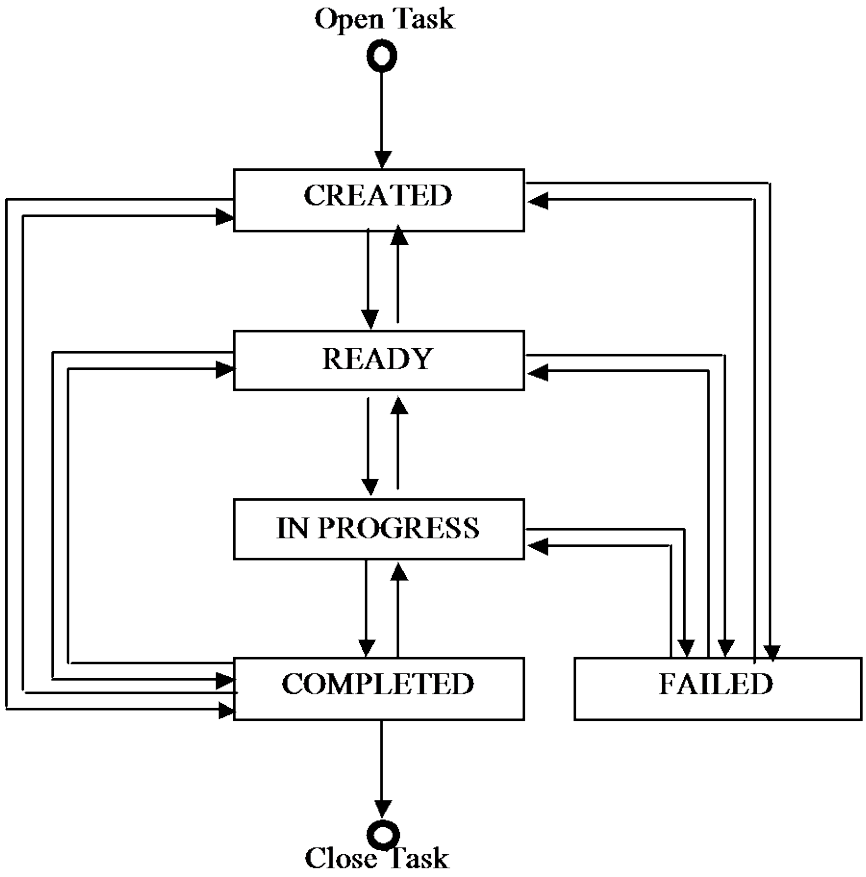

Figure 5.4.2.4-1: Task Status Transition

The element <XDWTask> and XDW <taskEvent> is constrained by XDW with a minimal set of elements required. These elements are fully extensible with any kind of attributes defined by Human Task standard. This allows specific Workflow Definition profiles to add elements defined in Human Task to manage for example intertask relationships, additional status, etc. to address more advanced specific workflow requirements.

5.4.3 Content Specification

The tables below represent all Workflow Document elements. The tables show for each element the Optionality and the standard from which the definition and the structure of the element derive.

Optionality:

R= element Required for XDW profile

R2= element Required if known for XDW profile

O= element Optional for XDW profile

X= element shall not be used

Inside the tables the column description is used to constrain the use of the attribute when referring to element defined in the underlining standard. When the description in blank no constrains is required. When the element is defined by XDW this is the complete description.

There are three functional roles for interacting with these elements.

- The "create" role specifies what elements shall be created. The Content Creator is permitted to include any optional element, and may include other elements.
- The "view" role specifies what elements shall be presented by Content Consumer or Content Updater that support viewing of the document. It may present for viewing any other element that it understands or has a means of presenting. There are elements that are required for viewing, while being optional for both creation and viewing.
- The "update" role specifies what elements shall be maintained with correct values when updating a document. An "update" operation shall preserve the value of all elements that are present, even if their meaning is unknown. This means that an updater might not update the contents of optional elements when updating a workflow document.

If one of the following tables does not specify separate values for the three roles, then the specified value applies to all three roles.

- `<XDW.WorkflowDocument>`

Table 5.4.3-1: Elements of the Workflow Document

XDW.WorkflowDocument element	Standard	Data Type	Optionality	Description
id	HL7 CDA	`cda:II`	R	Document ID
title	HL7 CDA	`cda:ST`	O	Displayable title
effectiveTime	HL7 CDA	`cda:st`	R	Time of most recent update
confidentialityCode	HL7 CDA	`cda:CE`	R	
languageCode	HL7 CDA	`cda:CS`	O	
patient	HL7 CDA	`tXDWpatient`	R	Patient information derived from R-MIM. Restricted to non-clinical necessary content. See table 5.4.3-2
author	HL7 CDA	`tXDWAu`	R	Author information

XDW.WorkflowDocument element	Standard	Data Type	Optionality	Description
		thor		derived from R-MIM. Restricted to non-clinical necessary content. See table 5.4.3-3
workflowInstanceId	IHE	xs:anyURI	R	
workflowDocumentSequenceNumber	IHE	xs:int	R	
workflowStatus	IHE	xs:token	R	OPEN if modifications are permitted to the document contents. CLOSED if modifications are not expected.
workflowStatusHistory	IHE	workflowStatusHistory_type	R	List of changes of the workflowStatus See table 5.4.3-4
workflowDefinitionReference	IHE	xs:anyURI	R	References (urn: OID:) to the documents that define this kind of workflow.
TaskList	OASIS_WS-HumanTask	TaskList_type	R	List of all tasks and their history See table 5.4.3-6

- <patient>

Table 5.4.3-2: Patient Element

Patient element	Standard	Data Type	Optionality	Description
id	HL7 CDA	cda:II	R	
name	HL7 CDA	cda:PN	O	
administrativeGenderCode	HL7 CDA	cda:CE	O	
birthTime	HL7 CDA	cda:TS	O	
martialStatusCode	HL7 CDA	cda:CE	O	

- `<author>`

  One or both of the following shall be present.

Table 5.4.3-3: Author Element

Author element	Standard	Data Type	Optionality	Definition
assignedAuthor	HL7 CDA	`cda:POCD_MT000040.AssignedAuthor`	R	Either assignedAuthoringDevice or assignedPerson should be specified

- `<workflowStatusHistory>`

Table 5.4.3-4: workflowStatusHistory Element

TaskList element	Standard	Data Type	Optionality	Description
documentEvent	IHE	`tXDWdocumentEvent_type`	R	A detailed event that represents a change of the workflowStatus The first documentEvent element is added when the workflow document is created. A documentEvent element is then added whenever the workflowStatus of the workflow document changes. See table 5.4.3-5

- `<documentEvent>`

Table 5.4.3-5: documentEvent Element

documentEvent element	Standard	Data Type	Optionality	Description
eventTime	OASIS_WS	`xs:dat`	R	Time when the

documentEvent element	Standard	Data Type	Optionality	Description
	- HumanTask	eTime		specific documentEvent element is added to the workflow document.
eventType	OASIS_WS - HumanTask	ht:tTaskEventType	R	The type of event that happens that solicits the modification of the workflowStatus. It should be valorized with one of these types: create, stop, suspend, resume, fail, complete. These types are defined in the HumanTask specification (C. WS-HumanTask Data Types Schema, <!-- Defines the human task event types -- >).
taskEventIdentifier	IHE	xs:anyURI	R	Element that permits to track the reference to the taskEvent that solicits the modification of the workflowStatus. It stores the same value of the element taskEvent/identifier of the taskEvent of reference.
author	IHE	xs:string	R	Actual owner of the workflow after the event
previousStatus	IHE	xs:token	R	The previous value of workflowStatus. Either “OPEN” or “CLOSED”. In case of a Workflow Document just created this element

documentEvent element	Standard	Data Type	Optionality	Description
				shall be valorized with “”
actualStatus	IHE	xs:token	R	Equal to the current value of the workflowStatus element. Either “OPEN” or “CLOSED”.

- <TaskList>

Table 5.4.3-6: TaskList Element

TaskList element	Standard	Data Type	Optionality	Description
XDWTask	OASIS_WS-HumanTask	tXDWTask	R	List of tasks See table 5.4.3-7

- <XDWTask>

Table 5.4.3-7: XDWTask Element

XDWTask element	Standard	Data Type	Optionality	Description
taskData	OASIS_WS-HumanTask	ht:tTaskInstanceData	R	Description of the current task (status, inputs, outputs, etc.) See table 5.4.3-8
taskEventHistory	OASIS_WS-HumanTask	tXDWeventHistory	R	History of the changes to the current task (dates, changes, etc.) See table 5.4.3-10

- <taskData>

The XDW profile adds the following restrictions to the OASIS definition for taskDetails:

- The taskData/input shall contain a taskData/input/part for every clinical document that is to be used as input to the task. This shall be an element of type tXDSregisteredDocument.

- The taskData/output shall contain a taskData/output/part for every clinical document that is created as a result of the task that is to be shared. This shall be an element of type tXDSregisteredDocument.
- Any clinical documents that are registered in an XDS Registry shall be identified in the taskData/input/part, taskData/output/part, or taskData/attachmentInfos/info by the documentID.
- The element type tXDSRegisteredDocument is a tAttachmentInfo that has XDSDocumentEntry.uniqueId as the attachmentInfo/identifier and an attachmentInfo/accesstype of "urn:IHE:iti:2011:xdw:XDSregistered".

Table 5.4.3-8: taskData Element

taskData element	Standard	Data Type	Optionality			Description
			Create	View	Update	
taskDetails	OASIS_WS-HumanTask	`tTaskDetails`	R	R	R	See table 5.4.3-9
description	OASIS_WS-HumanTask	`xsd:string`	R	R	R	Textual description
input	OASIS_WS-HumanTask	`tMessagePartsData`	R	R	R	URI references to all input clinical documents.
output	OASIS_WS-HumanTask	`tMessagePartsData`	R	R	R	URI references to all output documents.
fault	OASIS_WS-HumanTask	`tFaultData`	O	R	O	Description of fault
comments	OASIS_WS-HumanTask	`xs:string`	O	R	O	Simple text comments about the task

- `<taskDetails>`

Table 5.4.3-9 taskDetails Element

taskDetails element	Standard	Data Type	Optionality			Description
			Create	View	Update	

taskDetails element	Standard	Data Type	Optionality			Description
			Create	View	Update	
id	OASIS_WS-HumanTask	xsd:anyURI	R	R	R	Internal ID for the task
taskTypes	OASIS_WS-HumanTask	Enumeration	R	R	R	
name	OASIS_WS-HumanTask	xsd:QName	R	R	R	The name of the task
status	OASIS_WS-HumanTask	ht:tStatus	R	R	R	Recommend limiting values to the statuses described above.
priority	OASIS_WS-HumanTask	htt:tPriority	O	R	O	
taskInitiator	OASIS_WS-HumanTask	tuser	O	O	O	
taskStakeholders	OASIS_WS-HumanTask	tOrganizationalEntity	O	O	O	
potentialOwners	OASIS_WS-HumanTask	tOrganizationalEntity	O	O	O	Owners in Human Task terminology are people/organizations/ etc. that perform the task.
businessAdministrators	OASIS_WS-HumanTask	tOrganizationalEntity	O	O	O	
actualOwner	OASIS_WS-HumanTask	tUser	R	R	R	The actual performer of the task.
notificationRecipients	OASIS_WS-HumanTask	tOrganizationalEntity	O	R	O	Notification Recipient may be used to contain information about persons to be notified. Use of

taskDetails element	Standard	Data Type	Optionality			Description
			Create	View	Update	
						this element does not imply that Human Task "notification" will be used. This element may be used to trigger notification mechanisms outside of XDW (e.g., IHE DSUB or NAV Profile). This combined use is not part of this profile specification
createdTime	OASIS_WS-HumanTask	xsd:dateTime	R	R	O	
createdBy	OASIS_WS-HumanTask	tUser	R	R	O	
lastModifiedTime	OASIS_WS-HumanTask	xsd:dateTime	R (Note 1)	R	R	
lastModifyBy	OASIS_WS-HumanTask	tUser	O	R	R	
activationTime	OASIS_WS-HumanTask	xsd:dateTime	O	R	O	
expirationTime	OASIS_WS-HumanTask	xsd:dateTime	O	R	O	
isSkipable	OASIS_WS-HumanTask	xsd:boolean	O	R	O	
hasPotentialOwners	OASIS_WS-HumanTask	xsd:boolean	O	O	O	
startedByTimeExists	OASIS_WS-HumanTask	xsd:boolean	X	X	X	

taskDetails element	Standard	Data Type	Optionality			Description
			Create	View	Update	
completedByTimeExists	OASIS_WS-HumanTask	xsd:boolean	X	X	X	
presentationName	OASIS_WS-HumanTask	tPresentationName	O	O	O	
presentationSubject	OASIS_WS-HumanTask	tPresentationSubject	O	O	O	
renderingMethodExists	OASIS_WS-HumanTask	xsd:boolean	R	R	R	Value shall be "false"
hasOutput	OASIS_WS-HumanTask	xsd:boolean	X	X	X	
hasFault	OASIS_WS-HumanTask	xsd:boolean	X	X	X	
hasAttachments	OASIS_WS-HumanTask	xsd:boolean	X	X	X	
hasComments	OASIS_WS-HumanTask	xsd:boolean	X	X	X	
escalated	OASIS_WS-HumanTask	xsd:boolean	O	R	O	
searchBy	OASIS_WS-HumanTask	xsd:string	X	X	X	
outcome	OASIS_WS-HumanTask	xsd:string	X	X	X	
parentTaskId	OASIS_WS-HumanTask	xsd:anyURI	X	X	X	XDW prohibits use of subTasks
hasSubTasks	OASIS_WS-HumanTask	xsd:boolean	X	X	X	XDW prohibits use of subTasks.

945 Note 1: lastModifiedTime shall be the same as createdTime

- `<taskEventHistory>`

Table 5.4.3-10: taskEventHistory Element

taskEventHistory element	Standard	Data Type	Optionality	Description
taskEvent	OASIS_WS-HumanTask	`taskEvent_type`	R	See table 5.4.3-11

- `<taskEvent>`

Table 5.4.3-11: taskEvent Element

taskEvent element	Standard	Data Type	Optionality	Description
id	OASIS_WS-HumanTask	`xs:integer`	R	
eventTime	OASIS_WS-HumanTask	`xs:dateTime`	R	
identifier	OASIS_WS-HumanTask	`xs:anyURI`	R	
principal	OASIS_WS-HumanTask	`xs:string`	O	
eventType	OASIS_WS-HumanTask	`ht:tTaskEventType`	R	The type of event that happens that solicits the modification of the status of the task (adding a new taskEvent). It should be valorized with one of these types: create, stop, suspend, resume, fail, complete. These types are defined in the HumanTask specification (C. WS-HumanTask Data Types Schema, <!-- Defines the human task event types -->).
startOwner	OASIS_WS-HumanTask	`xs:string`	O	
endOwner	OASIS_WS-HumanTask	`xs:string`	O	

taskEvent element	Standard	Data Type	Optionality	Description
status	OASIS_WS-HumanTask	ht:tStatus	R	
hasData	OASIS_WS-HumanTask	xs:Boolean	O	
eventData	OASIS_WS-HumanTask	xs:anyType	R2	This structure includes the data elements that were changed by this event.
faultName	OASIS_WS-HumanTask	xs:string	O	
	OASIS_WS-HumanTask			

5.4.4 Complete Example

In the example in figure 5.4.4-1 represents the xlm of the XDW Workflow Document for the use case described in ITI TF 1: 30.4.2.1. This example represents the complete Workflow Document at the end of the process (column C of figure 30.4.2.1.1-1).

In this case there are two tasks:

- the first task has been created in status “COMPLETED” and so it has only one taskEvent in the taskEventHistory;
- the second task ends the process in status “COMPLETED” and it has two taskEvent.

```
<?xml version="1.0" encoding="UTF-8"?>
<xdw:XDW.WorkflowDocument xmlns:hl7="urn:hl7-org:v3"
   xmlns:ws-ht="http://docs.oasis-open.org/ns/bpel4people/ws-humantask/types/200803"
   xmlns:xdw="urn:ihe:iti:2011:xdw"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:ihe:iti:2011:xdw file:XDW-2012-06-21.xsd">
   <xdw:id root="1.2.3.4.5"/>
   <xdw:effectiveTime value="20110401031520"/>
   <xdw:confidentialityCode code="1.24.3.3.3"/>
   <xdw:patient>
      <xdw:id root="1.3.6.1.4.1.21367.13.20.1000" extension="33333"  assigningAuthorityName="IHERED"/>
   </xdw:patient>
   <xdw:author>
      <xdw:assignedAuthor>
         <hl7:id root="1.2.3.4.5" extension="11111"/>
         <hl7:assignedPerson>
                  <hl7:name>
                           <hl7:family>Blum</ns1:family>
                           <hl7:prefix>Dr.</ns1:prefix>
                  </hl7:name>
         </hl7:assignedPerson>
      </xdw:assignedAuthor>
```



```
  </xdw:author>
  <xdw:workflowInstanceID>urn:oid:1.2.3.4</xdw:workflowInstanceID>
  <xdw:workflowDocumentSequenceNumber>3</xdw:workflowDocumentSequenceNumber>
  <xdw:workflowStatus>CLOSED</xdw:workflowStatus>
  <xdw:workflowStatusHistory>
    <xdw:documentEvent>
      <xdw:eventTime>2011-03-28T10:00:12.0Z</xdw:eventTime>
      <xdw:eventType>create</xdw:eventType>
      <xdw:taskEventIdentifier> urn:oid:1.2.3.4.5</xdw:taskEventIdentifier>
      <xdw:author>Mr. Rossi</xdw:author>
      <xdw:previousStatus></xdw:previousStatus>
      <xdw:actualStatus>OPEN</xdw:actualStatus>
    </xdw:documentEvent>
    <xdw:documentEvent>
      <xdw:eventTime>2011-04-01T03:15:20.0Z</xdw:eventTime>
      <xdw:eventType>complete</xdw:eventType>
      <xdw:taskEventIdentifier> urn:oid:1.2.3.4.7</xdw:taskEventIdentifier>
      <xdw:author>Dr. Brum</xdw:author>
      <xdw:previousStatus>OPEN</xdw:previousStatus>
      <xdw:actualStatus>CLOSED</xdw:actualStatus>
    </xdw:documentEvent>
  </xdw:workflowStatusHistory>
<xdw:workflowDefinitionReference>urn:oid:1.2.3.4.5.6.7.8.9</xdw:workflowDefinitionReference>
  <xdw:TaskList>
    <xdw:XDWTask>
      <xdw:taskData>
        <ws-ht:taskDetails>
          <ws-ht:id>1</ws-ht:id>
          <ws-ht:taskType>Requested</ws-ht:taskType>
          <ws-ht:name>ReferralRequested</ws-ht:name>
          <ws-ht:status>COMPLETED</ws-ht:status>
          <ws-ht:actualOwner>Mr. Rossi</ws-ht:actualOwner>
          <ws-ht:createdTime>2011-03-28T10:00:12.0Z</ws-ht:createdTime>
          <ws-ht:createdBy>Mr. Rossi</ws-ht:createdBy>
          <ws-ht:lastModifiedTime>2011-03-28T10:00:12.0Z</ws-ht:lastModifiedTime>
          <ws-ht:renderingMethodExists>false</ws-ht:renderingMethodExists>
        </ws-ht:taskDetails>
        <ws-ht:description>Request for a specialist visit</ws-ht:description>
        <ws-ht:input>
          <!-- one part element for each document in input -->
          <ws-ht:part name="ClinicalRepDoc1">        <!--ClinicalRepDoc1-->
            <ws-ht:reference uid="urn:oid:1.2.3.4.4.1"/>
          </ws-ht:part>
        </ws-ht:input>
        <ws-ht:output>
          <!-- one part element for each document in output -->
          <ws-ht:part name="eReferralDoc1">      <!--eReferralDoc1-->
            <ws-ht:reference uid="urn:oid:1.2.3.4.4.2"/>
          </ws-ht:part>
        </ws-ht:output>
      </xdw:taskData>
      <xdw:taskEventHistory>
        <xdw:taskEvent>
          <xdw:id>101</xdw:id>
          <xdw:eventTime>2011-03-28T10:00:12.0Z</xdw:eventTime>
          <xdw:identifier>urn:oid:1.2.3.4.5</xdw:identifier>
```

```
            <xdw:eventType>create</xdw:eventType>
            <xdw:status>COMPLETED</xdw:status>
          </xdw:taskEvent>
        </xdw:taskEventHistory>
      </xdw:XDWTask>
      <xdw:XDWTask>
        <xdw:taskData>
          <ws-ht:taskDetails>
            <ws-ht:id>2</ws-ht:id>
            <ws-ht:taskType>Referral Referred</ws-ht:taskType>
            <ws-ht:name>Referred</ws-ht:name>
            <ws-ht:status>COMPLETED</ws-ht:status>
            <ws-ht:actualOwner>Dr. Brum</ws-ht:actualOwner>
            <ws-ht:createdTime>2011-03-29T09:20:01.0Z</ws-ht:createdTime>
            <ws-ht:createdBy>Dr. Brum</ws-ht:createdBy>
            <ws-ht:lastModifiedTime>2011-04-01T03:15:20.0Z</ws-ht:lastModifiedTime>
            <ws-ht:renderingMethodExists>false</ws-ht:renderingMethodExists>
          </ws-ht:taskDetails>
          <ws-ht:description>Specialist visit</ws-ht:description>
          <ws-ht:input>
            <!-- one part element for each document in input -->
            <ws-ht:part name="eReferralDoc1">     <!--eReferralDoc1-->
              <ws-ht:reference uid="urn:oid:1.2.3.4.4.2"/>
            </ws-ht:part>
          </ws-ht:input>
          <ws-ht:output>
            <!-- one documentReference element for each document in input -->
            <ws-ht:part name="clinicalRepDoc2">     <!--ClinicalRepDoc2-->
              <ws-ht:reference uid="urn:oid:1.2.3.4.4.4"/>
            </ws-ht:part>
          </ws-ht:output>
        </xdw:taskData>
        <xdw:taskEventHistory>
          <xdw:taskEvent>
            <xdw:id>201</xdw:id>
            <xdw:eventTime>2011-03-29T09:20:01.0Z</xdw:eventTime>
            <xdw:identifier>urn:oid:1.2.3.4.6</xdw:identifier>
            <xdw:eventType>create</xdw:eventType>
            <xdw:status>IN_PROGRESS</xdw:status>
          </xdw:taskEvent>
          <xdw:taskEvent>
            <xdw:id>202</xdw:id>
            <xdw:eventTime>2011-04-01T03:15:20.0Z</xdw:eventTime>
            <xdw:identifier>urn:oid:1.2.3.4.7</xdw:identifier>
            <xdw:eventType>complete</xdw:eventType>
            <xdw:status>COMPLETED</xdw:status>
          </xdw:taskEvent>
        </xdw:taskEventHistory>
      </xdw:XDWTask>
    </xdw:TaskList>
</xdw:XDW.WorkflowDocument>
```

Figure 5.4.4-1: Sample XDW Workflow Document

5.4.5 Workflow Document Management

5.4.5.1 Workflow Document Lifecycle Management

The Cross-Enterprise Document Workflow profile takes advantage of the lifecycle management of XDS when used in an XDS environment.

A Workflow Document shall be created and submitted in a Folder. The initial document shall include at least one task on the TaskList, and have a workflowStatus of OPEN. The Workflow Document is updated when:

- The information about a task is modified. This may be due to a change in some other task related information like updating the output information.
- A new task is added to the `<TaskList>`.
- The workflow status is changed to CLOSED.

Each update shall be done using the XDS Document Replace when in an XDS environment. The series of steps to be taken is:

- Update the XDW document to reflect the desired changes. This is often replacement of the `<TaskData>`. It may also be a change by adding a new task to the `<TaskList>` or a new `<taskEvent>` to a Task.
- Use the XDS Replace operation to replace the old document with this modified document. By virtue of the standard XDS handling of folders, this replacement document will be in the same XDS Folder as the original Workflow Document.
- It is possible that a document replace will be rejected by the XDS Document Registry if another actor has also done a replace in the time since the Workflow Document instance was obtained. In this case (attempting to replace a document already replaced), the XDW Document Creator or Updater shall obtain the most recent version of the Workflow Document which was updated by another XDW Actor, consider the evolution of the workflow, and performed a new content update. This kind of race condition should be very rare because updating is much faster than the rate at which people perform tasks. If certain workflows definitions require reducing the likelihood of such race conditions, one should consider placing in the Workflow Description one or more tasks "In Progress" and requiring that other Actor wait while such tasks are in-progress.

When using XDR or XDM, the receiving actor shall perform an equivalent local update process.

When an XDW Actor decides that a workflow status code shall be placed in a CLOSED status, a final update to set the workflow status code to CLOSED shall be performed. The specific rules for determining when and which XDW Actors are allowed or should set the workflow status code to CLOSED are not specified by the XDW Profile. They may be determined within the

Workflow Definition. XDW Content Consumer and Content Updater Actors shall support the means to query for Workflow Documents that are in a workflow status OPEN.

This profile does not further constrain the rules for document lifecycle management, but a specific Workflow Definition may add requirements requiring that certain kinds of tasks be created initially, restricting the kinds of tasks that can be added, and requiring that updates be performed for specific task changes.

5.4.5.2 Associations Types and Folder

A clinical document can be referenced by many Workflow Documents in different steps and for different reasons. When the content of a Workflow Document is known, the related clinical documents are always reached through the references (XDSDocumentEntry.uniqueId and homeCommunityId) tracked inside the different task in the “input" and “output” elements.

The use of the Folder to group all versions of a Workflow Document is necessary to have a fixed id to identify the whole workflow. Since the Workflow Document will be replaced many times (it is replaced at each step), its uid/id is not useful for maintaining a fixed reference. The document uniqueId of each of the successive XDW documents can be used to identify a particular state of the workflow. Placing all versions of the same Workflow Document into the same folder allows the use of the FolderId as the fixed link to the workflow as a whole.

A Workflow Document shall be contained in only one Workflow Content Folder. Within a folder there is only one Workflow Document with Approved status, along with all the previous versions of the same Workflow Document with status Deprecated.

If a workflow generates another workflow there shall be two different folders, one for each workflow. The relationship between the different workflows is always guaranteed to be inside the Workflow Documents by referencing the workflow as output of the task of the parent Workflow Document and as the input of the first task in the child Workflow Document.

5.4.5.3 Create workflow

When the first step of a new workflow is completed, the XDW Content Creator shall:

- create the first version of the Workflow Document.

Then the XDW Content Creator Actor shall, using ITI-41 Provide and Register Document Set-b (in the case of XDS):

- create a new Workflow Context Folder
- submit the Workflow Document to the XDS Document Repository, in the new Workflow Context Folder created, using ITI-41 Provide and Register Document Set-b.

5.4.5.4 Update Workflow Document

For each subsequent step in the Workflow an XDW Content Updater shall:

- obtain the most recent version of the Workflow Document. This is the only version approved. For example, it may be obtained using a grouped XDS Document Consumer querying the approved workflow document in the Workflow Context Folder;
- update the content in the Workflow Document (by adding a new task or updating an existing task with a new `<taskEvent>`);
- re-register (update) the Workflow Document by performing a document replace (e.g., in a XDS environment using a grouped XDS Document Source).

This new version is automatically added to the correct Workflow Context Folder by normal XDS rules for document replacement in the context of a folder.

In a Document Sharing infrastructure (e.g., an XDS environment) two different Content Updaters could be in the situation of race condition when both update, in the same time, the same Workflow Document.

In this case two actors (Content Updater A and Content Updater B) retrieve the same Workflow Document (e.g., Workflow Document with document uniqueId 1) and change it.

Content Updater A deprecates the previous version and publishes a new version updated with a new document uniqueId (e.g., document uniqueId 2).

When Updater Creator B tries to replace the same Workflow Document (document uniqueId 1) with his updated version this transaction generates an error because the document uniqueID 1 is deprecated and replaced with document uniqueId2.

Content Updater B shall retrieve the current version of the Workflow Document (document uniqueId 2) and update it with a new version of the document with document uniqueId 3.

5.4.5.5 Association of a clinical document to a task and <taskEvent>

Any clinical documents included as input or output documents within the taskData element that are registered in an XDS Registry shall be referenced using uniqueId and homeCommunityId of the Clinical Document referenced..

5.4.5.6 Get the Workflow Document and a clinical document associated to the workflow

The most recent version of the Workflow Document may be retrieved at any point during the workflow.

The version of the Workflow Document with an approved status contains the most current information on the workflow and its tasks. So an XDW Content Consumer needs to analyze only the approved version to get all current information.

Any Workflow Document contains details of each task that has been performed. A task or `<taskEvent>` includes the references (XDSDocumentEntry.uniqueId and homeCommunityId) to zero or more input and/or output clinical documents. These documents may be obtained by

means of XDS, or should be included along with the Workflow Document if XDR or XDM is used.

5.4.5.7 Use of the eventCodeList to manage the status of a Workflow Document

An overall workflow status is required to be set by each author of a new step. This value is either OPEN or CLOSED.

This workflow status is required to be present in every workflow step, and shall take either the value OPEN or CLOSED.

By setting this workflow status to OPEN, a step author indicates that, for the workflow definition and the step author further steps are expected to be performed.

By setting this workflow status to CLOSED, a step author indicates that, for the workflow definition and the step author no further steps are expected to be performed.

This workflow status shall be present for all XDW documents in its eventCodeList metadata.

This use of workflow status enables the use of query to locate OPEN or CLOSED workflows with certain other properties.

The EventCodeList contains the workflow status with two possible code values: either OPEN or CLOSED.

5.4.5.8 Parameters for Required Queries

The section below documents some examples of the possible queries in an XDS environment (defined in the Query Stored Registry transaction [ITI-18]) to obtain the different documents related to the workflow from some parameters available:

- Find all open Workflow Documents for a patient

  An XDS Registry Stored Query “FindDocuments” maybe used with patientId, XDW document formatCode and eventCodeList with the value “OPEN” for the Workflow Document.

- Find all particular type of open Workflow Documents for a patient

  An XDS Registry Stored Query “FindDocuments” maybe used with patientId, XDW document formatCode, eventCodeList with the value “OPEN” for the Workflow Document and a specific XDW document typeCode.

- Get one or more documents referenced in a Workflow Document

  An XDS Registry Stored Query “FindDocuments” which retrieves the Workflow Document (like in the first example) and an XDS Registry Stored Query “GetDocuments” with document uniqueId and homeCommunityId to retrieve one or more documents referenced inside the Workflow Document.

- Get the Workflow Folder in which a Workflow Document is contained

An XDS Registry Stored Query "GetFoldersForDocument" maybe used with document uniqueId and homeCommunityId.

- Find the Workflow Folder for a patient

  An XDS Registry Stored Query "FindFolders" maybe used with patientId and an XDW Folder codeList.

- Find an open Workflow Document in a Workflow Folder

  An XDS Registry Stored Query "GetFolderAndContents" maybe used with Folder uniqueId, XDW document formatCode and eventCodeList with the value "OPEN" for the Workflow Document.

5.4.6 XDS Metadata

5.4.6.1 Document Metadata

The following metadata elements shall be used to describe the Workflow Document in an XDS Affinity Domain. The XDW profile does not introduce new metadata and all the metadata elements used are the common XDS document metadata specified in ITI TF-3: 4.2.3.2.

Table 5.4.6.1-1: Document Metadata Attribute Definition

XDSDocumentEntry Attribute	Definition
author	Represents the humans and/or machines that authored the document. In the Workflow Document the Author is the human and/or machine which most recently updated the Workflow Document. This means that when a Workflow Document is updated by a different person or machine, the author changes.
authorInstitution (sub-attribute of author)	No special requirements for Workflow Document
authorPerson (sub-attribute of author)	No special requirements for Workflow Document
authorRole (sub-attribute of author)	No special requirements for Workflow Document
authorSpecialty (sub-attribute of author)	No special requirements for Workflow Document
availabilityStatus	No special requirements for Workflow Document
classCode	*Comment: This code will be requested from LOINC and will be inserted when provided*
classCode DisplayName	*The classCodeDisplayName will be provided by LOINC to identify an XDW Workflow Document.*
comments	No special requirements for Workflow Document
confidentialityCode	No special requirements for Workflow Document
creationTime	No special requirements for Workflow Document
entryUUID	No special requirements for Workflow Document
eventCodeList	For a Workflow Document, one code of this list shall be used to define the overall status of the workflow. This code shall have one of the following two values:

XDSDocumentEntry Attribute	Definition
	• code value = urn:ihe:iti:xdw:2011:eventCode:open, scheme = 1.3.6.1.4.1.19376.1.2.3 • code value = urn:ihe:iti:xdw:2011:eventCode:closed, scheme = 1.3.6.1.4.1.19376.1.2.3 (See ITI-TF-3 5.4.5.7.)
eventCodeListDisplay Name	This code should have one of the following two values: • Open Workflow • Closed Workflow
formatCode	urn:ihe:iti:xdw:2011:workflowDoc codesystem: 1.3.6.1.4.1.19376.1.2.3
hash	No special requirements for Workflow Document
healthcareFacility TypeCode	No special requirements for Workflow Document
healthcareFacility TypeCodeDisplay Name	No special requirements for Workflow Document
homeCommunityId	No special requirements for Workflow Document
languageCode	No special requirements for Workflow Document
legalAuthenticator	No special requirements for Workflow Document
mimeType	No special requirements for Workflow Document
patientId	No special requirements for Workflow Document
practiceSettingCode	No special requirements for Workflow Document
practiceSettingCode DisplayName	No special requirements for Workflow Document
repositoryUniqueId	No special requirements for Workflow Document
serviceStartTime	Shall be the starting time the service being documented took place For the Workflow Document the serviceStartTime is the time at which work began on the earliest task for this workflow. If present, shall have a single value.
serviceStopTime	No special requirements for Workflow Document
size	No special requirements for Workflow Document
sourcePatientId	No special requirements for Workflow Document
sourcePatientInfo	No special requirements for Workflow Document
title	No special requirements for Workflow Document
typeCode	Shall be assigned codes from the value set specifying the precise kind of document (e.g., Pulmonary History and Physical, Discharge Summary, Ultrasound Report). The different IHE domains and/or XDS Affinity Domains are expected to define values for the different kinds of workflow.

XDSDocumentEntry Attribute	Definition
typeCodeDisplay Name	No special requirements for Workflow Document
uniqueId	No special requirements for Workflow Document
URI	No special requirements for Workflow Document

5.4.6.2 XDS SubmissionSet Metadata

No additional constraints. See ITI TF-3: 4.2.3.3.

5.4.6.3 XDS Folder Metadata

The metadata elements shall be used to describe the XDS Folder that contains all versions of the Workflow Document (and/or possibly clinical documents related to it). The XDW profile does not introduce new metadata and all the metadata elements used are the common XDS folder metadata specified in ITI TF-3: 4.2.3.4. At the level of the XDS Affinity Domain is necessary to define some specific codes to identify the type of the Folder used.

Table 5.4.6.3-1: Folder Metadata Attribute Definition

XDSFolder Attribute	Definition
availabilityStatus	No special requirements for Workflow Document
codeList	Contains the set of codes specifying the type of clinical activity that resulted in placing XDS Document in this XDSFolder. This profile specifies one coded value for use in this field. Other profiles that specify kinds of workflows will define additional codes that can be used in this codelist. (urn:ihe:iti:xdw:2011:unspecworkflow, “IHE”, “Unspecified Workflow”)
codeListDisplayName	The value shall be “XDW Workflow Context Folder”.
comments	No special requirements for Workflow Document
entryUUID	No special requirements for Workflow Document
homeCommunityId	No special requirements for Workflow Document
lastUpdateTime	No special requirements for Workflow Document
patientId	No special requirements for Workflow Document
title	No special requirements for Workflow Document
uniqueId	Contains the unique identifier for the overall workflow instance. This identifier is created by the creator of the first step and remains fixed, even as new workflow steps are been added. This value shall match the workflow ID in the Workflow Documents.

5.4.6.3.1 XDSDocumentEntry.formatCode

The XDSDocumentEntry.formatCode shall be **urn:ihe:iti:xdw:2011:workflowDoc**. The formatCode codeSystem shall be 1.3.6.1.4.1.19376.1.2.3.

Volume 2x

Add Appendix X to volume 2x

Appendix X Basic Unstructured Workflow Definition Example

This is a Workflow Definition example that is intended to be used in conjunction with XDW Profile.

X.1 Workflow definition identifier

Basic Unstructured Workflow is a very simple workflow in which a group of physicians/organizations act on the same patient in an a priori unpredictable way.

This workflow is performed to allow the continuity of care for a patient in a generic and flexible way.

We expect actual deployment to modify this example when developing basic workflows. It has two simple types of Tasks: the first one is useful for recording and sharing single user actions ("*I did this task"*) and the second one used to request that a task be performed by another organization and reporting its completion (*"please do this, I did it"*).

Any specific workflow can include any combination of these two types of tasks. This example shows no dependencies among the tasks that are explicitly managed.

The catalog of task that maybe used in this workflow definition is not specified by this profile, and remains to be agreed in the affinity domain where the workflow is been deployed. This definition will result in a list of names and codes for any potential task.

X.2 Workflow definition identifier

The workflow definition identifier shall be inserted into the DefinitionReference element of the Workflow Document.

Workflow Definition name	DefinitionReference
Basic Unstructured Workflow	Urn: OID: reference to Workflow definition Document

X.3 Workflow opening and closing

The workflow should be opened by a physician or an Organization that participate to the workflow (e.g., continuity of care process). Any participant may choose to close the workflow.

X.4 Tasks descriptions

X.4.1 Task type “born completed”

Tasks of the type “born completed” are used by any workflow participant when the workflow is open or at any later point in time. This type is used for a workflow in which a participant want to share some actions perform in his enterprise on the patient. Typical examples can be a visit, or an emergency admission, or a patient self-monitoring event.

Task attributes	Rules for the task “born completed”
Task dependencies	none
States allowed	COMPLETED
States transactions	None
input	Zero or more clinical document of unconstrain types
output	Zero or more clinical document of unconstrain types
owner	every physician/organization
owner changes	No
<taskEvent>	Only one

X.4.2 Task type “two states task”

This type of task is used by any workflow participant when the workflow is open or at any later point in time. This task type is used for a workflow in which a participant wants to share some actions performed in his enterprise on the patient. Typical examples can be a visit, or an emergency admission, or a patient self-monitoring event.

Task attributes	Rules for the task “two states task”
Task dependencies	None
States allowed	CREATED COMPLETED
States transactions	When a workflow participant request that this task be performed by another workflow participant he places the task in a Workflow Document with CREATED status (no owner). When the requested task is performed by a participant the task status shall be COMPLETED.
input	Zero or more clinical document of unconstrain types (e.g., eReferral Document, ePrescription)
output	Zero or more clinical document of unconstrain types (e.g., reports, radiological images, advice documents, dispensation documents)
owner	any physician/organization that process this task in CREATED state
changes of task owner	Allowed
<taskEvent>	At least two