Integrating the Healthcare Enterprise

IHE Pharmacy
Technical Framework Supplement

Community Medication Prescription and Dispense
(CMPD)

Trial Implementation

Date: September 29, 2014

Author: IHE Pharmacy Technical Committee

Email: pharmacy@ihe.net

Please verify you have the most recent version of this document. See here for Trial Implementation and Final Text versions and here for Public Comment versions.

Foreword

This is a supplement to the forthcoming IHE Pharmacy Technical Framework. 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 29, 2014 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 forthcoming Pharmacy Technical Framework. Comments are invited and may be submitted at http://www.ihe.net/Pharmacy_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 Pharmacy domain can be found at: http://ihe.net/IHE_Domains.

Information about the organization of IHE Technical Frameworks and Supplements and the process used to create them can be found at: http://ihe.net/IHE_Process and http://ihe.net/Profiles.

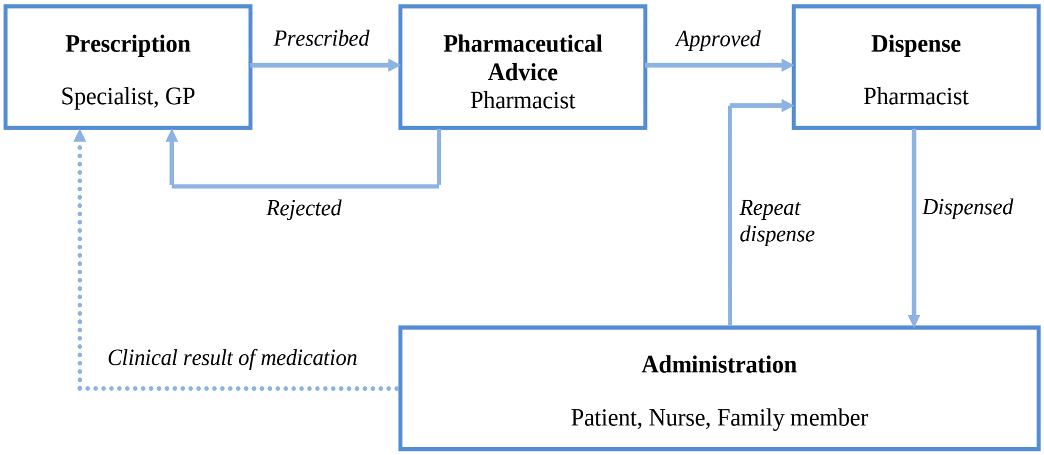
The current version of the IHE Pharmacy Technical Framework can be found at: http://ihe.net/Technical_Frameworks.

CONTENTS

Introduction

The Community Medication Prescription and Dispense Integration Profile (CMPD) describes the process of prescription, validation and dispense of medication in the community domain. This document is a detailed description of the generic implementation structure defined in the Common Parts document[1].

In general, the medication business process consists of four distinct processes, which have to be connected through interactions that transfer information and/or guide the workflow. The following figure shows this flow:



In the Community Pharmacy domain, the process of "administration of medication" can usually not be governed by IT based systems so just the processes "Prescription", "Pharmaceutical Advice" and "Dispense" are covered by the Community Pharmacy Prescription and Dispense Profile only.

The CMPD Profile is intended to be used in the context of the Pharmacy Content Profiles[2]:

- Pharmacy Prescription Supplement (PRE)
- Pharmacy Pharmaceutical Advice Supplement (PADV)

[1] This document is part of the IHE Pharmacy domain and can be obtained from the IHE web site.

[2] These supplements are part of the IHE Pharmacy domain and can be obtained from the IHE web site.

- Pharmacy Dispense Supplement (DIS)
- Pharmacy Medication List (PML)

These Content Profiles are based on the Patient Care Coordination (PCC) Technical Framework and define the semantic of the payload transported by the CMPD Profile.

This supplement also references other documents[3]. The reader should have already read and understood these documents:

1. PHARM Common parts document
2. IT Infrastructure Technical Framework Volume 1
3. IT Infrastructure Technical Framework Volume 2
4. IT Infrastructure Technical Framework Volume 3
5. HL7 and other standards documents referenced in this document

Open Issues and Questions

- The profile does not yet include the process step of getting the “current medication”, which is needed for checking interactions (ICAs) to the prescribed item.
- Grouping of XDW with the Community Pharmacy Manager: What, if the client-side actors (Prescription Placer, Pharmaceutical Adviser, Medication Dispenser) are not allowed to manage the workflow and this should be done by the CPM.

Closed Issues

- Question: Should be medication processes message or document-based? Decision at F2F meeting in Bordeaux (15./16.04.2010): Community domain is document based with XDS as persistence layer, Hospital domain will be message-based.
- Clarification to whitepaper: In community domain, the term “repository” in the whitepaper is intended to be interpreted as a technical system for persisting documents implementing XDS transactions as interface. XDS registry/repository systems as well as database or other persisting systems are likely to be used for this purpose.
- Changes to whitepaper:
  - “Consumer” actors will be removed, because they are just relaying transactions (don’t implement any own transactions). Sequence diagrams have been adapted.

[3] The first four documents can be located on the IHE Website at http://ihe.net/Technical_Frameworks/. The remaining documents can be obtained from their respective publishers.

- The transient aspects of “Ordering” are excluded in the profile. This should be generally discussed together with ITI in conjunction with all other “Ordering/Workflow” topics (e.g., Lab, Referral, etc.). (see CP-PHARM-018_v5)

Volume 1 – Integration Profiles

1.n Copyright Permission

Add the following to Sections 1.n:

Health Level Seven, Inc., has granted permission to the IHE to reproduce tables from the HL7 standard. The HL7 tables in this document are copyrighted by Health Level Seven, Inc. All rights reserved. Material drawn from these documents is credited where used.

2.1 Dependencies among Integration Profiles

Add the following to Table 2-1

Community Pharmacy Prescription and Dispense CMPD	XDS	CMPD Actors are based on XDS Document Source, Document Consumer, Registry and Repository actors and use XDS transactions.	Required to manage query, submission and retrieve of documents.
Community Pharmacy Prescription and Dispense CMPD	On-Demand Documents	CMPD Community Pharmacy Manager acts as an On-Demand Document Source Actor	Required to manage request of the Medication List
Community Pharmacy Prescription and Dispense CMPD	ATNA	Each CMPD Actor shall be grouped with Secure Node or Secure Application Actor	Required due to XDS grouping.
Community Pharmacy Prescription and Dispense CMPD	CT	Each CMPD Actor shall be grouped with the Time Client Actor	Required due to ATNA grouping.
Community Pharmacy Prescription and Dispense CMPD	XDW	Some CMPD actors can be optionally grouped with XDW Workflow Management	Optional due to XDW grouping

Add the following section to Section 2.2

2.2.4 Community Medication Prescription and Dispense Integration Profile

The Community Medication Prescription and Dispense Integration Profile (CMPD) describes the process of prescription, validation and dispense of medication in the community domain.

The CMPD Profile is intended to be used in the context of the Pharmacy Content Profiles[4]:

- Pharmacy Prescription Supplement (PRE)

[4] These supplements are part of the IHE Pharmacy domain and can be obtained from the IHE web site.

- Pharmacy Pharmaceutical Advice Supplement (PADV)
- Pharmacy Dispense Supplement (DIS)
- Pharmacy Medication List (PML)

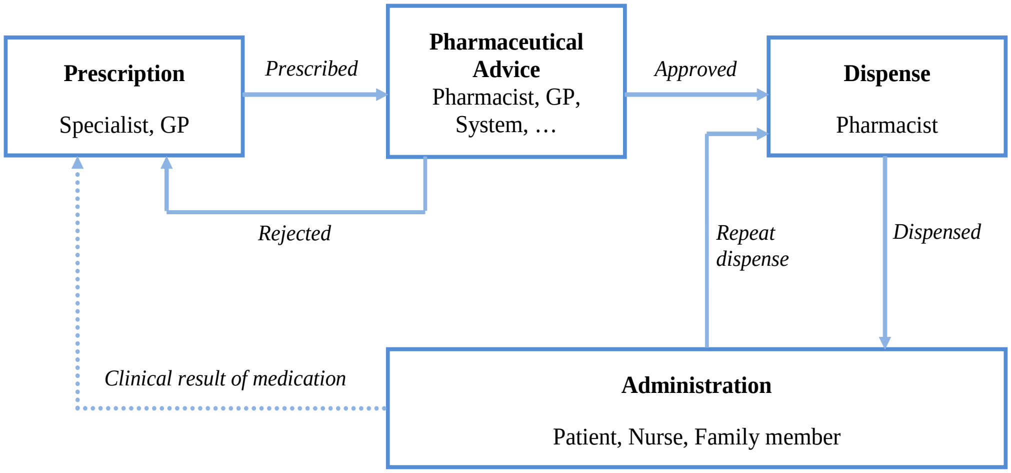
These Content Profiles are based on the Patient Care Coordination (PCC) Technical Framework and define the semantic of the payload transported by the CMPD Profile.

Add Section 4

4 Community Medication Prescription and Dispense Integration Profile

The Community Medication Prescription and Dispense Integration Profile (CMPD) describes the
210 process of prescription, validation and dispense of medication in the community domain.

In general, the medication business process consists of four distinct processes, which have to be connected through interactions that transfer information and/or guide the workflow. The following figure shows this flow:



215 **Figure 4-1: Medication Prescription and Dispense Process**

In the Community Pharmacy domain, the process of “administration of medication” can usually not be governed by IT based systems so just the processes “Prescription”, “Pharmaceutical Advice” and “Dispense” are covered by the Community Pharmacy Prescription and Dispense Profile only.

220 The CMPD Profile is intended to be used in the context of the Pharmacy Content Profiles[5]:

- Pharmacy Prescription Supplement (PRE)
- Pharmacy Pharmaceutical Advice Supplement (PADV)
- Pharmacy Dispense Supplement (DIS)

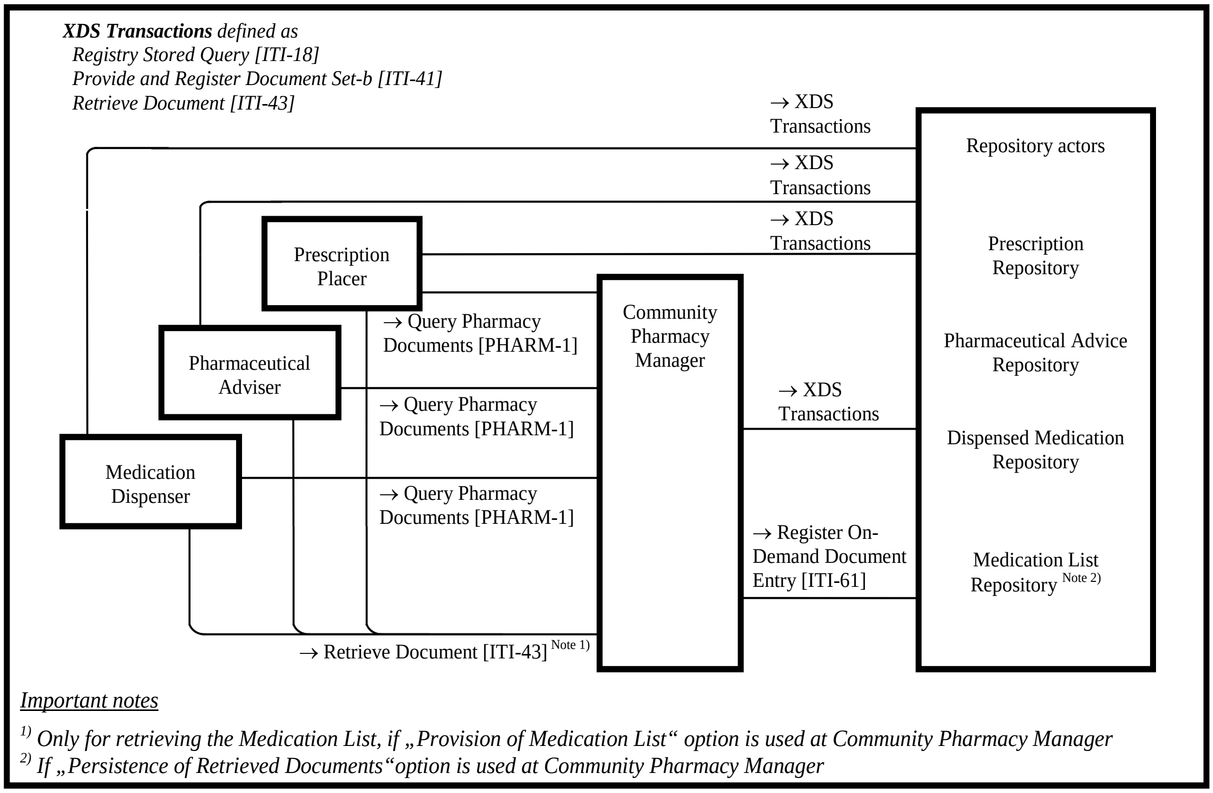
[5] These supplements are part of the IHE Pharmacy domain and can be obtained from the IHE web site.

- Pharmacy Medication List (PML)

These Content Profiles are based on the Patient Care Coordination (PCC) Technical Framework and define the semantic of the payload transported by the CMPD Profile.

4.1 Actors/ Transactions

Figure 4.1-1 shows the actors directly involved in the Community Medication Prescription and Dispense Integration Profile and the relevant transactions between them. Other actors that may be indirectly involved due to their participation in the XDS integration profiles, etc., are not necessarily shown.


XDS Transactions defined as
Registry Stored Query [ITI-18]
Provide and Register Document Set-b [ITI-41]
Retrieve Document [ITI-43]
→ XDS Transactions
→ XDS Transactions
→ XDS Transactions
Repository actors
Prescription Repository
Pharmaceutical Advice Repository
Dispensed Medication Repository
Medication List Repository Note 2)
Prescription Placer
Pharmaceutical Adviser
Medication Dispenser
Community Pharmacy Manager
→ Query Pharmacy Documents [PHARM-1]
→ Query Pharmacy Documents [PHARM-1]
→ Query Pharmacy Documents [PHARM-1]
→ XDS Transactions
→ Register On-Demand Document Entry [ITI-61]
→ Retrieve Document [ITI-43] Note 1)

Important notes

[1] Only for retrieving the Medication List, if „Provision of Medication List“ option is used at Community Pharmacy Manager

[2] If „Persistence of Retrieved Documents“option is used at Community Pharmacy Manager

Figure 4.1-1: Community Medication Prescription and Dispense Actor Diagram

Important Note:

The Community Pharmacy Manager Actor (CPM) is currently restricted to perform data “filtering" by the PHARM-1 “Query Pharmacy Documents” transaction as well as data “relaying” of the ITI-43 “Retrieve Document Set” transaction. All other XDS transactions are performed directly between the client actors (Prescription Placer, Pharmaceutical Adviser, Medication Dispenser) and the Registry/Repository actors (Prescription, Pharmaceutical Advice, Dispense repositories).

The “relaying” of transactions for client actors is not shown in this Actor diagram for readability. Please see chapter “CMPD Implementation scenarios” for details to the usage of the “relaying” functionality of the CPM in case e.g., of a multi-domain implementation scenario.

Table 4.1-1 lists the transactions for each actor directly involved in the Community Medication Prescription and Dispense Profile. In order to claim support of this Integration Profile, an implementation must perform the required transactions (labeled “R”). Transactions labeled “O” 240 are optional. A complete list of options defined by this Integration Profile and that implementations may choose to support is listed in Volume 1, Section 4.2.

Table 4.1-1: Community Medication Prescription and Dispense Integration Profile - Actors and Transactions

Actors	Transactions	Optionality	Section in Vol. 2
Community Pharmacy Manager	Registry Stored Query (ITI-18) (acting as a Document Consumer)	R	ITI-TF-2a:3.18
	Retrieve Document Set (ITI-43) (acting as a Document Consumer)	R	ITI-TF-2b:3.43
	Register On-Demand Document Entry (ITI-61) (acting as an On-Demand Document Source)	O[6]	ITI-TF-2b:3.61
	Query Pharmacy Documents (PHARM-1)	R	PHARM-TF-2:3.1
Prescription Placer	Registry Stored Query (ITI-18)	O[7]	ITI-TF-2a:3.18
	Provide and Register Document Set-b (ITI-41)	R	ITI-TF-2b:3.41
	Retrieve Document Set (ITI-43)	R	ITI-TF-2b:3.43

[6] R if „Provision of Medication List“ Option is used

[7] R if „Workflow Management“ Option is used

Actors	Transactions	Optionality	Section in Vol. 2
	Query Pharmacy Documents (PHARM-1)	R[8]	PHARM-TF-2:3.1
Pharmaceutical Adviser	Registry Stored Query (ITI-18)	O[9]	ITI-TF-2a:3.18
	Provide and Register Document Set-b (ITI-41)	R	ITI-TF-2b:3.41
	Retrieve Document Set (ITI-43)	R	ITI-TF-2b:3.43
	Query Pharmacy Documents (PHARM-1)	R[10]	PHARM-TF-2:3.1
Medication Dispenser	Registry Stored Query (ITI-18)	O[11]	ITI-TF-2a:3.18
	Provide and Register Document Set-b (ITI-41)	R	ITI-TF-2b:3.41
	Retrieve Document Set (ITI-43)	R	ITI-TF-2b:3.43
	Query Pharmacy Documents (PHARM-1)	R[12]	PHARM-TF-2:3.1
Repository actors Prescription Pharmaceutical Advice Dispense Medication List[13]	Registry Stored Query (ITI-18)	R	ITI-TF-2a:3.18
	Provide and Register Document Set-b (ITI-41)	R	ITI-TF-2b:3.41
	Retrieve Document Set (ITI-43)	R	ITI-TF-2b:3.43

4.1.1 Actors

4.1.1.1 Community Pharmacy Manager

The main role of this actor consists in providing the business logic for status management and other purposes. It may also serve as provider of the business logic for creating the Medication List if "Provision of Medication List" Option is supported. As a second role it acts as a "relaying role" where certain standard XDS communication is routed through for providing the possibility of applying project-specific business logic on it.

It provides special query-transactions which consuming actors (Prescription Placer, Pharmaceutical Adviser or Medication Dispenser) use for reducing the amount of data flowing to them. They return just "relevant" information for specific purposes (e.g., returning just all

[8] O if „Workflow Management“ Option is used

[9] R if „Workflow Management“ Option is used

[10] O if „Workflow Management“ Option is used

[11] R if „Workflow Management“ Option is used

[12] O if „Workflow Management“ Option is used

[13] If „Persistence of Retrieved Documents“ Option is used at Community Pharmacy Manager

"active" prescriptions ready for being validated or dispensed together with all related documents).

Furthermore it may provide special query-transactions which consuming actors (Prescription Placer, Pharmaceutical Adviser or Medication Dispenser) use to request a Medication List to a patient. Fulfilling the request the actor gathers and assembles Prescription- and Dispense items to a Medication List document according to the "Pharmacy Medication List" (PML) Profile. Subsequently this resulting document is returned to the requesting actor. For this functionality this actor acts as an ITI On-Demand Document Source Actor as described in the "On-Demand documents" supplement.

This actor is usually a system actor without human participation.

4.1.1.2 Prescription Placer

The main role of this actor consists in placing the prescription (initial or modified in case of a substitution of invalidation, for example). It sends the cancelation of the prescription or its discontinuation, as well. In order to fulfill this task, the Prescription Placer retrieves the current treatment of the patient and medication already dispensed recently.

4.1.1.3 Pharmaceutical Adviser

This actor is responsible for the validation of prescriptions from a pharmacist's perspective. Therefore, it receives the initial prescription, validates it and sends it back (accepted, cancelled, modified, substitution of pharmaceutical product); therefore it provides the pharmaceutical advice. To perform this task it checks the current treatment.

Pharmaceutical Advisers (e.g., automated ICA check modules) may also provide "draft" advices which don't affect the status of a prescription but serve as a foundation for the advice performed by another Pharmaceutical Adviser.

4.1.1.4 Medication Dispenser

This actor is responsible for the process of dispensing medication to the patient, fulfilling the prescription. Therefore it produces the information on the medication dispensed to the patient. In order to achieve this, it receives prescriptions already validated. It also confirms drug availability for administration and it receives the administration plan and administration reports. This actor may be implemented as the point of sale software of a community pharmacy or the hospital pharmacy module of a hospital information system. The human actor behind this system actor is usually a pharmacist or a pharmacist assistant.

4.1.1.5 Repository actors

Formally the Community Pharmacy process defines different "repositories" for Prescriptions, Pharmaceutical Advices and Dispenses, but they shall be seen as abstract repository-roles for persisting the appropriate document types the documents, not as XDS repositories defined in the "Cross Document Sharing" (XDS) Integration Profile of the ITI Technical Framework.

This profile rather makes use of the XDS Profile for defining abstract XDS registry and repository actors for modeling the abstract repository-roles for real implementations.

Description of the abstract repository-roles:

- Prescription Repository
  - This repository contains the medication prescribed to the patient from the Prescription Placer and may receive updates to the current treatment (cancelations, changes, etc.). It also provides information about the current prescribed medication to other actors such as the Community Pharmacy Manager.
- Pharmaceutical Advice Repository
  - This repository contains the pharmaceutical advice issued by the Pharmaceutical Adviser (typically a pharmacist). It provides this information to other actors such as the Community Pharmacy Manager.
- Dispensed Medication Repository
- This repository contains the medication actually dispensed to the patient; this information is received from the Medication Dispenser. The Dispensed Medication Repository provides the medication record of the patient to other actors such as the Community Pharmacy Manager.

Conforming to the ITI XDS Technical Framework, registry actors are used for storing metadata of the submitted documents, the repository actors store the actual documents.

Implementation scenarios in real-world projects will most likely differ from the topology of having exactly three repositories. They may vary from single XDS affinity domain scenarios with just one registry/repository system for storing all document-types to most complex scenarios including many different XDS affinity domains for covering the organizational and strategic need of separation of the participating parties (e.g., Prescribers and Pharmacists).

All mechanisms defined in the XDS Integration Profile for accessing XDS Registry/Repository systems apply and may be used for communicating, e.g., “Cross Community Access” (XCA).

4.1.2 Transactions

4.1.2.1 Query Pharmacy Documents

This transaction defines how a querying actor has to query the Community Pharmacy Manager for prescriptions (PRE) and their related documents. Related documents are Pharmaceutical Advice (PADV) and Dispense (DIS) documents.

Querying actors may be:

- Prescription Placer
- Pharmaceutical Adviser

- Medication Dispenser

This transaction provides a set of specialized queries:

(1) Specialized queries allow the finding of prescriptions and their related documents for specific purposes (e.g., for validation).

These are:

- **FindPrescriptions**
    - Find prescriptions and their related documents
- **FindDispenses**
    - Find dispense documents and their related documents
- **FindPrescriptionsForValidation**
    - Find prescriptions and their related documents containing Prescription Items ready to be validated
- **FindPrescriptionsForDispense**
    - Find prescriptions and their related documents containing Prescription Items ready to be dispensed

Both specialized queries can be parameterized to …

1. … either check the status of a given prescription (e.g., if the patient shows the printed prescription to the operator and the prescription ID can be read off it).
   In this case the ID of the given prescription is set in the query parameters - if the prescription is in the requested status (e.g., “ready for dispense”) it shows up in the query result (together with its related documents); otherwise the query result is empty which indicates that the given prescription is not in the requested status.
2. … or to search for prescriptions which are in a specific status (e.g., if the patient has no printed prescription and the implementation allows searching for prescriptions).
   In this case the query returns all prescriptions which are in the requested status (e.g., “ready for dispense”). The operator can choose and pick the right one.

(2) A query for requesting the Medication List, if “Provision of Medication List” Option is supported.

This is:

- **FindMedicationList**
    - Find the Medication List to a patient.

4.1.2.2 Registry Stored Query

This transaction is used by a Prescription Placer, Pharmaceutical Adviser or Medication Dispenser Actor to a registry actor (Prescription/Pharmaceutical Advice/Dispensed medication registry) in order to query for Prescription, Pharmaceutical Advice or Dispense documents based on the querying actor’s query parameters.

See the XDS Integration Profile of the ITI Technical Framework for a detailed description of this transaction (ITI-TF2a:3.18).

4.1.2.3 Provide and Register Document Set-b

This transaction is sent by a Prescription Placer, Pharmaceutical Adviser or Medication Dispenser Actor to a repository actor (Prescription/Pharmaceutical Advice/Dispensed Medication Repository) in order to submitting one or more Prescription, Pharmaceutical Advice or Dispense documents. See the XDS Integration Profile of the ITI Technical Framework for a detailed description of this transaction (ITI-TF2b:3.41).

4.1.2.4 Retrieve Document Set

This transaction is sent by a Prescription Placer, Pharmaceutical Adviser or Medication Dispenser Actor to a repository actor (Prescription/Pharmaceutical Advice/Dispensed Medication Repository) or the Community Pharmacy Manager Actor in order to retrieve one or more Prescription, Pharmaceutical Advice or Dispense documents.

See the XDS Integration Profile of the ITI Technical Framework for a detailed description of this transaction (ITI-TF2b:3.43).

4.2 CMPD Integration Profile Options

Options that may be selected for this Integration Profile are listed in the Table 4.2-1 along with the Actors to which they apply. Dependencies between options when applicable are specified in notes.

Table 4.2-1: Community Medication Prescription and Dispense - Actors and Options

Actor	Options	Vol. & Section
Community Pharmacy Manager	*Provision of Medication List*	PHARM TF-1: 4.2.1
	Persistence of Retrieved Documents[14]	PHARM TF-1: 4.2.2
Prescription Placer	*Workflow Management*	PHARM TF-1: 4.2.3
Pharmaceutical Adviser	*Workflow Management*	PHARM TF-1: 4.2.3
Medication Dispenser	*Workflow Management*	PHARM TF-1: 4.2.3

[14] Only applicable, if option “Provision of Medication List” is supported.

Actor	Options	Vol. & Section
Repository Actor Prescription Pharmaceutical Advice Dispense	*No options defined*	- -

4.2.1 Provision of Medication List Option

A Community Pharmacy Manager Actor implementing this option offers the ability to query for a Medication List and return an on-demand created version of the Medication List document. See use case described in Volume 1, chapter 4.4.3.

4.2.2 Persistence of Retrieved Documents Option

A Community Pharmacy Manager Actor implementing this option offers the ability to persist the returned on-demand created version of the Medication List document when querying for a Medication List. This option requires the “Provision of Medication List” Option to be implemented. See use case described in Volume 1, chapter 4.4.3 and ITI TF-1: 10.2.7.

4.2.3 Workflow Management Option

An actor implementing this option offers the ability to manage workflow according to the Workflow Definitions described in Volume 2, chapter 4.

4.3 CMPD Actor Groupings and Profile Interactions

Actor	Groups with	Note
Prescription Placer	Content Creator: PRE Content Consumer: PRE, PADV, DIS, PML	The Prescription Placer Actor shall create Prescription documents according to the PRE Content Profile.
Pharmaceutical Adviser	Content Creator: PADV Content Consumer: PRE, PADV, DIS, PML	The Pharmaceutical Adviser Actor shall create Pharmaceutical Advice documents according to the PADV Content Profile.
Medication Dispenser	Content Creator: DIS Content Consumer: PRE, PADV, DIS, PML	The Medication Dispenser Actor shall create Medication Dispense documents according to the DIS Content Profile.

Note: All three actors shall also be able to consume Prescription-, Pharmaceutical Advice- and Medication Dispense-documents in order to determine the status of Prescription Items.

4.4 CMPD Process Flow

Current implementations of the community pharmacy process (prescribe & dispense medication) may be categorized in two different alternatives.

The first alternative is the so-called publish & pull. In this model, generally speaking, information is generated by a placer type actor (Prescriber, Pharmaceutical Adviser or

Dispenser) and stored by means of a repository type actor. Other actors retrieve data by pulling it from repositories. This approach may apply to health systems where information is accessed on a centralized basis and, therefore, is made available to a collective of potential users (such as prescriptions available for dispense in any community pharmacy).

The alternative approach is the direct push model where information is sent directly to the actor intended to use it (e.g., prescriptions sent directly to the pharmacy named by the patient) and therefore no information is stored on a centralized basis. This model focuses on direct communication instead of availability to (more) potential users.

The current revision of the Integration Profile covers use cases relying on the publish & pull model only.

Workflow scenarios

The CMPD Process Flow can be principally differentiated in two basic workflow scenarios, one including a validation step by a Pharmaceutical Adviser Actor and another excluding it:

- **Scenario 1: Including a validation step by a Pharmaceutical Adviser**
- **Scenario 2: Not including a validation step by a Pharmaceutical Adviser**

A domain using CMPD has to define in which workflow scenario it operates. Workflow scenarios cannot be used compounded.

Any software implementations of the CMPD Profile have to be able to operate in both workflow scenarios.

4.4.1 Use Case community pharmacy-active substance, publish & pull (Scenario 1: “Including validation step”)

The purpose of this use case is to illustrate the prescription-dispense process in community pharmacy when the prescriber orders an active-substance (generic) medicine in the publish & pull model.

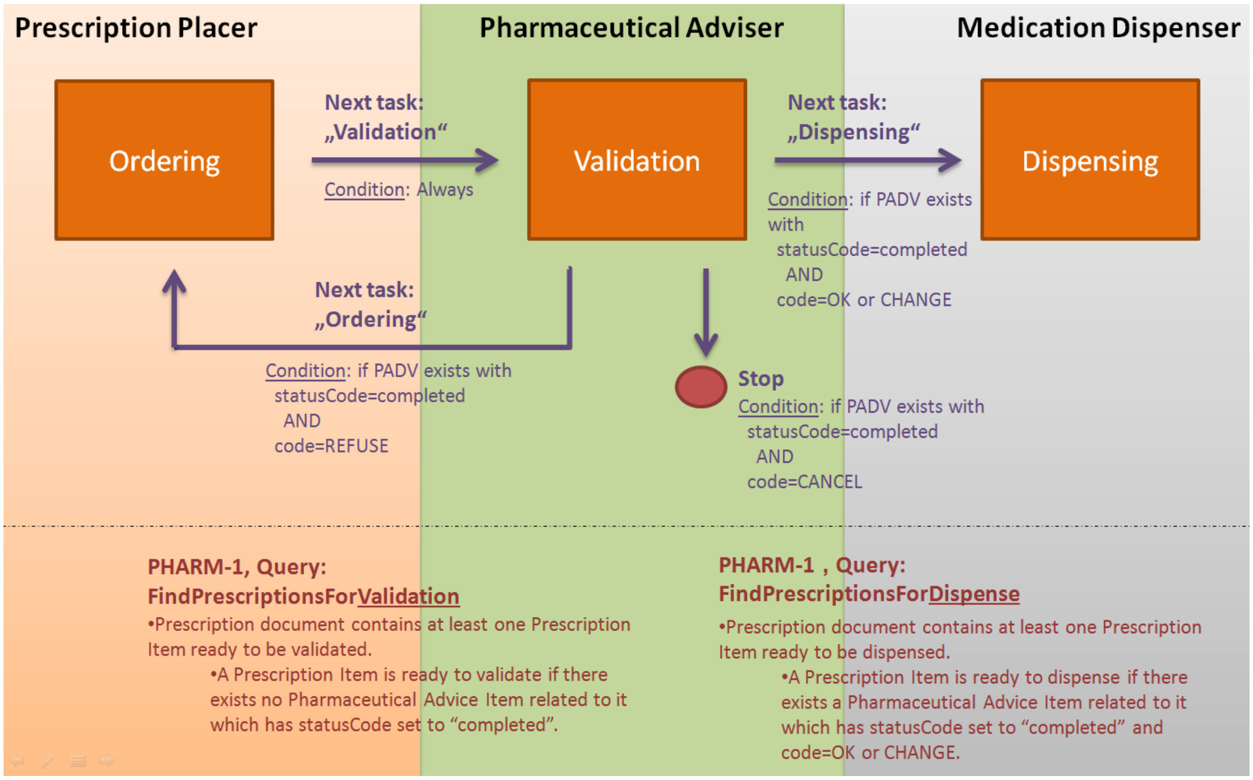
The process of this use case includes the validation step performed by a Pharmaceutical Adviser Actor.

The following diagram shows the workflow of this use case and illustrates the overall context of

- … (workflow) tasks
  - Ordering, Validation, Dispensing
- … which actor performs the task
  - Prescription Placer, Pharmaceutical Adviser, Medication Dispenser
- … the conditions leading to the next task

- In some cases depending on the outcome of the pharmaceutical validation documented in a Pharmaceutical Advice document (see PADV Profile)

440 - … on which task-transition each query of transaction PHARM-1 is used and which business rule it has to follow
    - FindPrescriptionsForValidation (by the Pharmaceutical Adviser)
    - FindPrescriptionsForDispense (by the Medication Dispenser)


Prescription Placer
Pharmaceutical Adviser
Medication Dispenser
Ordering
Next task: „Validation“
Condition: Always
Validation
Next task: „Dispensing“
Condition: if PADV exists with statusCode=completed AND code=OK or CHANGE
Dispensing
Next task: „Ordering“
Condition: if PADV exists with statusCode=completed AND code=REFUSE
Stop
Condition: if PADV exists with statusCode=completed AND code=CANCEL

PHARM-1, Query: FindPrescriptionsForValidation

- Prescription document contains at least one Prescription Item ready to be validated.
    - A Prescription Item is ready to validate if there exists no Pharmaceutical Advice Item related to it which has statusCode set to "completed".

PHARM-1 , Query: FindPrescriptionsForDispense

- Prescription document contains at least one Prescription Item ready to be dispensed.
    - A Prescription Item is ready to dispense if there exists a Pharmaceutical Advice Item related to it which has statusCode set to "completed" and code=OK or CHANGE.

445 **Figure 4.4.1-1: Scenario 1: Overall context of the workflow**

This workflow is implicitly specified by the narrative descriptions in both this profile as well as in the Pharmacy Pharmaceutical Advice Profile.

Note that software implementations shall be able to perform it in any case, whether or not actors
450 are grouped with option "Workflow Management" (grouping with that option does not change the actual workflow, it just allows "technical" workflow management).

Please refer to:

- Community Prescription and Dispense (CMPD) Profile
    - Volume 2, chapter 3.1.4.1.2.1.1.1 FindPrescriptionsForValidation

455 - Volume 2, chapter 3.1.4.1.2.1.1.2 FindPrescriptionsForDispense

  - In case of grouping with XDW: Volume 2, chapter 4 Workflow Definitions
- Pharmacy Pharmaceutical Advice (PADV) Profile
  - Vol. 2, chapter 6.3.4.3.3.6 Status Code
  - Vol. 2, chapter 6.3.4.3.3.4 Observation Code

4.4.1.1 Story Board

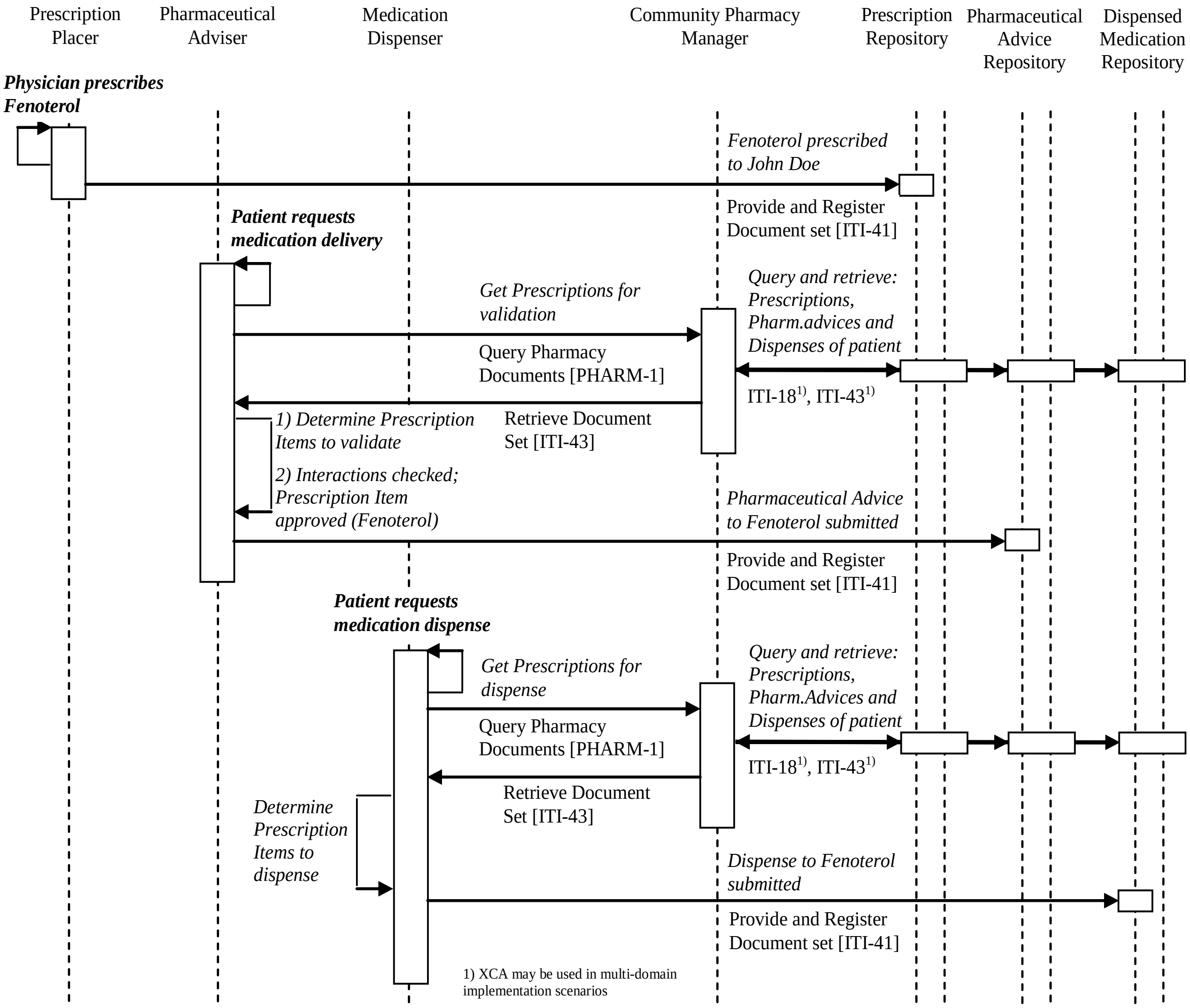
John Doe attends a consultation to his general practitioner, GP, because he is experiencing some breathing difficulty. The practitioner examines John and prescribes the active substance “Fenoterol” in his “Prescription Placer” software. The prescription is electronically sent to the “Prescription Repository”.

Since prescriptions are available to a wide range of pharmacies, John picks the pharmacy closest to his office. The pharmacist asks for John’s health card in order to retrieve the patient’s active prescriptions (from the Community Pharmacy Manager). Since John also suffers from arthritis he has been prescribed Ibuprofen. The pharmacists checks for interactions and finds nothing outstanding. The information on the pharmaceutical advice is electronically sent to the “Pharmaceutical Advice Repository”.

He consults his inventory and picks Berotec® which is in the range of prices approved by the health system. He gives out this medicine to the patient and records the transaction in the “Medication Dispenser”. The information on the medication dispensed is electronically sent to the “Dispensed Medication Repository”.

4.4.1.2 Sequence Diagram

The following diagram represents the sequence of data exchanged between “system actors” involved in this use case.


Prescription Placer
Pharmaceutical Adviser
Medication Dispenser
Community Pharmacy Manager
Prescription Repository
Pharmaceutical Advice Repository
Dispensed Medication Repository
Physician prescribes Fenoterol
Fenoterol prescribed to John Doe
Provide and Register Document set [ITI-41]
Patient requests medication delivery
Get Prescriptions for validation
Query Pharmacy Documents [PHARM-1]
Query and retrieve: Prescriptions, Pharm.advices and Dispenses of patient
ITI-18 1), ITI-43 1)
1) Determine Prescription Items to validate
Retrieve Document Set [ITI-43]
2) Interactions checked; Prescription Item approved (Fenoterol)
Pharmaceutical Advice to Fenoterol submitted
Provide and Register Document set [ITI-41]
Patient requests medication dispense
Get Prescriptions for dispense
Query Pharmacy Documents [PHARM-1]
Query and retrieve: Prescriptions, Pharm.Advices and Dispenses of patient
ITI-18 1), ITI-43 1)
Retrieve Document Set [ITI-43]
Determine Prescription Items to dispense
Dispense to Fenoterol submitted
Provide and Register Document set [ITI-41]
1) XCA may be used in multi-domain implementation scenarios

480

Figure 4.4.1.2-1: Use Case community pharmacy-active substance, publish & pull - Process Flow (Scenario 1: "Including validation step")

This diagram illustrates the complete workflow of the prescription of a medication, the successful validation of the Prescription Item and the dispense of the medication.

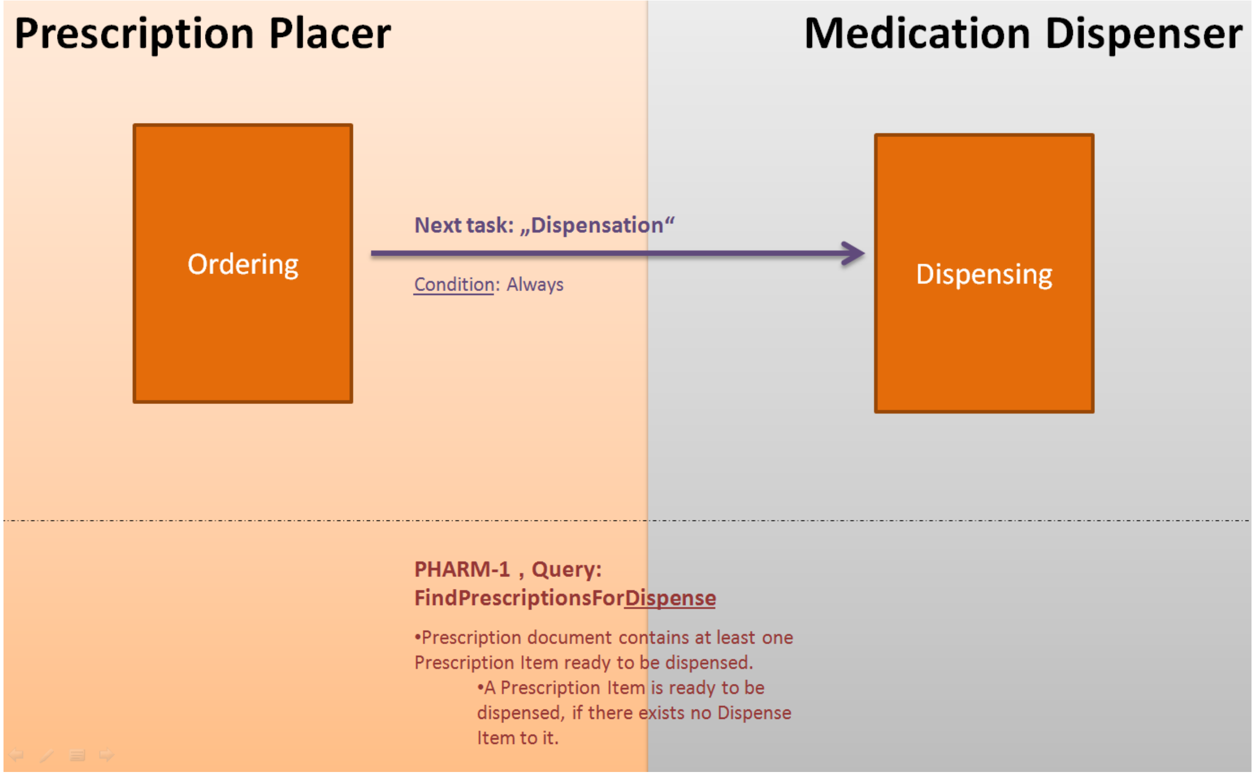
485 4.4.2 Use Case community pharmacy-active substance, publish & pull (Scenario 2: "Not including validation step")

The purpose of this use case is to illustrate the prescription-dispense process in community pharmacy when the prescriber orders an active-substance (generic) medicine in the publish & pull model.

The process of this use case does not include the validation step performed by a Pharmaceutical Adviser Actor.

The following diagram shows the workflow of this use case and illustrates the overall context of

- … (workflow) tasks
  - Ordering, Dispensing
- … which actor performs the task
- Prescription Placer, Medication Dispenser
- … the conditions leading to the next task
- … on which task-transition each query of transaction PHARM-1 is used and which business rule it has to follow
  - FindPrescriptionsForDispense (by the Medication Dispenser)



PHARM-1 , Query: FindPrescriptionsForDispense

•Prescription document contains at least one Prescription Item ready to be dispensed.

•A Prescription Item is ready to be dispensed, if there exists no Dispense Item to it.

Figure 4.4.2-1: Scenario 2: Overall context of the workflow

This workflow is implicitly specified by the narrative descriptions in this profile.

Note that software implementations shall be able to perform it in any case, whether or not actors are grouped with option “Workflow Management” (grouping with that option does not change the actual workflow, it just allows “technical” workflow management).

Please refer to:

- Community Prescription and Dispense (CMPD) Profile
  - Volume 2, chapter 3.1.4.1.2.1.1.2 FindPrescriptionsForDispense
  - In case of grouping with XDW: Volume 2, chapter 4 Workflow Definitions

4.4.2.1 Story Board

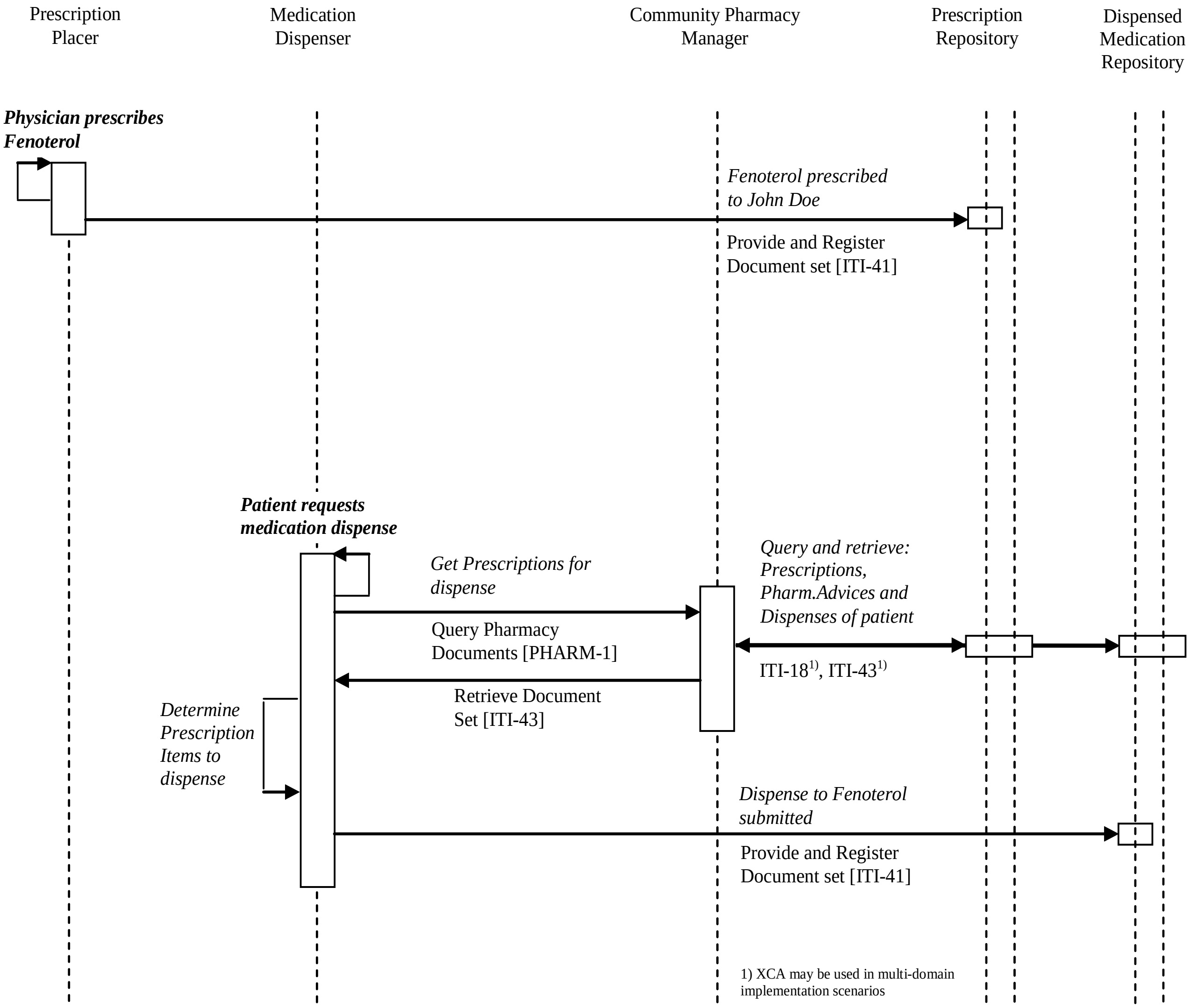
John Doe attends a consultation to his general practitioner, GP, because he is experiencing some breathing difficulty. The practitioner examines John and prescribes the active substance “Fenoterol” in his “Prescription Placer” software. The prescription is electronically sent to the “Prescription Repository”.

Since prescriptions are available to a wide range of pharmacies, John picks the pharmacy closest to his office. The pharmacist asks for John’s health card in order to retrieve the patient’s active prescriptions (from the Community Pharmacy Manager).

He consults his inventory and picks Berotec® which is in the range of prices approved by the health system. He gives out this medicine to the patient and records the transaction in the “Medication Dispenser”. The information on the medication dispensed is electronically sent to the “Dispensed Medication Repository”.

4.4.2.2 Sequence Diagram

The following diagram represents the sequence of data exchanged between “system actors” involved in this use case.


Prescription Placer
Medication Dispenser
Community Pharmacy Manager
Prescription Repository
Dispensed Medication Repository
Physician prescribes Fenoterol
Fenoterol prescribed to John Doe
Provide and Register Document set [ITI-41]
Patient requests medication dispense
Get Prescriptions for dispense
Query Pharmacy Documents [PHARM-1]
Query and retrieve: Prescriptions, Pharm.Advices and Dispenses of patient
ITI-18 1), ITI-43 1)
Retrieve Document Set [ITI-43]
Determine Prescription Items to dispense
Dispense to Fenoterol submitted
Provide and Register Document set [ITI-41]
1) XCA may be used in multi-domain implementation scenarios

530

Figure 4.4.2.2-1: Use Case community pharmacy-active substance, publish & pull - Process Flow (Scenario 2: "Not including validation step")

535 This diagram illustrates the complete workflow of the prescription and the dispense of the medication.

4.4.3 Use Case: Physician requests Medication List

The purpose of this use case is to illustrate the process of requesting the Medication List by a physician. This requires the support of the "Provision of Medication List" Option at the
540 Community Pharmacy Manager.

4.4.3.1 Story Board

John Doe attends a consultation to his general practitioner, GP, because he is experiencing some breathing difficulty. The practitioner examines John and wants to prescribe the active substance “Fenoterol” in his “Prescription Placer” software.

To ensure that there are no conflicts between the new medication and the patient’s current medication status, the physician requests the Medication List.

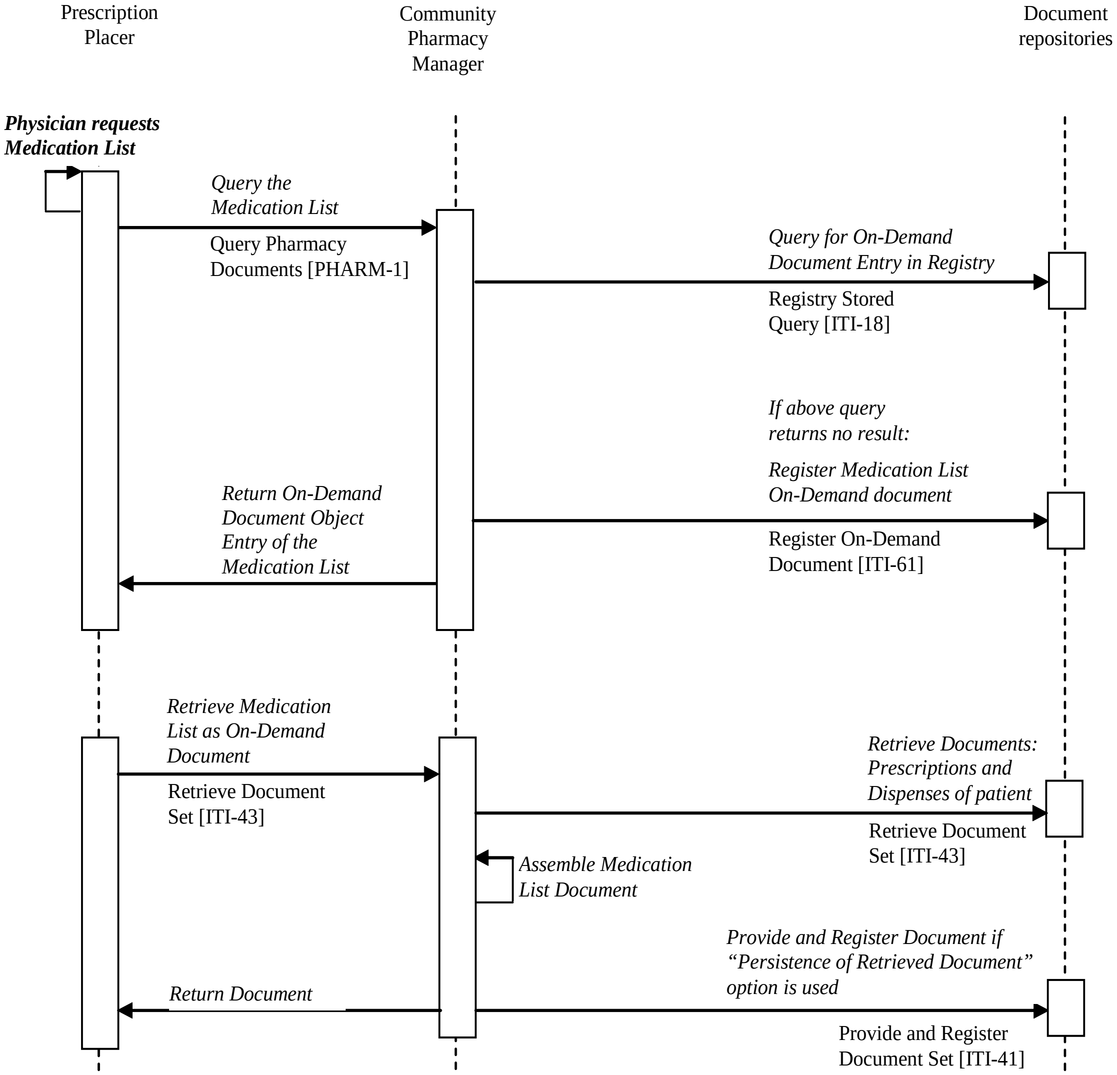
The Prescription Placer Actor uses transaction “Query Pharmacy Document” (PHARM-1) with query “FindMedicationList” to query the Medication List at the Community Pharmacy Manager Actor.

The Community Pharmacy Manager queries the registry for the on-demand document entry of the Medication List to this patient. In case this query does not return a valid result the Community Pharmacy Manager uses transaction “Register On-Demand Document” (ITI-61) to register the Medication List On-Demand Document. Either the found or just created Document Entry will be returned to the calling Prescription Placer Actor.

The Prescription Placer Actor now uses transaction “Retrieve Document Set” (ITI-43) to retrieve the actual Medication List Document from the Community Pharmacy Manager. The Community Pharmacy Manager Actor uses ITI XDS transactions to query and retrieve Prescription-, Pharmaceutical Advice- and Dispense Documents from the repositories in order to assemble the Medication List Document. Once the document is assembled it returns the document to the calling Prescription Placer Actor. If the “Persistence of Retrieved Documents” Option is used the returned document is also provided and registered in the registry/repository backend.

565 4.4.3.2 Sequence Diagram

The following diagram represents the sequence of data exchanged between "system actors" involved in this use case.


Prescription Placer
Community Pharmacy Manager
Document repositories
Physician requests Medication List
Query the Medication List
Query Pharmacy Documents [PHARM-1]
Query for On-Demand Document Entry in Registry
Registry Stored Query [ITI-18]
If above query returns no result:
Register Medication List On-Demand document
Register On-Demand Document [ITI-61]
Return On-Demand Document Object Entry of the Medication List
Retrieve Medication List as On-Demand Document
Retrieve Document Set [ITI-43]
Retrieve Documents: Prescriptions and Dispenses of patient
Retrieve Document Set [ITI-43]
Assemble Medication List Document
Provide and Register Document if "Persistence of Retrieved Document" option is used
Return Document
Provide and Register Document Set [ITI-41]

570

Figure 4.4.3-1: Use Case community pharmacy-active substance, publish & pull - Process Flow (Scenario 2: "Not including validation step")

This diagram illustrates the complete workflow of the process of requesting the Medication List.

4.5 CMPD Security Considerations

Relevant XDS Affinity Domain Security background is discussed in the XDS Security Considerations Section (see ITI TF-1: 10.7).

4.6 CMPD Implementation Scenarios

The following chapter describes several implementation scenarios for the Community Prescription and Dispense Integration Profile.

The prescription and dispense process of real-world projects involves several parties acting in the different abstract roles (Prescription Placer, Pharmaceutical Adviser, Medication Dispenser). The Prescription Placer role is usually taken by physicians; the Pharmaceutical Adviser and Medication Dispenser role is usually taken by pharmacists, which both are usually organized in different organizations.

This results in a wide variety of implementation requirements together with the need of not only organizational but also technical separation of systems. Physicians may want to store prescriptions in another repository than pharmacists the dispenses. In a strict separation even the use of separate IHE affinity domains is required to arrange a throughout distinct scenario. CMPD was designed to be used in either ***single-domain*** or ***multi-domain*** scenarios to fit to these requirement.

Any political intended separation has to be technically bridged at one point otherwise a common prescription and dispense process cannot be established. To minimize the possible points of contact between the domains the Community Pharmacy Manager was introduced.

Explanation to the diagrams used in the following implementation scenario chapters:

- Dotted lines mean separation of concerns
- Different background colors mean different XDS affinity domains

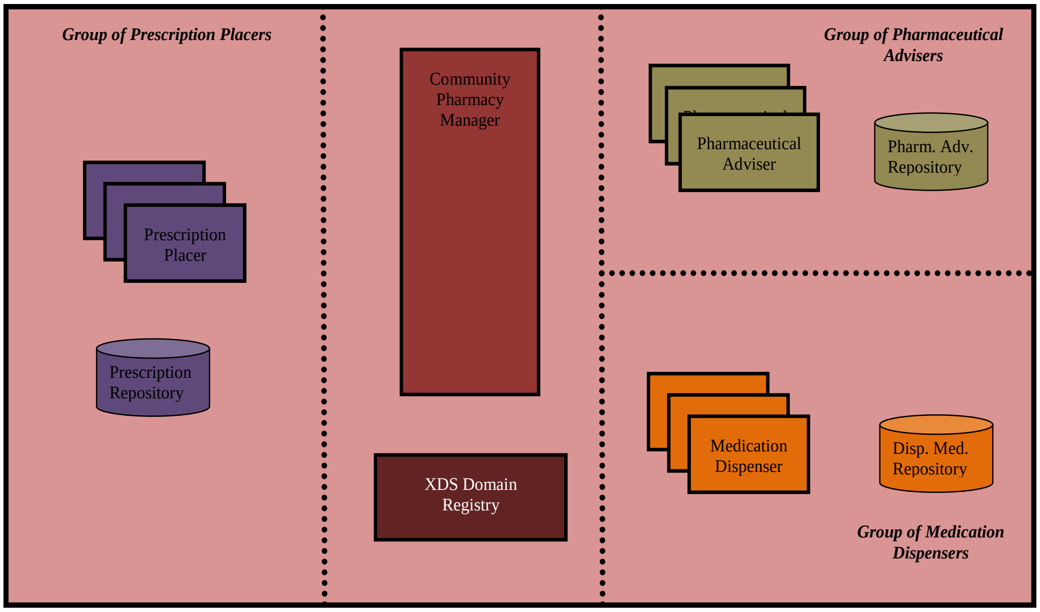
4.6.1 Usage of CMPD in a “single-domain” scenario

The descriptions of CMPD in the previous chapters are aligned to the usage of the profile in a scenario where all actors are hosted in a ***single XDS Affinity domain***.

Operating within a single XDS Affinity domain is the most simple implementation scenario and has several benefits, like e.g., that just one registry holds any document metadata, which eases query, retrieving and publishing of documents, etc.

On the other hand a simple scenario like this may not be applicable to scenarios in reality, where organizational, strategical or political reasons require more separation between the participating parties (physicians, pharmacists).

The following diagram shows a simple example of a single-domain implementation scenario to demonstrate the capabilities of CMPD.



Description of the example scenario

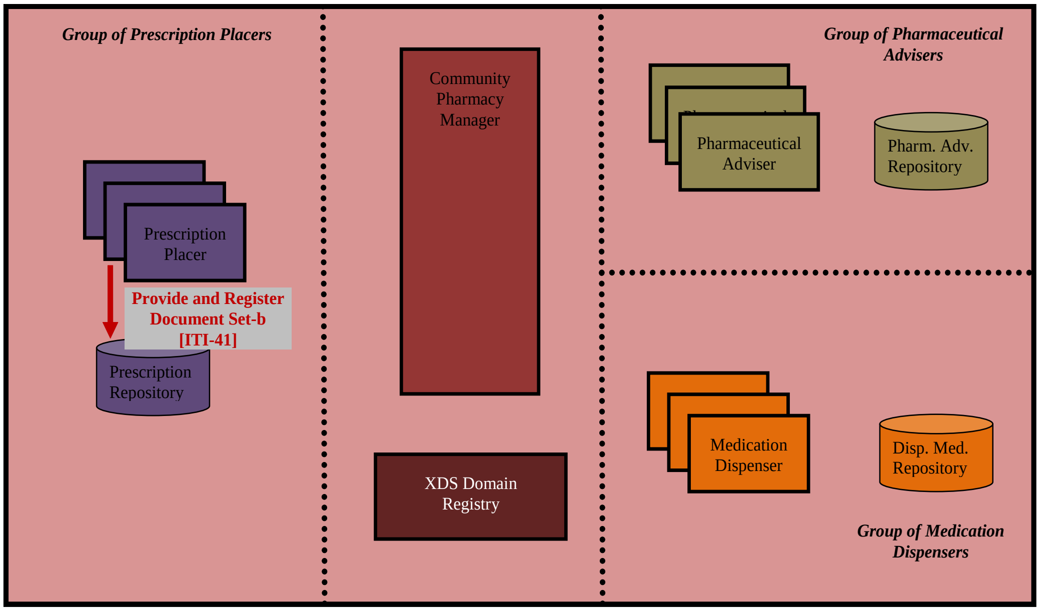
The group of Prescription Placers, Pharmaceutical Advisers and Medication Dispensers are altogether located in one XDS affinity domain. Each group stores its documents in its own dedicated repository, but all use the same document registry of the affinity domain.[15]

[15] In an even more simplified scenario the different document repositories could be merged into one single document repository, but this would not change the principles of the example.

4.6.1.1 Demonstration of use case 1 in example scenario (simple)

Step 1: Prescription Placer creates a prescription

The Prescription document is submitted to the appropriate Prescription Repository.

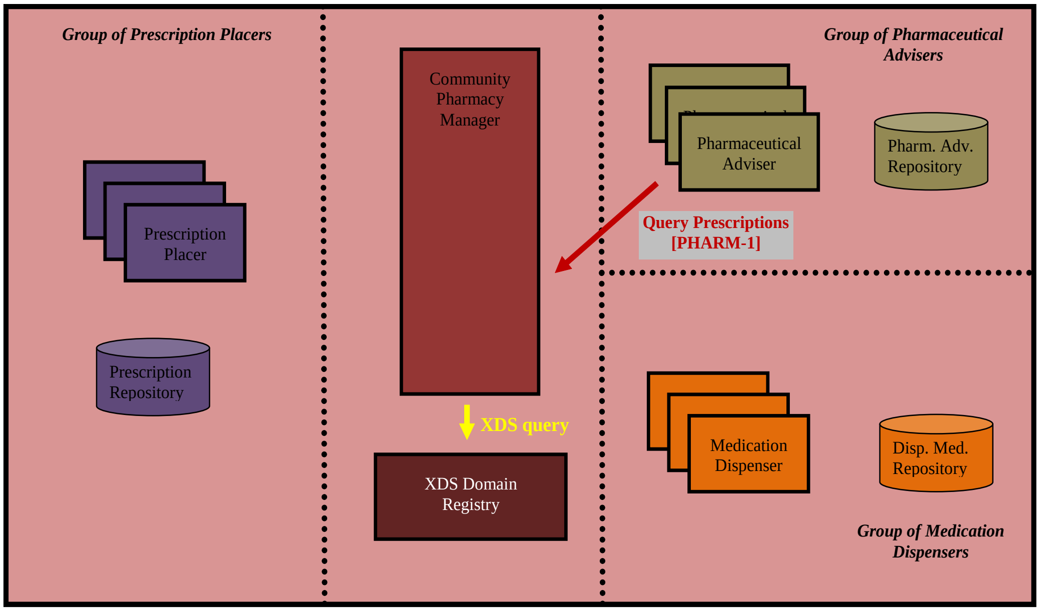

Group of Prescription Placers
Prescription Placer
Provide and Register Document Set-b [ITI-41]
Prescription Repository
Community Pharmacy Manager
XDS Domain Registry
Group of Pharmaceutical Advisers
Pharmaceutical Adviser
Pharm. Adv. Repository
Medication Dispenser
Disp. Med. Repository
Group of Medication Dispensers

Step 2: Pharmaceutical Adviser queries the prescription

The Pharmaceutical Adviser queries the prescription by using transaction PHARM-1, query “**FindPrescriptionsForValidation**”.

The CPM queries the common XDS domain registry for prescription, pharmaceutical advice and dispense documents. Then it retrieves all these documents from the appropriate document repositories.

After retrieving it does linking of the documents by their document IDs and determines the status of each prescription. It applies appropriate filtering according to the semantic question “for Validation” and returns just “relevant” document UUIDs to the Pharmaceutical Adviser Actor, which proceeds with step 3.

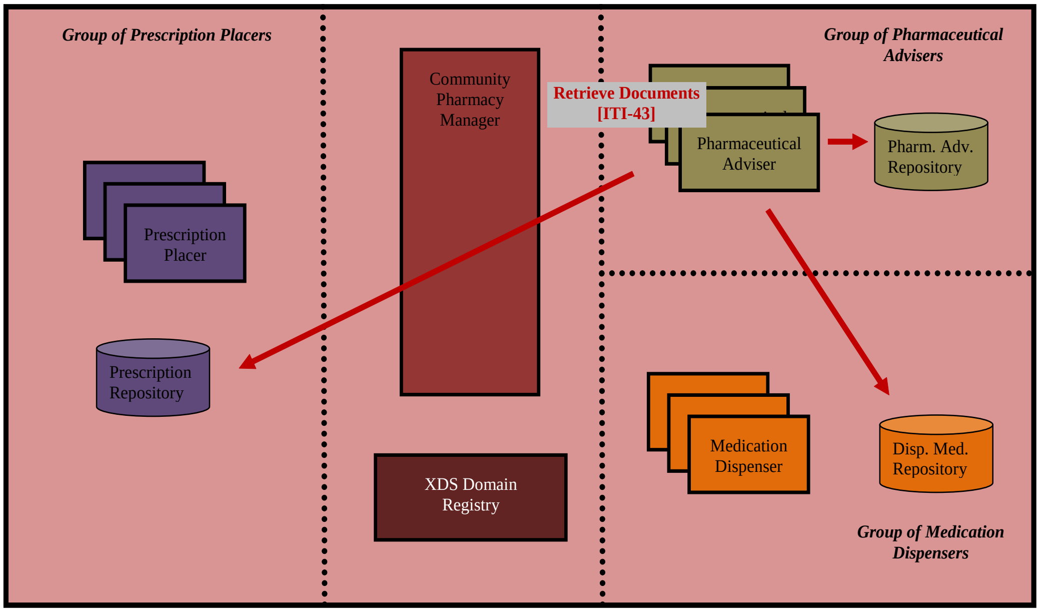


Step 3: Pharmaceutical Adviser retrieves the documents of the query result

650 The Pharmaceutical Adviser Actor asks the CPM to retrieve all documents identified by the returned document UUIDs from the according document repositories.

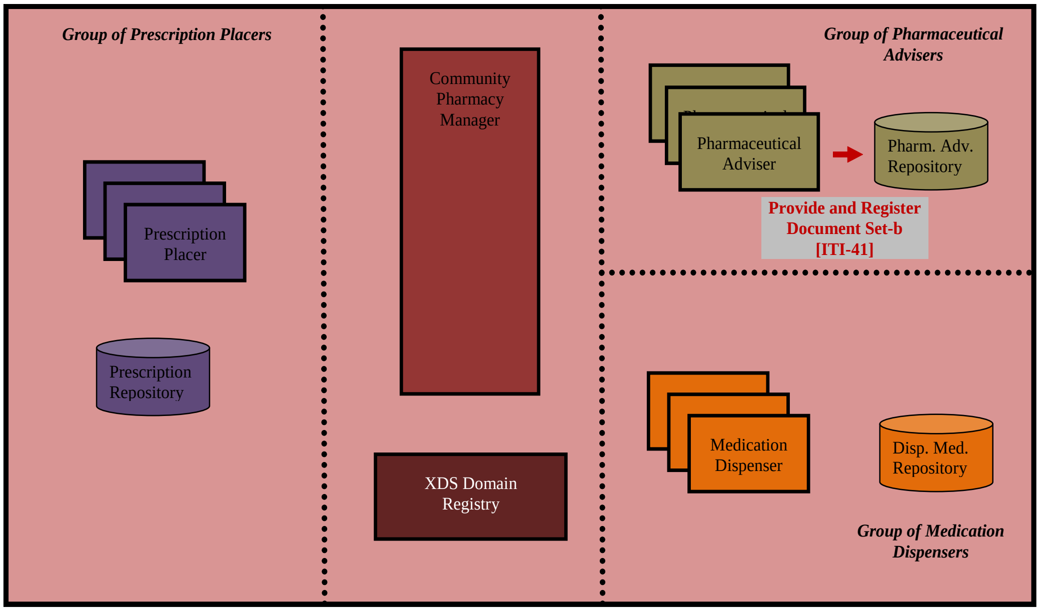
The Pharmaceutical Adviser Actor (as a machine) parses and relinks the returned documents by their document IDs. Then the system or the human operator performs validation and proceeds with step 4.

655



Step 4: Pharmaceutical Adviser submits a pharmaceutical advice

After the validation step the outcome of the validation is documented in a Pharmaceutical Advice document. This document is submitted to the appropriate Pharmaceutical Advice Repository.

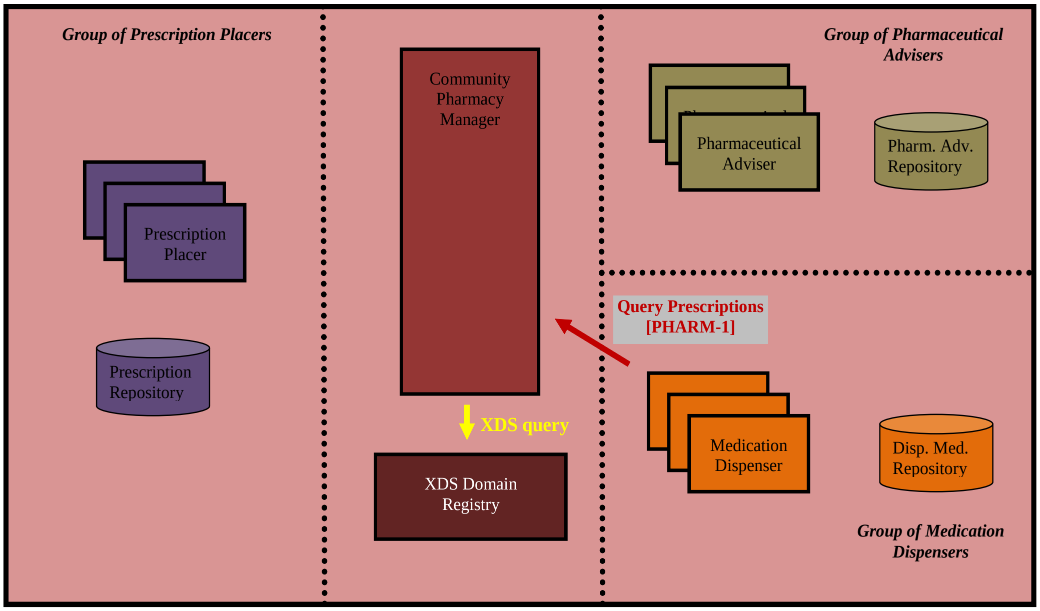

Group of Prescription Placers
Prescription Placer
Prescription Repository
Community Pharmacy Manager
XDS Domain Registry
Group of Pharmaceutical Advisers
Pharmaceutical Adviser
Pharm. Adv. Repository
Provide and Register Document Set-b [ITI-41]
Medication Dispenser
Disp. Med. Repository
Group of Medication Dispensers

Step 5: Medication Dispenser queries the prescription

The Medication Dispenser queries the prescription by using transaction PHARM-1, query "**FindPrescriptionsForDispense**".

Analog to step 2, the CPM queries the XDS domain registry for prescription, pharmaceutical advice and dispense documents. Then it retrieves all these documents from the appropriate document repositories.

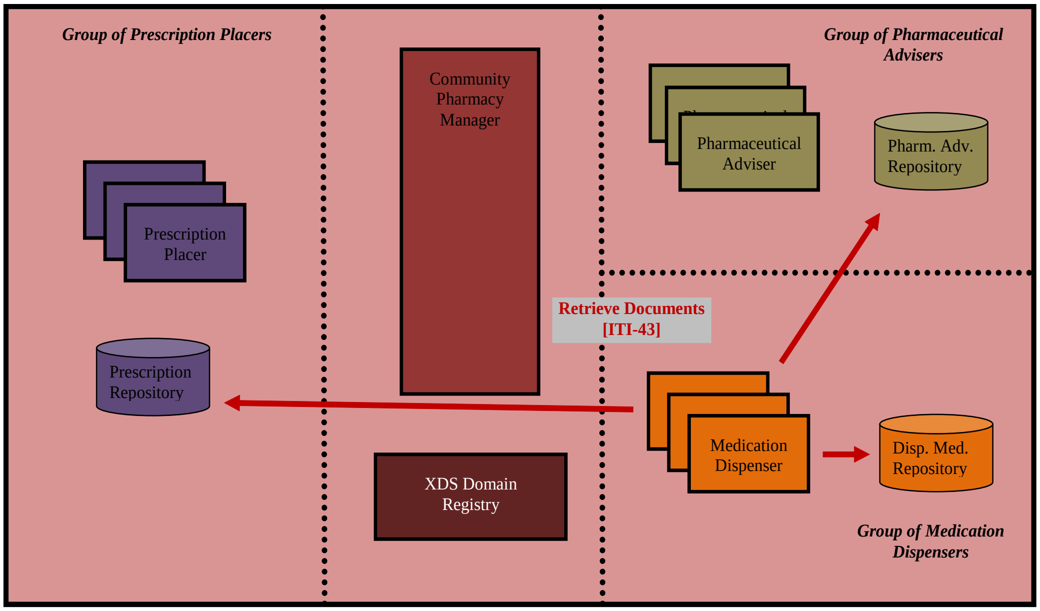
After retrieving it performs linking of the documents by their document IDs and determines the status. It applies appropriate filtering according to the semantic question "for Dispense", and returns just "relevant" document UUIDs to the Medication Dispenser Actor, which proceeds with step 6.



Step 6: Medication Dispenser retrieves the documents of the query result

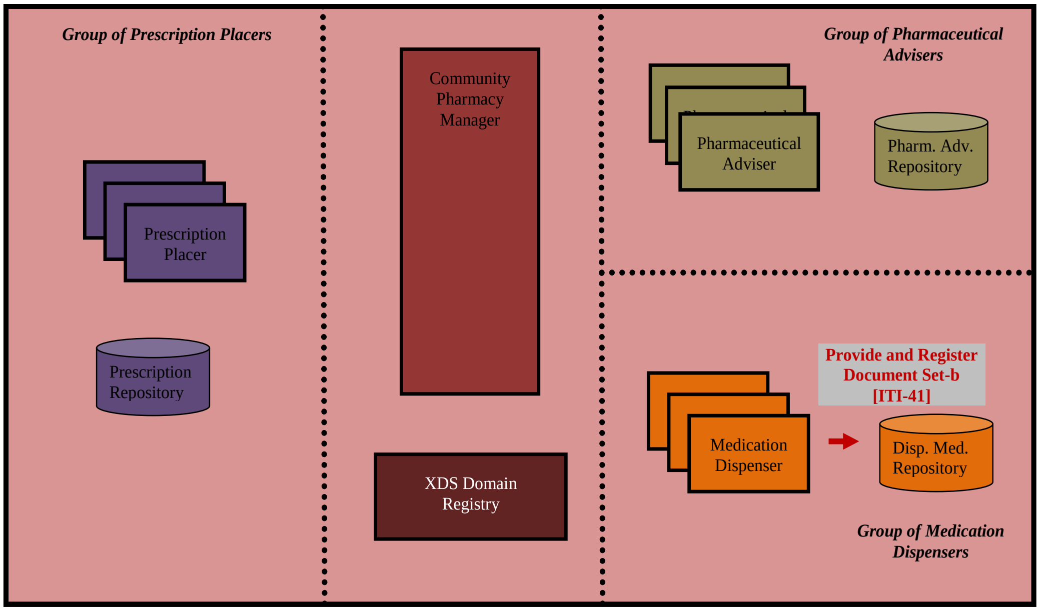
The Medication Dispenser Actor asks the CPM to retrieve all documents identified by the returned document UUIDs from the according document repositories.

The Medication Dispenser Actor (as a machine) parses and relinks the returned documents by their document IDs. Then the human operator performs the dispense and proceeds with step 7.



Step 7: Medication Dispenser submits a dispense

After the dispense has taken place it is documented in a Dispense document. This document is submitted to the appropriate Dispensed Medication Repository.


Group of Prescription Placers
Prescription Placer
Prescription Repository
Community Pharmacy Manager
XDS Domain Registry
Group of Pharmaceutical Advisers
Pharmaceutical Adviser
Pharm. Adv. Repository
Provide and Register Document Set-b [ITI-41]
Medication Dispenser
Disp. Med. Repository
Group of Medication Dispensers

4.6.2 Usage of CMPD in a “multi-domain” scenario

The descriptions of CMPD in the previous chapters are aligned to the usage of the profile in a scenario where all actors are hosted in a single XDS Affinity domain. Nevertheless the profile can also be used in ***multi XDS Affinity domain*** scenarios.

Operating within a scenario consisting of multiple XDS Affinity domains is a complex but rather realistic implementation scenario. Its main benefit is that a minimum of technical contact is required between the participating parties of such a system (physicians, pharmacists) for achieving technical interoperability. Such utmost separation might be an organizational, strategical or political requirement.

Such an implementation scenario requires the usage of the CPM’s “relaying” functionality shown in the following *more detailed* Actor Diagram:

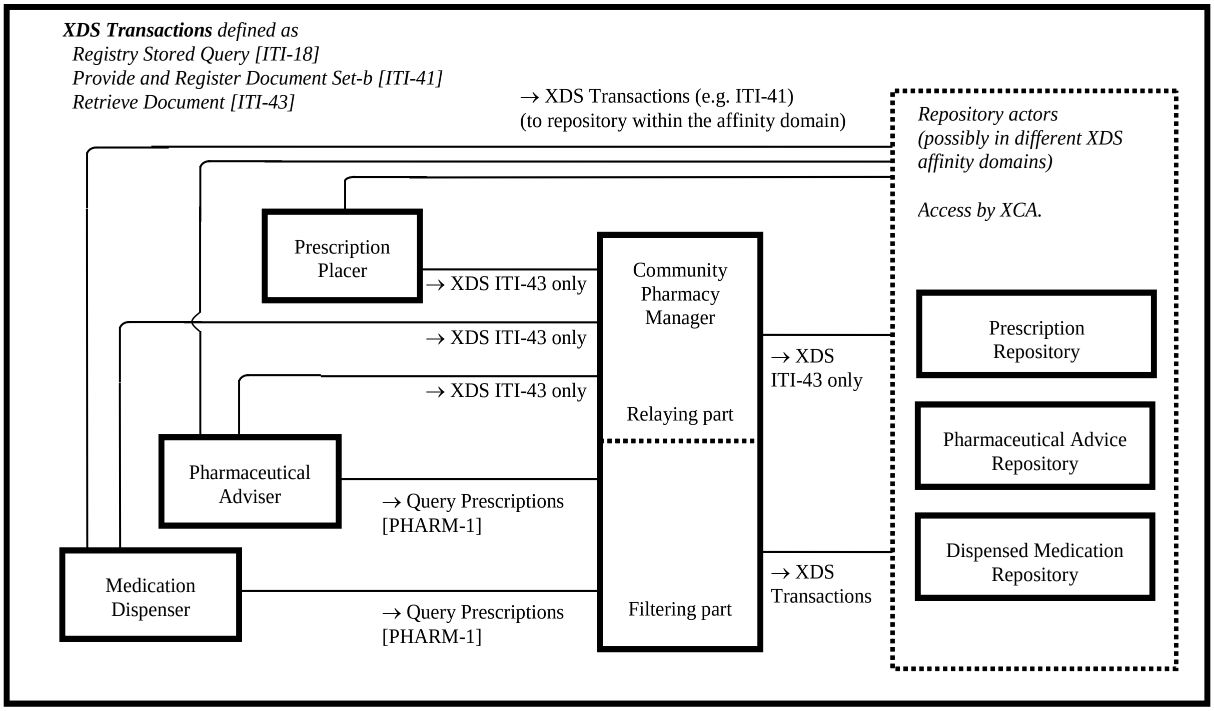
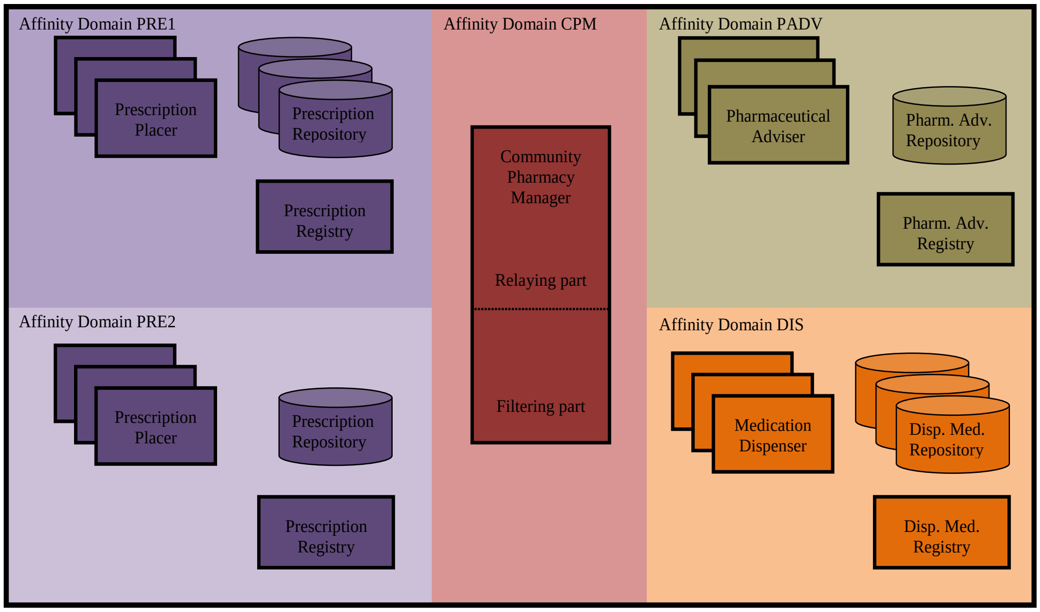

XDS Transactions defined as
Registry Stored Query [ITI-18]
Provide and Register Document Set-b [ITI-41]
Retrieve Document [ITI-43]
→ XDS Transactions (e.g. ITI-41)
(to repository within the affinity domain)
Repository actors
(possibly in different XDS
affinity domains)
Access by XCA.
Prescription Placer
→ XDS ITI-43 only
Community Pharmacy Manager
→ XDS ITI-43 only
→ XDS ITI-43 only
Relaying part
→ XDS
ITI-43 only
Prescription Repository
Pharmaceutical Advice Repository
Dispensed Medication Repository
Pharmaceutical Adviser
→ Query Prescriptions
[PHARM-1]
Medication Dispenser
→ Query Prescriptions
[PHARM-1]
Filtering part
→ XDS
Transactions

Figure 4.6.2-1: More detailed Actor Diagram in a multi-domain scenario

The following diagram shows an example of a possible multi-domain implementation scenario to demonstrate the capabilities of CMPD.



715

Description of the example scenario

The group of Prescription Placers divides into 2 separate domains, the first (PRE1) showing a federated architecture with multiple repositories, the second (PRE2) with all clients connected to one. 720

The group of Pharmaceutical Advisers is organized in an own affinity domain (PADV), all storing in one repository.

The group of Medication Dispensers are all organized in a common affinity domain (DIS), but everyone stores its dispenses in their own application (also acting as repository).

725 ***All these different domains accept the XDS Affinity domain of the Community Pharmacy Manager (CPM) as the point of intersection to which they all maintain trusted relationships to. Bi-lateral trusts are not required.***

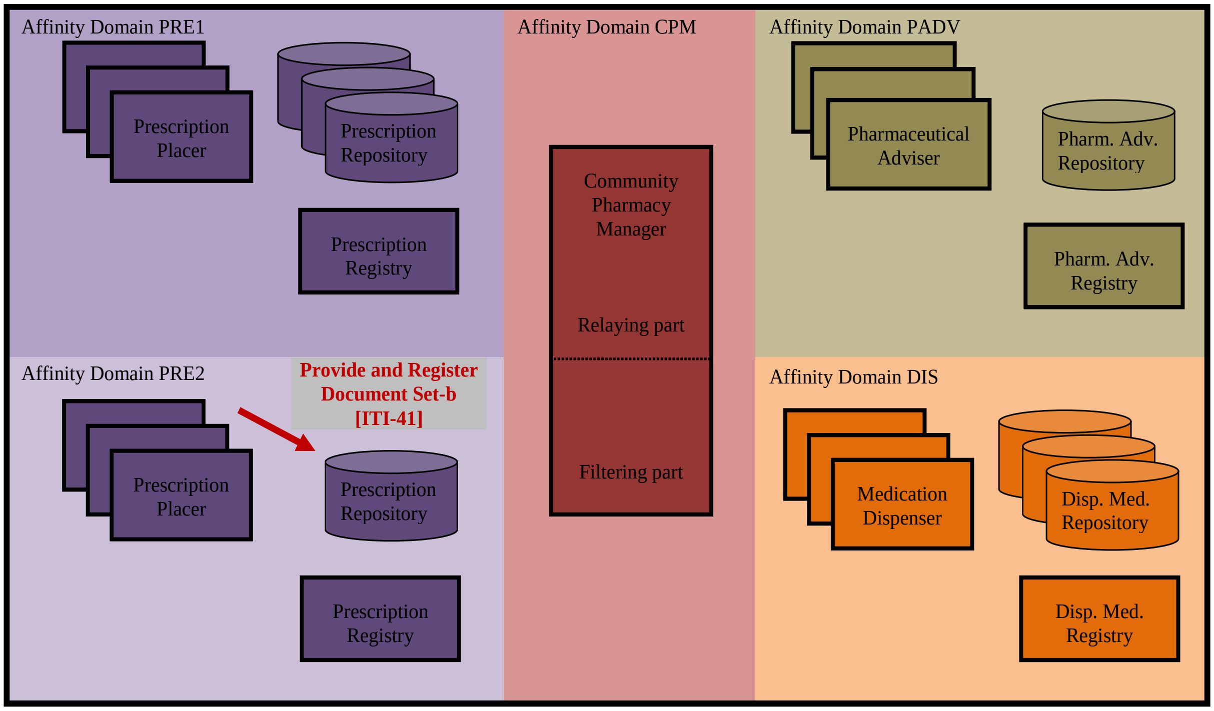
730

4.6.2.1 Demonstration of use case 1 in example scenario (complex)

Step 1: Prescription Placer creates a prescription

The Prescription document is submitted to the appropriate Prescription Repository.

735

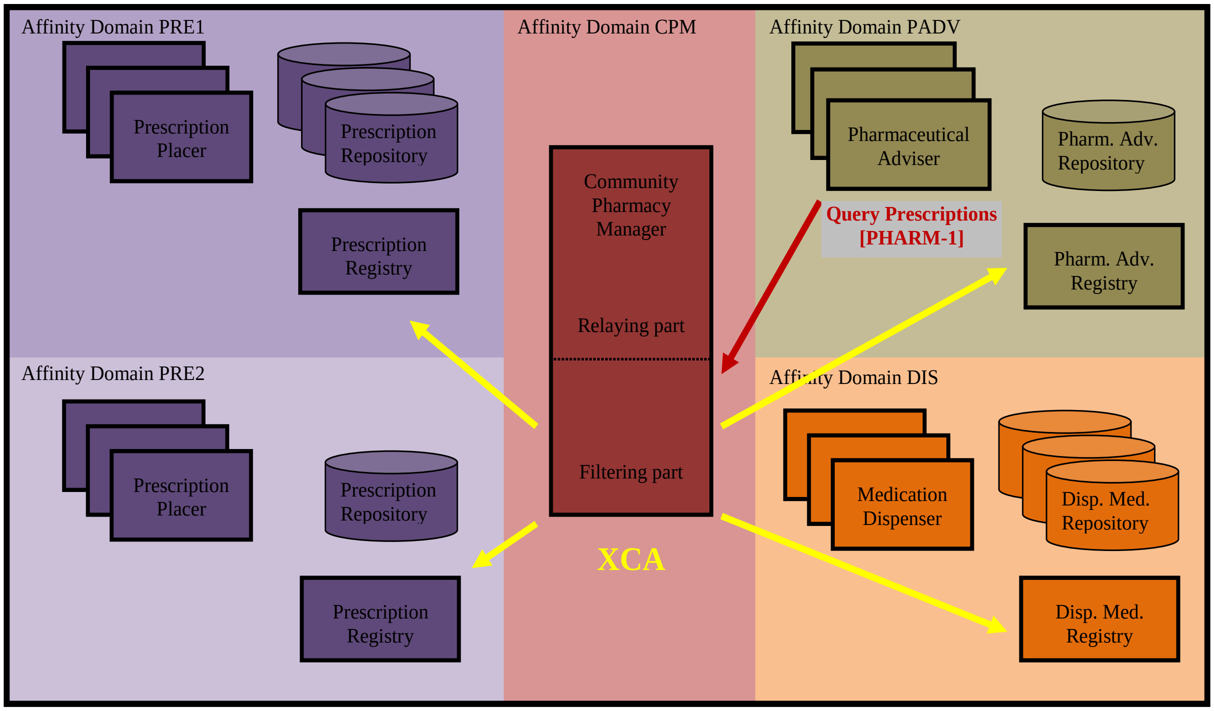


Step 2: Pharmaceutical Adviser queries the prescription

The Pharmaceutical Adviser queries the prescription by using transaction PHARM-1, query "**FindPrescriptionsForValidation**".

In this complex scenario the CPM has to use XCA mechanisms to query all other domains for prescription, pharmaceutical advice and dispense documents. Then it retrieves all these documents from the appropriate document repositories.

After retrieving it does linking of the documents by their document IDs and determines the status of each prescription. It applies appropriate filtering according to the semantic question "for Validation" and returns just "relevant" document UUIDs to the Pharmaceutical Adviser Actor, which proceeds with step 3.

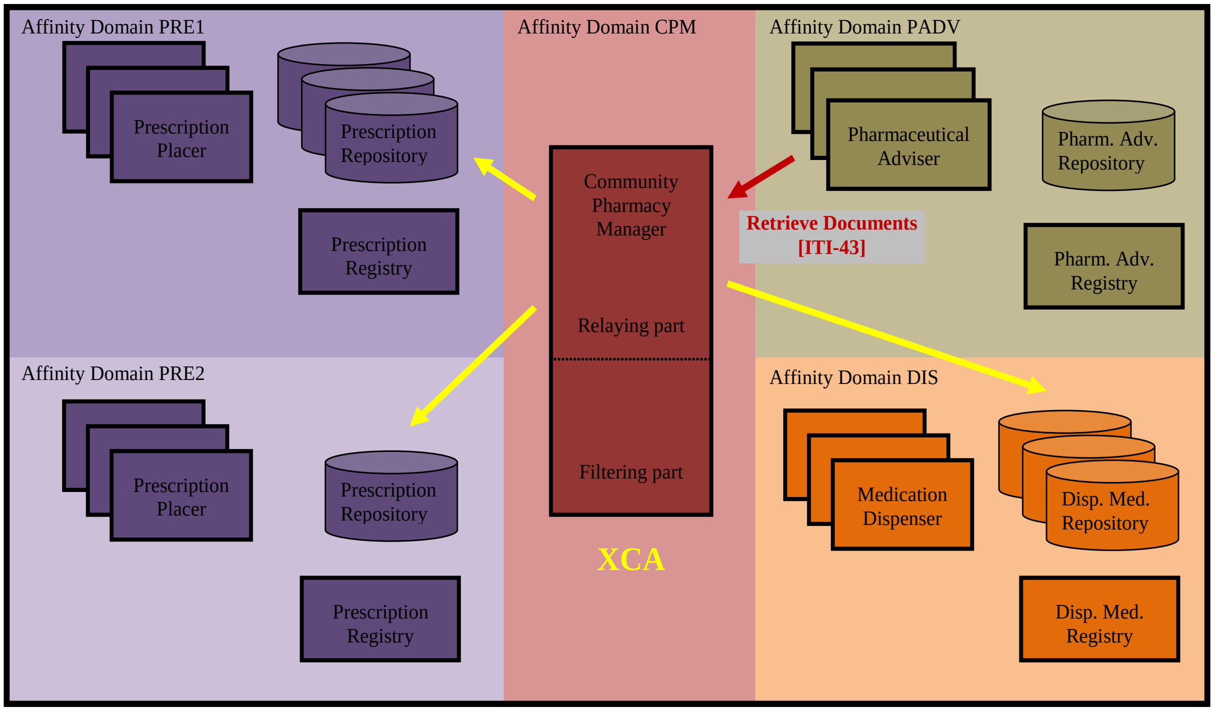


Step 3: Pharmaceutical Adviser retrieves the documents of the query result

The Pharmaceutical Adviser Actor asks the CPM to retrieve all documents identified by the returned document UUIDs.

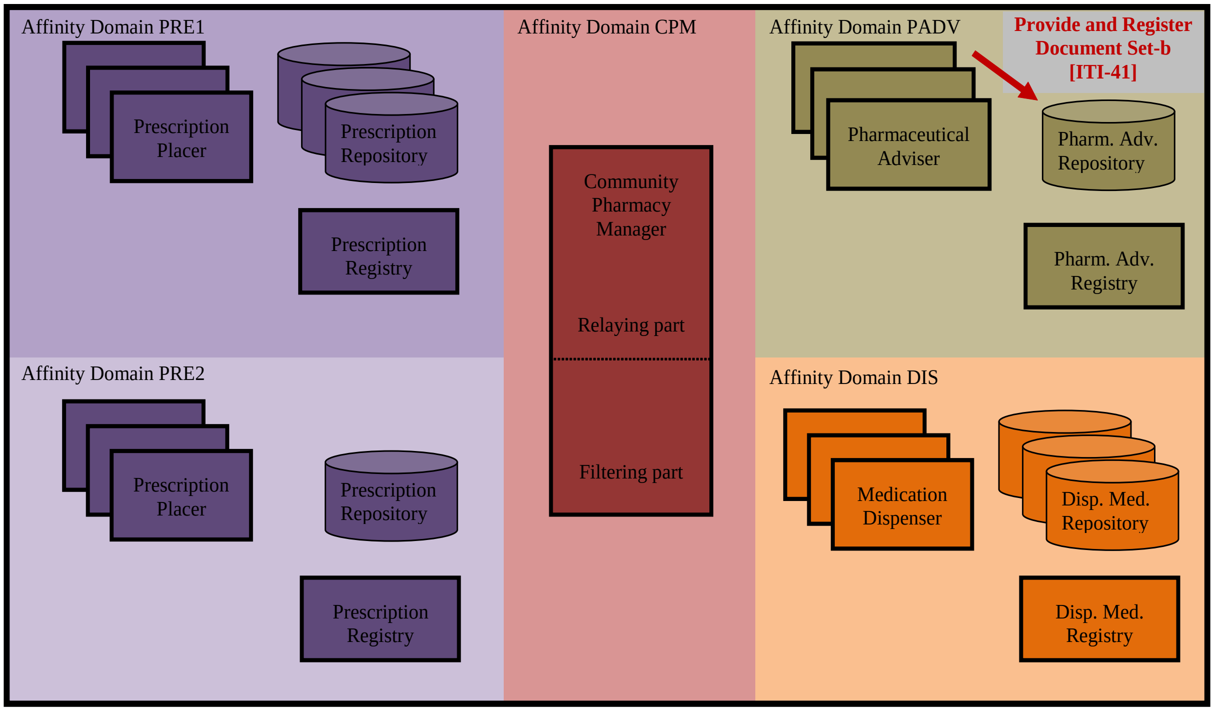
The CPM acts as a relaying entity and accesses all requested repositories for retrieving the documents by XCA. Then it returns them to the calling client. Note that the Pharmaceutical Adviser Actor has no need to have access to the other domains (which could be organizational prohibited).

The Pharmaceutical Adviser Actor (as a machine) parses and relinks the returned documents by their document IDs. Then the system or the human operator performs validation and proceeds with step 4.



765 Step 4: Pharmaceutical Adviser submits a pharmaceutical advice

After the validation step the outcome of the validation is documented in a Pharmaceutical Advice document. This document is submitted to the appropriate Pharmaceutical Advice Repository.



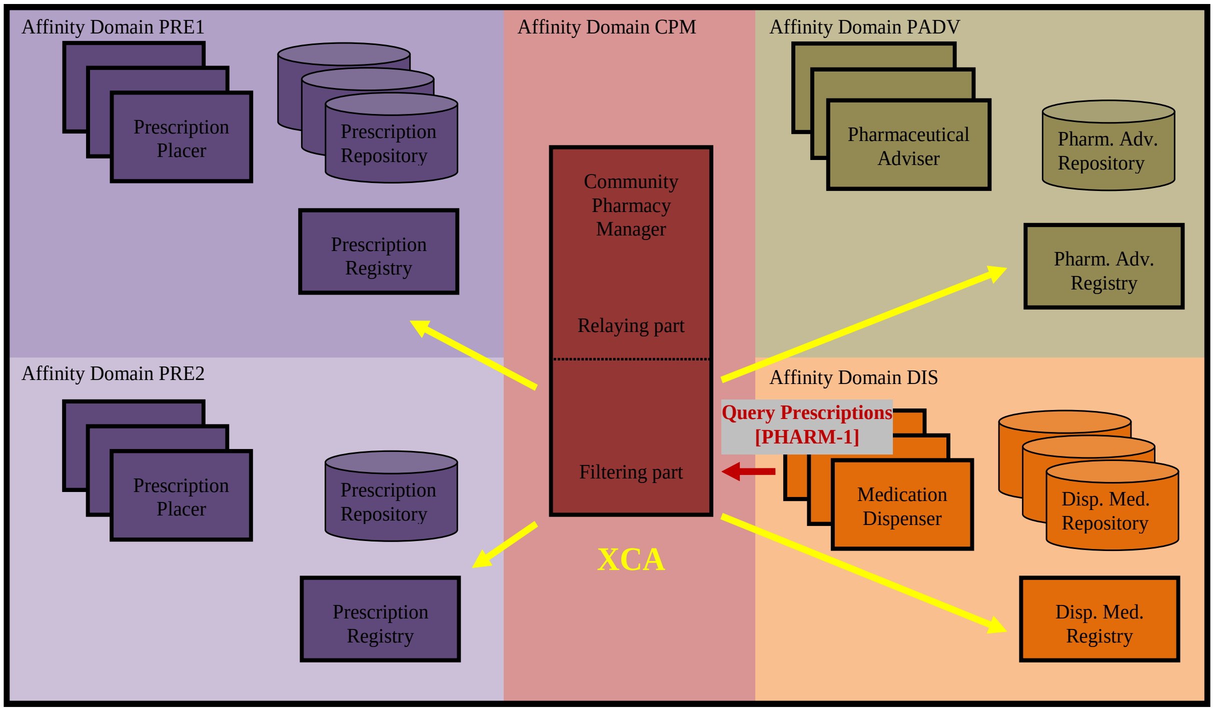
770

Step 5: Medication Dispenser queries the prescription

The Medication Dispenser queries the prescription by using transaction PHARM-1, query "**FindPrescriptionsForDispense**".

Analog to step 2, the CPM uses XCA mechanisms to query all other domains for prescription, pharmaceutical advice and dispense documents. Then it retrieves all these documents from the appropriate document repositories.

After retrieving it performs linking of the documents by their document IDs and determines the status. It applies appropriate filtering according to the semantic question "for Dispense", and returns just "relevant" document UUIDs to the Medication Dispenser Actor, which proceeds with step 6.

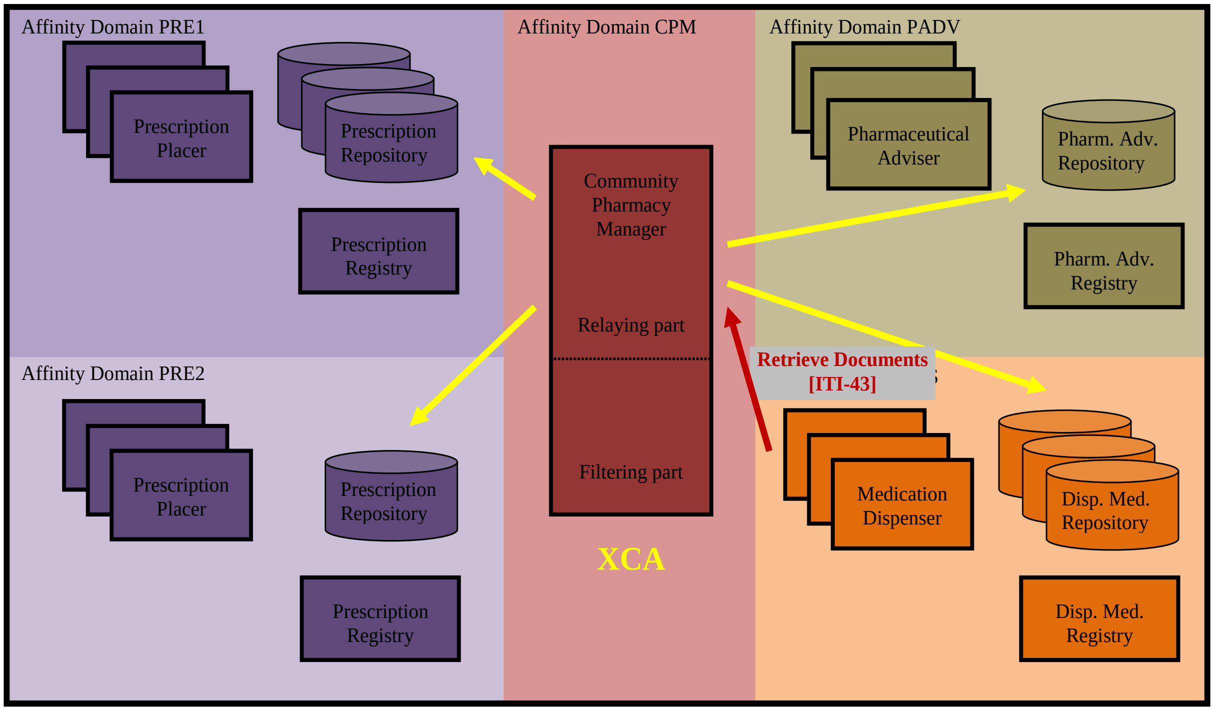


Step 6: Medication Dispenser retrieves the documents of the query result

The Medication Dispenser Actor asks the CPM to retrieve all documents identified by the returned document UUIDs.

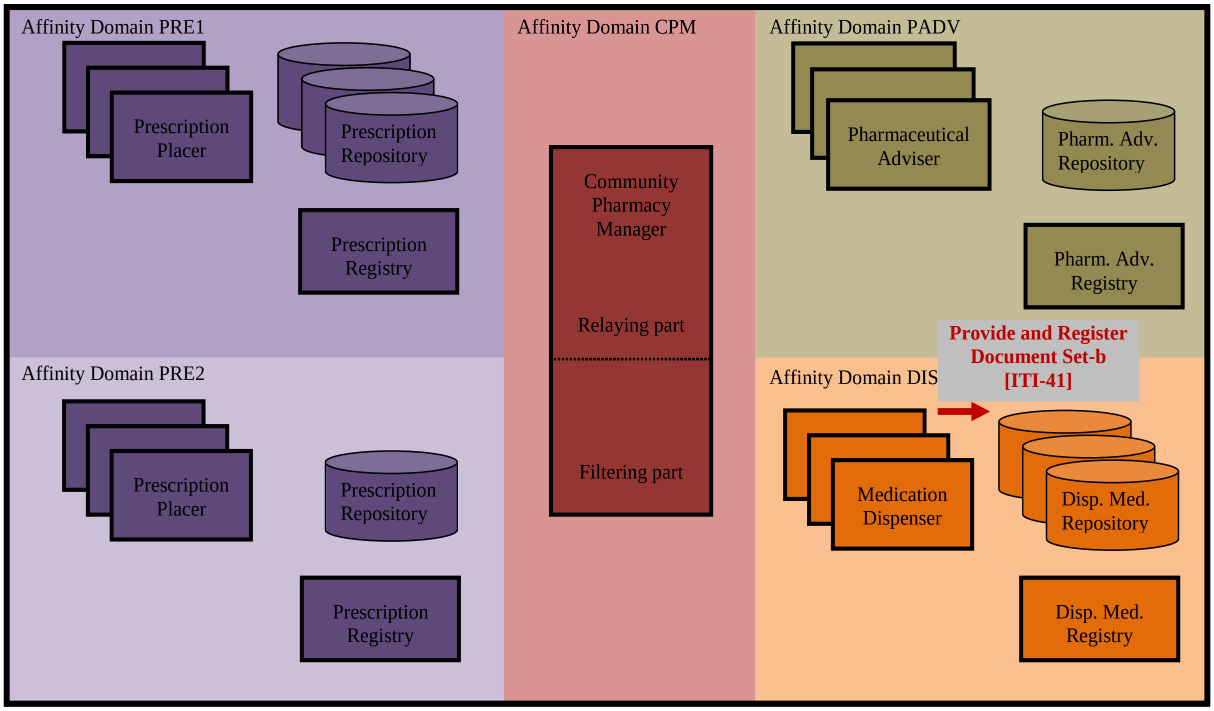
The CPM acts as a relaying entity and accesses all requested repositories for retrieving the documents by XCA. Then it returns them to the calling client. Note that the Medication Dispenser Actor has no need to have access to the other domains (which could be organizational prohibited).

The Medication Dispenser Actor (as a machine) parses and relinks the returned documents by their document IDs. Then the human operator performs the dispense and proceeds with step 7.



Step 7: Medication Dispenser submits a dispense

After the dispense has taken place it is documented in a Dispense document. This document is submitted to the appropriate Dispensed Medication Repository.



Appendix A Actor Summary Definitions

Community Pharmacy Manager - Actor providing the business logic for status management and other purposes. As a second role it acts as a “relaying role” where certain standard XDS communication is routed through for providing the possibility of applying project-specific business logic on it.

Prescription Placer - Actor for placing prescriptions (initial or modified in case of a substitution of invalidation, for example). It provides Prescription documents containing one or more Prescription Items representing the prescribed medication.

Pharmaceutical Adviser - Actor responsible for the validation of prescriptions from a pharmacist’s perspective. It sends provides the Pharmaceutical Advice document as the result of the validation. Pharmaceutical Advisers (e.g., automated ICA check modules) may also provide “draft” advices which don’t affect the status of a prescription but serve as a foundation for the advice performed by another Pharmaceutical Adviser.

Medication Dispenser - Actor responsible for the process of dispensing medication to the patient, fulfilling the prescription. It receives prescriptions already validated and provides a dispense document as result of the act of delivering the medication to the patient.

Registry/Repository actors - Formally the Community Pharmacy process defines three different “repositories” for Prescriptions, Pharmaceutical Advices and Dispenses. They shall be seen as abstract repository-roles for persisting the appropriate document types the documents. This profile makes use of the XDS Profile for defining abstract XDS registry and repository actors for modeling the abstract repository-roles for real implementations.

Appendix B Transaction Summary Definitions

Query Pharmacy Documents - This transaction defines how a querying actor has to query the Community Pharmacy Manager for prescriptions (PRE) and their related documents. Related documents are Pharmaceutical Advice (PADV) and Dispense (DIS) documents. It defines specialized queries allow the finding of prescriptions and their related documents for specific purposes (e.g., “for validation” or “for dispense”).

Registry Stored Query - See the XDS Integration Profile of the ITI Technical Framework for a detailed description of this transaction (ITI-TF2a:3.18)

Provide and Register Document Set-b - See the XDS Integration Profile of the ITI Technical Framework for a detailed description of this transaction (ITI-TF2a:3.41)

Retrieve Document Set - See the XDS Integration Profile of the ITI Technical Framework for a detailed description of this transaction (ITI-TF2a:3.43)

Volume 2 – Transactions

3.0 IHE Transactions

Add Section 3.1

3.1 Query Pharmacy Documents [PHARM-1]

This transaction defines how a querying actor has to query the Community Pharmacy Manager for prescriptions (PRE) and their related documents. Related documents are Pharmaceutical Advice (PADV) and Dispense (DIS) documents.

Specialized queries allow the finding of prescriptions and their related documents for specific purposes (e.g., for validation).

Querying actors may be:

- Pharmaceutical Adviser
- Medication Dispenser

This transaction is very similar to the concept of the Registry Stored Query (ITI-18) transaction in the XDS Integration Profile of the ITI Technical Framework, except that the query itself targets not a single registry (like described in the XDS Integration Profile) but shall be able to sub-query one to many registry/repository systems (by using XCA in case of multi-domain scenarios) to get the requested query result.

The querying actor faces the same interface as if querying a XDS Document registry actor, although the query result may contain references to documents of many different domains.

3.1.1 Scope

The Query Pharmacy Documents transaction supports the following specialized queries:

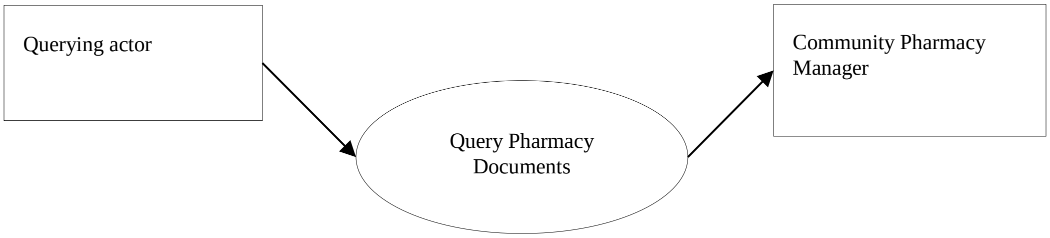
- **FindPrescriptions**
    - Find prescriptions and their related documents
- **FindDispenses**
    - Find dispense documents and their related documents
- **FindPrescriptionsForValidation**
    - Find prescriptions and their related documents containing Prescription Items ready to be validated
- **FindPrescriptionsForDispense**

  - Find prescriptions and their related documents containing Prescription Items ready to be dispensed
- **FindMedicationList (if „Provision of Medication List“ Option is supported)**
  - Find the medication list to the patient

All queries return:

- Metadata for one or more registry objects, or
- Object references for one or more registry objects (registry UUIDs).

3.1.2 Use Case Roles



Actors: Querying actor

Role: Requests a query by identifier (UUID), and passes parameters to the query. A parameter controlling the format of the returned data is passed; it selects either object references or full objects.

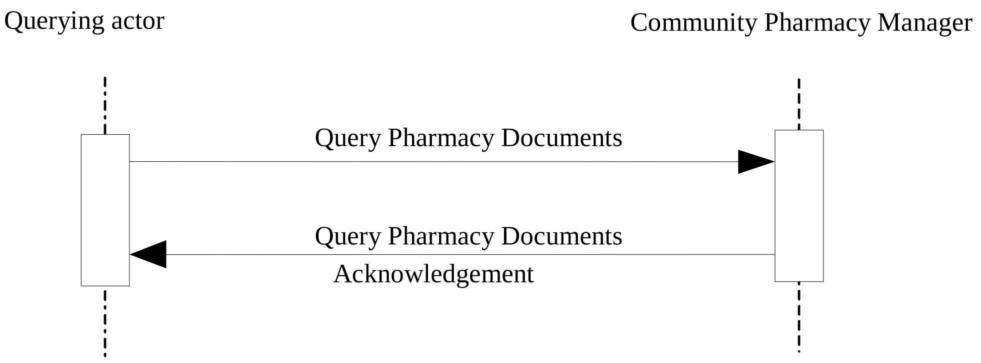
Actor: Community Pharmacy Manager

Role: Services the query using its stored definitions of the queries defined for CMPD.

3.1.3 Referenced Standard

ITI-18: Registry Stored Query and all its related standards.

3.1.4 Interaction Diagram



3.1.4.1 Query Pharmacy Documents

This is a query request to the Community Pharmacy Manager from a Querying actor. The query request contains:

- A reference to a pre-defined query stored on the Document Registry Actor.
- Parameters to the query. The query parameters are matched up with the query variables defined in the query definition on the Document Registry Actor.

3.1.4.1.1 Trigger Events

This message is initiated when the Querying actor wants to query/retrieve document metadata.

This may be the case, if:

1. A Prescription Placer, Pharmaceutical Adviser or Medication Dispenser wants to find prescriptions (and their related pharmaceutical advices and dispenses) for general purpose.
2. A Prescription Placer, Pharmaceutical Adviser or Medication Dispenser wants to find dispense documents (and their related pharmaceutical advices and prescriptions) for general purpose.
3. A Pharmaceutical Adviser Actor wants to find active prescriptions (and their related pharmaceutical advices and dispenses) ready to validate.
4. A Medication Dispenser wants to find active prescriptions (and their related pharmaceutical advices and dispenses) which are already validated and ready for dispense.
5. A Prescription Placer, Pharmaceutical Adviser or Medication Dispenser wants to find the Medication List to the patient.

3.1.4.1.2 Message Semantics

The message semantics of this message are based on the definitions of the [ITI-18] transaction, but incorporate some important changes defined in the chapters below.

References to: ITI TF-2a: [ITI-18]

3.1.4.1.2.1 Required Queries

The Registry Stored Query (ITI-18) transaction defines several kinds of Stored Queries (FindDocuments, FindSubmissionSets, etc.).

The PHARM-1 transaction is alike to this concept but provides a different set of Stored Queries.

The provided Stored Queries are:

- **FindPrescriptions**
  - Find prescriptions and their related documents
- **FindDispenses**
  - Find dispense documents and their related documents
- **FindPrescriptionsForValidation**
  - Find prescriptions and their related documents containing Prescription Items ready to be validated
- **FindPrescriptionsForDispense**
  - Find prescriptions and their related documents containing Prescription Items ready to be dispensed
- **FindMedicationList (if „Provision of Medication List“ Option is supported)**
  - Find the medication list to the patient

3.1.4.1.2.1.1 Parameters for Required Queries

This chapter defines the parameters for the Required Queries.

3.1.4.1.2.1.1.1 FindPrescriptions

Find prescriptions and their related documents (XDSDocumentEntry objects) containing Prescription Items for a given patientID and other matching attributes. The other parameters can be used to restrict the set of XDSDocumentEntry objects returned.

Returns: XDSDocumentEntry objects according to the following business rules.

Business rule 1: Returns ***Prescription*** documents matching the query parameters:

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- XDSDocumentEntry matches all other optional query parameters
- FormatCode matches **urn:ihe:pharm:pre:2010**

Business rule 2: Returns related ***Pharmaceutical Advice*** documents to the Prescriptions found

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches **urn:ihe:pharm:padv:2010**
- Pharmaceutical Advice document contains a Pharmaceutical Advice Entry Item related to a Prescription Item of the found Prescription documents.[16]

Business rule 3: Returns related ***Dispense*** documents to the Prescriptions found

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches **urn:ihe:pharm:dis:2010**
- Dispense document contains a Dispense Entry Item related to a Prescription Item of the found Prescription documents.[17]

Explanation

Returning Prescription documents according to business rule 1 is the primary result of the query, where all optional query parameters which might affect the result of the query are applied.

The secondary result of the query, the related Pharmaceutical Advice and Dispense documents to the Prescriptions (Business rule 2 and 3), is dependent on the primary result (found Prescriptions) only and contains just directly related documents.

Query parameters:

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryPatientId	XDSDocumentEntry. patientId	R	--

[16] See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details how the relation between Pharmaceutical Advice Entries and Prescription Items is defined, chapter: "Pharmaceutical Advice Item Entry Content Module" (1.3.6.1.4.1.19376.1.9.1.3.3)

[17] See the Pharmacy Dispense Content Profile (DIS) for details how the relation between Dispense Entries and Prescription Items is defined, chapter: "Dispense Item Entry Content Module" (1.3.6.1.4.1.19376.1.9.1.3.4)

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryEntryUUID	XDSDocumentEntry. entryUUID	O[1]	M
$XDSDocumentEntryUniqueId	XDSDocumentEntry. uniqueId	O[1]	M
$XDSDocumentEntryPracticeSettingCode[3]	XDSDocumentEntry. practiceSettingCode	O	M
$XDSDocumentEntryCreationTimeFrom	Lower value of XDSDocumentEntry. creationTime	O	--
$XDSDocumentEntryCreationTimeTo	Upper value of XDSDocumentEntry. creationTime	O	--
$XDSDocumentEntryServiceStartTimeFrom	Lower value of XDSDocumentEntry. serviceStartTime	O	--
$XDSDocumentEntryServiceStartTimeTo	Upper value of XDSDocumentEntry. serviceStartTime	O	--
$XDSDocumentEntryServiceStopTimeFrom	Lower value of XDSDocumentEntry. serviceStopTime	O	--
$XDSDocumentEntryServiceStopTimeTo	Upper value of XDSDocumentEntry. serviceStopTime	O	--
$XDSDocumentEntryHealthcareFacilityTypeCode[3]	XDSDocumentEntry. healthcareFacilityTypeCode	O	M
$XDSDocumentEntryEventCodeList[3]	XDSDocumentEntry. eventCodeList[3]	O	M
$XDSDocumentEntryConfidentialityCode[3]	XDSDocumentEntry. confidentialityCode[3]	O	M
$XDSDocumentEntryAuthorPerson[4]	XDSDocumentEntry. Author	O	M
$XDSDocumentEntryStatus	XDSDocumentEntry. Status	R	M

[1] Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

[3] Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

[4] The value for this parameter is a pattern compatible with the SQL keyword LIKE which allows the use of the following wildcard characters: % to match any (or no) characters and _ to match a single character. The match shall be applied to the text contained in the Value elements of the authorPerson Slot on the author Classification (value strings of the authorPerson sub-attribute)

Examples for the "FindPrescriptions" query

Assume the following situation of persisted documents in the Prescription-/Pharmaceutical Advice- and Dispense repositories:

Prescriptions		Pharmaceutical Advice	Dispenses	Remark
PRE 1	PRE Item 1-1	PADV 1	DIS 1	This item is already validated and dispensed
	PRE Item 1-2	PADV 2		This item is already validated and ready for dispense
PRE 2	PRE Item 2-1	PADV 3	DIS 2	This item is already validated and dispensed
	PRE Item 2-2	PADV 4		This item is already validated and ready for dispense
	PRE Item 2-3			This item is not validated yet
PRE 3	PRE Item 3-1			This item is not validated yet

Example 1: Standard query

Used Query Parameters:

- Patient ID
- Document Status

This is what should be returned by the query:

Returned XDSDocumentEntries		Explanation
Prescriptions	Related documents	
PRE 1	PADV 1, PADV 2, DIS 1	Prescription 1 is returned with all its related documents
PRE 2	PADV 3, PADV 4, DIS 2	Prescription 2 is returned with all its related documents
PRE 3		Prescription 3 is returned (no related documents available)

Example 2: Search for a specific prescription

Query Parameters set:

- Patient ID
- Document Status
- Document uniqueId of the specific prescription (e.g., because patient showed a paper prescription with the uniqueId printed on it)

In case the uniqueId of PRE 2 is given as query parameter, this is what should be returned by the query:

Returned XDSDocumentEntries		Explanation
Prescriptions	Related documents	
PRE 2	PADV 3, PADV 4, DIS 2	Prescription 2 is returned with all its related documents

3.1.4.1.2.1.1.2 FindDispenses

Find dispense documents and their related documents (XDSDocumentEntry objects) containing Dispense Items for a given patientID and other matching attributes. The other parameters can be used to restrict the set of XDSDocumentEntry objects returned.

Returns: XDSDocumentEntry objects according to the following business rules.

<u>Business rule 1: Returns ***Dispense*** documents matching the query parameters:</u>

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- XDSDocumentEntry matches all other optional query parameters
- FormatCode matches **urn:ihe:pharm:dis:2010**

<u>Business rule 2: Returns related ***Pharmaceutical Advice*** documents to the Dispense documents found</u>

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches **urn:ihe:pharm:padv:2010**
- Pharmaceutical Advice[18] document contains:
  - (1) a Pharmaceutical Advice Entry Item related to a Dispense Item of the found Dispense documents
  - (2) a Pharmaceutical Advice Entry Item related to a Prescription Item on behalf the Dispense Item has been dispensed of the found Prescription documents.

<u>Business rule 3: Returns related ***Prescription*** documents to the Dispense documents found</u>

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches **urn:ihe:pharm:pre:2010**

[18] See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details how the relation between Pharmaceutical Advice Entries and Prescription Items is defined, chapter: “Pharmaceutical Advice Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.3)

- Prescription document contains a Prescription Item on behalf of which the found Dispense Item has been dispensed.[19]

Explanation

Returning Dispense documents according to business rule 1 is the primary result of the query, where all optional query parameters which might affect the result of the query are applied.

The secondary result of the query, the related Pharmaceutical Advice and Prescription documents to the Dispense documents (Business rule 2 and 3), is dependent on the primary result (found Dispense) only and contains just directly related documents.

Query parameters:

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryPatientId	XDSDocumentEntry. patientId	R	--
$XDSDocumentEntryEntryUUID	XDSDocumentEntry. entryUUID	O[1]	M
$XDSDocumentEntryUniqueId	XDSDocumentEntry. uniqueId	O[1]	M
$XDSDocumentEntryPracticeSettingCode[3]	XDSDocumentEntry. practiceSettingCode	O	M
$XDSDocumentEntryCreationTimeFrom	Lower value of XDSDocumentEntry. creationTime	O	--
$XDSDocumentEntryCreationTimeTo	Upper value of XDSDocumentEntry. creationTime	O	--
$XDSDocumentEntryServiceStartTimeFrom	Lower value of XDSDocumentEntry. serviceStartTime	O	--
$XDSDocumentEntryServiceStartTimeTo	Upper value of XDSDocumentEntry. serviceStartTime	O	--
$XDSDocumentEntryServiceStopTimeFrom	Lower value of XDSDocumentEntry. serviceStopTime	O	--
$XDSDocumentEntryServiceStopTimeTo	Upper value of XDSDocumentEntry. serviceStopTime	O	--
$XDSDocumentEntryHealthcareFacilityTypeCode[3]	XDSDocumentEntry. healthcareFacilityTypeCode	O	M

[19] See the Pharmacy Dispense Content Profile (DIS) for details how the relation between Dispense Entries and Prescription Items is defined, chapter: “Dispense Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.4)

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryEventCodeList[3]	XDSDocumentEntry. eventCodeList[3]	O	M
$XDSDocumentEntryConfidentialityCode[3]	XDSDocumentEntry. confidentialityCode[3]	O	M
$XDSDocumentEntryAuthorPerson[4]	XDSDocumentEntry. Author	O	M
$XDSDocumentEntryStatus	XDSDocumentEntry. Status	R	M

[1] Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

[3] Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

[4] The value for this parameter is a pattern compatible with the SQL keyword LIKE which allows the use of the following wildcard characters: % to match any (or no) characters and _ to match a single character. The match shall be applied to the text contained in the Value elements of the authorPerson Slot on the author Classification (value strings of the authorPerson sub-attribute)

Examples for the "FindDispense" query

Assume the following situation of persisted documents in the Prescription-/Pharmaceutical Advice- and Dispense repositories:

Prescriptions		Pharmaceutical Advice	Dispenses	Remark
PRE 1	PRE Item 1-1	PADV 1	DIS 1	This item is already validated and dispensed
	PRE Item 1-2	PADV 2		This item is already validated and ready for dispense
PRE 2	PRE Item 2-1	PADV 3	DIS 2	This item is already validated and dispensed
	PRE Item 2-2	PADV 4		This item is already validated and ready for dispense
	PRE Item 2-3			This item is not validated yet
PRE 3	PRE Item 3-1			This item is not validated yet

<u>Example 1: Standard query</u>

Used Query Parameters:

- Patient ID
- Document Status

This is what should be returned by the query:

Returned XDSDocumentEntries		Explanation
Prescriptions	Related documents	
DIS 1	PADV 1, PADV 2, PRE 1	Dispense 1 is returned with all its related documents
DIS 2	PADV 3, PADV 4, PRE 2	Dispense 2 is returned with all its related documents

Example 2: Search for a specific prescription

Query Parameters set:

- Patient ID
- Document Status
- Document uniqueId of the specific dispense

In case the uniqueId of DIS 2 is given as query parameter, this is what should be returned by the query:

Returned XDSDocumentEntries		Explanation
Prescriptions	Related documents	
DIS 2	PADV 3, PADV 4, PRE 2	Dispense 2 is returned with all its related documents

3.1.4.1.2.1.1.3 FindPrescriptionsForValidation

Find prescriptions and their related documents (XDSDocumentEntry objects) containing Prescription Items ***ready to be validated*** for a given patientID and other matching attributes. The other parameters can be used to restrict the set of XDSDocumentEntry objects returned.

Returns: XDSDocumentEntry objects according to the following business rules.

The business rules are basically depending on the workflow scenario used (see Vol. 1, chapter 4.4 CMPD Process Flow).

Scenario 1 "Including validation step":

Business rule 1.1: Returns ***Prescription*** documents matching the query parameters:

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- XDSDocumentEntry matches all other optional query parameters

- FormatCode matches **urn:ihe:pharm:pre:2010**
- Prescription document contains at least one Prescription Item ready to validate
  - A Prescription Item is ready to validate if there exists no Pharmaceutical Advice Item related to it which has statusCode set to "completed"[20] and the result code equals OK or CHANGE[21].

Business rule 1.2: Returns related ***Pharmaceutical Advice*** documents to the Prescriptions found

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches **urn:ihe:pharm:padv:2010**
- Pharmaceutical Advice document contains a Pharmaceutical Advice Entry Item related to a Prescription Item of the found Prescription documents.[22]

Business rule 1.3: Returns related ***Dispense*** documents to the Prescriptions found

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches **urn:ihe:pharm:dis:2010**
- Dispense document contains a Dispense Entry Item related to a Prescription Item of the found Prescription documents.[23]

Explanation

Returning Prescription documents according to business rule 1.1 is the primary result of the query, where all optional query parameters which might affect the result of the query are applied.

The secondary result of the query, the related Pharmaceutical Advice and Dispense documents to the Prescriptions (Business rule 1.2 and 1.3), is dependent on the primary result (found Prescriptions) only and contains just directly related documents.

[20] See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details about the statusCode element (chapter "Status Code")

[21] See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details about the code element (chapter "Observation Code")

[22] See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details how the relation between Pharmaceutical Advice Entries and Prescription Items is defined, chapter: "Pharmaceutical Advice Item Entry Content Module" (1.3.6.1.4.1.19376.1.9.1.3.3)

[23] See the Pharmacy Dispense Content Profile (DIS) for details how the relation between Dispense Entries and Prescription Items is defined, chapter: "Dispense Item Entry Content Module" (1.3.6.1.4.1.19376.1.9.1.3.4)

<u>Scenario 2 “Not including validation step”:</u>

<u>Business rule 2.1: Returns ***Prescription*** documents matching the query parameters:</u>

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- XDSDocumentEntry matches all other optional query parameters
- FormatCode matches **urn:ihe:pharm:pre:2010**
- Prescription document contains at least one Prescription Item ready to be dispensed
  - A Prescription Item is ready to dispense if there exists no Dispense Item to it.

<u>Business rule 2.2: Returns related ***Dispense*** documents to the Prescriptions found</u>

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches **urn:ihe:pharm:dis:2010**
- Dispense document contains a Dispense Entry Item related to a Prescription Item of the found Prescription documents.[24]

Explanation

Returning Prescription documents according to business rule 2.1 is the primary result of the query, where all optional query parameters which might affect the result of the query are applied.

The secondary result of the query, the related Dispense documents to the Prescriptions (Business rule 2.2), is dependent on the primary result (found Prescriptions) only and contains just directly related documents.

Query parameters:

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryPatientId	XDSDocumentEntry. patientId	R	--
$XDSDocumentEntryEntryUUID	XDSDocumentEntry. entryUUID	O[1]	M
$XDSDocumentEntryUniqueId	XDSDocumentEntry. uniqueId	O[1]	M
$XDSDocumentEntryPracticeSettingCode[3]	XDSDocumentEntry. practiceSettingCode	O	M

[24] See the Pharmacy Dispense Content Profile (DIS) for details how the relation between Dispense Entries and Prescription Items is defined, chapter: “Dispense Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.4)

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryCreationTimeFrom	Lower value of XDSDocumentEntry. creationTime	O	--
$XDSDocumentEntryCreationTimeTo	Upper value of XDSDocumentEntry. creationTime	O	--
$XDSDocumentEntryServiceStartTimeFrom	Lower value of XDSDocumentEntry. serviceStartTime	O	--
$XDSDocumentEntryServiceStartTimeTo	Upper value of XDSDocumentEntry. serviceStartTime	O	--
$XDSDocumentEntryServiceStopTimeFrom	Lower value of XDSDocumentEntry. serviceStopTime	O	--
$XDSDocumentEntryServiceStopTimeTo	Upper value of XDSDocumentEntry. serviceStopTime	O	--
$XDSDocumentEntryHealthcareFacilityTypeCode[3]	XDSDocumentEntry. healthcareFacilityTypeCode	O	M
$XDSDocumentEntryEventCodeList[3]	XDSDocumentEntry. eventCodeList[3]	O	M
$XDSDocumentEntryConfidentialityCode[3]	XDSDocumentEntry. confidentialityCode[3]	O	M
$XDSDocumentEntryAuthorPerson[4]	XDSDocumentEntry. Author	O	M
$XDSDocumentEntryStatus	XDSDocumentEntry. Status	R	M

[1]Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

[3]Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

1135 [4]The value for this parameter is a pattern compatible with the SQL keyword LIKE which allows the use of the following wildcard characters: % to match any (or no) characters and _ to match a single character. The match shall be applied to the text contained in the Value elements of the authorPerson Slot on the author Classification (value strings of the authorPerson sub-attribute)

1140 **Examples for the “FindPrescriptionsForValidation” query**

Assume the following situation of persisted documents in the Prescription-/Pharmaceutical Advice- and Dispense repositories:

Prescriptions		Pharmaceutical Advice	Dispenses	Remark
PRE 1	PRE Item 1-1	PADV 1	DIS 1	This item is already validated and dispensed
	PRE Item 1-2	PADV 2		This item is already validated and ready for dispense
PRE 2	PRE Item 2-1	PADV 3	DIS 2	This item is already validated and dispensed
	PRE Item 2-2	PADV 4		This item is already validated and ready for dispense
	PRE Item 2-3			This item is not validated yet
PRE 3	PRE Item 3-1			This item is not validated yet

<u>Example 1: Standard query</u>

Used Query Parameters:

- Patient ID
- Document Status

This is what should be returned by the query:

Returned XDSDocumentEntries		Explanation
Prescriptions	Related documents	
PRE 2	PADV 3, PADV 4, DIS 2	PRE Item 2-2 of PRE 2 is not validated yet and therefore PRE 2 shall be returned as result. PADV 3, PADV 4 and DIS 2 are all documents which are related to (some PRE Items on) PRE 2 and shall also be returned as result.
PRE 3		PRE Item 3-1 of PRE 3 is not validated yet and therefore PRE 3 shall be returned as result. No other related documents are available.

<u>Example 2: Search for a specific prescription</u>

Query Parameters set:

- Patient ID
- Document Status
- Document uniqueId of the specific prescription (e.g., because patient showed a paper prescription with the uniqueId printed on it)

In case the uniqueId of PRE 2 is given as query parameter, this is what should be returned by the query:

Returned XDSDocumentEntries		**Explanation**
Prescriptions	**Related documents**	
PRE 2	PADV 3, PADV 4, DIS 2	PRE Item 2-2 of PRE 2 is not validated yet and therefore PRE 2 shall be returned as result. PADV 3, PADV 4 and DIS 2 are all documents which are related to (some PRE Items on) PRE 2 and shall also be returned as result.

Note: In case the uniqueId of PRE 1 is given as query parameter, it would result in an empty result set, because PRE 1 would be the only possible return but contains no Prescription Item which is ready to be validated.

3.1.4.1.2.1.1.4 FindPrescriptionsForDispense

Find prescriptions and their related documents (XDSDocumentEntry objects) containing Prescription Items ***already validated and ready to be dispensed*** for a given patientID and other matching attributes. The other parameters can be used to restrict the set of XDSDocumentEntry objects returned.

Returns: XDSDocumentEntry objects according to the following business rules:

Business rule 1: Returns ***Prescription*** documents matching the query parameters:

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- XDSDocumentEntry matches all other optional query parameters
- FormatCode matches **urn:ihe:pharm:pre:2010**
- Prescription document contains at least one Prescription Item ready to dispense
  - A Prescription Item is ready to dispense if there exists a Pharmaceutical Advice Item related to it which has statusCode set to “completed”.[25]

Business rule 2: Returns related ***Pharmaceutical Advice*** documents to the Prescriptions found

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches **urn:ihe:pharm:padv:2010**

[25] See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details about the statusCode element

- Pharmaceutical Advice document contains a Pharmaceutical Advice Entry Item related to a Prescription Item of the found Prescription documents.[26]

Business rule 3: Returns related ***Dispense*** documents to the Prescriptions found

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches **urn:ihe:pharm:dis:2010**
- Dispense document contains a Dispense Entry Item related to a Prescription Item of the found Prescription documents.[27]

Explanation

Returning Prescription documents according to business rule 1 is the primary result of the query, where all optional query parameters which might affect the result of the query are applied.

The secondary result of the query, the related Pharmaceutical Advice and Dispense documents to the Prescriptions (Business rule 2 and 3), is dependent on the primary result (found Prescriptions) only and contains just directly related documents.

Query parameters:

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryPatientId	XDSDocumentEntry. patientId	R	--
$XDSDocumentEntryEntryUUID	XDSDocumentEntry. entryUUID	O[1]	M
$XDSDocumentEntryUniqueId	XDSDocumentEntry. uniqueId	O[1]	M
$XDSDocumentEntryPracticeSettingCode[3]	XDSDocumentEntry. practiceSettingCode	O	M
$XDSDocumentEntryCreationTimeFrom	Lower value of XDSDocumentEntry. creationTime	O	--
$XDSDocumentEntryCreationTimeTo	Upper value of XDSDocumentEntry. creationTime	O	--
$XDSDocumentEntryServiceStartTimeFrom	Lower value of XDSDocumentEntry. serviceStartTime	O	--

[26] See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details how the relation between Pharmaceutical Advice Entries and Prescription Items is defined, chapter: “Pharmaceutical Advice Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.3)

[27] See the Pharmacy Dispense Content Profile (DIS) for details how the relation between Dispense Entries and Prescription Items is defined, chapter: “Dispense Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.4)

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryServiceStartTimeTo	Upper value of XDSDocumentEntry. serviceStartTime	O	--
$XDSDocumentEntryServiceStopTimeFrom	Lower value of XDSDocumentEntry. serviceStopTime	O	--
$XDSDocumentEntryServiceStopTimeTo	Upper value of XDSDocumentEntry. serviceStopTime	O	--
$XDSDocumentEntryHealthcareFacilityTypeCode[3]	XDSDocumentEntry. healthcareFacilityTypeCode	O	M
$XDSDocumentEntryEventCodeList[3]	XDSDocumentEntry. eventCodeList[3]	O	M
$XDSDocumentEntryConfidentialityCode[3]	XDSDocumentEntry. confidentialityCode[3]	O	M
$XDSDocumentEntryAuthorPerson[4]	XDSDocumentEntry. Author	O	M
$XDSDocumentEntryStatus	XDSDocumentEntry. Status	R	M

[1] Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

[3] Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

1205 [4] The value for this parameter is a pattern compatible with the SQL keyword LIKE which allows the use of the following wildcard characters: % to match any (or no) characters and _ to match a single character. The match shall be applied to the text contained in the Value elements of the authorPerson Slot on the author Classification (value strings of the authorPerson sub-attribute)

1210 Examples for the “FindPrescriptionsForDispense” query

Assume the following situation of persisted documents in the Prescription-/Pharmaceutical Advice- and Dispense repositories:

Prescriptions		Pharmaceutical Advice	Dispenses	Remark
PRE 1	PRE Item 1-1	PADV 1	DIS 1	This item is already validated and dispensed
	PRE Item 1-2	PADV 2		This item is already validated and ready for dispense
PRE 2	PRE Item 2-1	PADV 3	DIS 2	This item is already validated and dispensed
	PRE Item 2-2	PADV 4		This item is already validated and ready for dispense

Prescriptions		Pharmaceutical Advice	Dispenses	Remark
	PRE Item 2-3			This item is not validated yet
PRE 3	PRE Item 3-1			This item is not validated yet

Example 1: Standard query

Used Query Parameters:

- Patient ID
- Document Status

This is what should be returned by the query:

Returned XDSDocumentEntries		Explanation
Prescriptions	Related documents	
PRE 1	PADV 1, PADV 2, DIS 1	PRE Item 1-2 of PRE 1 is validated but not dispensed yet, therefore PRE 1 shall be returned as result. PADV 1, PADV 2 and DIS 1 are all documents which are related to (some PRE Items on) PRE 1 and shall also be returned as result.
PRE 2	PADV 3, PADV 4, DIS 2	PRE Item 2-3 of PRE 2 is validated but not dispensed yet and therefore PRE 2 shall be returned as result. PADV 3, PADV 4 and DIS 2 are all documents which are related to (some PRE Items on) PRE 2 and shall also be returned as result.

Example 2: Search for a specific prescription

Query Parameters set:

- Patient ID
- Document Status
- Document uniqueId of the specific prescription (e.g., because patient showed a paper prescription with the uniqueId printed on it)

In case the uniqueId of PRE 1 is given as query parameter, this is what should be returned by the query:

Returned XDSDocumentEntries		Explanation
Prescriptions	Related documents	
PRE 1	PADV 1, PADV 2, DIS 1	PRE Item 1-2 of PRE 1 is validated but not dispensed yet, therefore PRE 1 shall be returned as result. PADV 1, PADV 2 and DIS 1 are all documents which are related to (some PRE Items on) PRE 1 and shall also be returned as result.

Note: In case the uniqueId of PRE 3 is given as query parameter, it would result in an empty result set, because PRE 3 would be the only possible return but contains no Prescription Item which is ready to be dispensed.

3.1.4.1.2.1.1.5 FindMedicationList

Find the Medication List On-Demand Document (XDSDocumentEntry object) according to the Pharmacy Medication List (PML) Profile containing Prescription- and Dispense Items for a given patientID and other matching attributes.

Returns: XDSDocumentEntry object according to the following business rules.

Business rule 1: Returns ***Medication List*** documents

This is the basic and ready to implement mechanism to retrieve a medication list. In this case, the Community Pharmacy Manager contains, or has access to, business rules to retrieve an existing list, or to create the patient's medication lists from the Pharmacy documents.

A set of parameters is provided in the query to obtain such list:

- XDSDocumentEntry matches all the required query parameters below (PatientID, …)
- FormatCode of the returned document match **urn:ihe:pharm:pml:2013** (implicit business rule, parameter shall not be given in parameter list)

Query parameters:

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryPatientId	XDSDocumentEntry.patientId	R	--
The following yellow parameters are parameters to parameterize the business logic for assembling of the resulting Medication List document:			
$XDSDocumentEntryServiceStartFrom	For the meaning of these parameters see explanation below.	O	--
$XDSDocumentEntryServiceStartTo		O	--
$XDSDocumentEntryServiceEndFrom		O	--

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryServiceEndTo		O	--
$XDSDocumentEntryFormatCode		O[28]	M
$XDSDocumentEntryType		O[29]	M
$XDSDocumentEntryStatus	XDSDocumentEntry.Status	R	M
$XDSDocumentEntryType[30]	XDSDocumentEntry.objectType	R	M

Explanation of yellow parameters

1) $XDSDocumentEntryServiceStartFrom, $XDSDocumentEntryServiceStartTo

This query parameter is used to find all medication treatments that were started during the interval specified by the requester. The exact definition of the starting point of a medication treatment is not in the scope of this profile.

For example:

(1) find and return all medication treatments that were started between Nov 2012 to June 2013

(2) find and return all medication treatments that were started in the last 3 months

2) $XDSDocumentEntryServiceEndFrom, $XDSDocumentEntryServiceEndTo,

This query parameter is used to find all medication treatments that were finished / completed in the interval specified by the requester. The exact definition of the point where a medication treatment is finished/completed is not in the scope of this profile.

For example:

(1) find and return all medication treatments that were completed between Nov 2012 to June 2013

(2) find and return all medication treatments that were completed in the last 3 months or are not yet completed

3) $XDSDocumentEntryFormatCode

[28] Note: Omitting this parameter means that no filtering according to format code takes place, so “all” available data types will be returned.

[29] Note: Omitting this parameter means that “all” available types of Medication List documents (on-demand created or previously persisted snapshots) are returned.

[30] See ITI TF-2a:3.18.4.1.2.3.6.2

If this parameter is given just the given type of information shall be returned in the Medication List.

Parameter	Meaning
urn:ihe:pharm:pre:2010	Prescription Items shall be returned (and optional the related Pharmaceutical Advice documents related to them).
urn:ihe:pharm:dis:2010	Dispense Items shall be returned (and optional the related Pharmaceutical Advice documents related to them).

4) $XDSDocumentEntryType

If this parameter is given documents of just the provided document entry type (on-demand or stable) shall be returned.

Parameter	Meaning
urn:uuid:34268e47-fdf5-41a6-ba33-82133c465248	On-Demand document entry types are returned. This is the on-demand created Medication List document.
urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1 See note [31]	Stable document entry types are returned. Previously persisted snapshots of on-demand created Medication List document.

3.1.4.1.2.2 Stored Query IDs

The Registry Stored Query (ITI-18) transaction defines several kinds of Stored Queries (FindDocuments, FindSubmissionSets, etc.).

The PHARM-1 transaction has to provide a different set of Stored Queries. They are assigned the following Query IDs. These IDs are used in the AdhocQueryRequest to reference queries stored on the Community Pharmacy Manager Actor. Query IDs are in UUID format (RFC4122). An error shall be returned when an unsupported stored query ID is received.

Query Name	Query ID
FindPrescriptions	urn:uuid:0e6095c5-dc3d-47d9-a219-047064086d92
FindDispenses	urn:uuid:ac79c7c7-f21b-4c88-ab81-57e4889e8758
FindPrescriptionsForValidation	urn:uuid:c1a43b20-0254-102e-8469-a6af440562e8
FindPrescriptionsForDispense	urn:uuid:c875eb9c-0254-102e-8469-a6af440562e8

[31] Note: This parameter is applicable only if the “Persistence of Retrieved Documents” Option is supported.

Query Name	Query ID
FindMedicationList	urn:uuid:80ebbd83-53c1-4453-9860-349585962af6

3.1.4.1.2.3 Web Services Transport

The Registry Stored Query (ITI-18) transaction defines the transmission using Web Services.

This chapter describes the differences of the PHARM-1 transaction to the ITI-18.

IHE-WSP201) The attribute /wsdl:definitions/@name shall be “CommunityPharmacyManager”.

The following WSDL naming conventions shall apply:

```
wsdl:definitions/@name="CommunityPharmacyManager":
query message    -> "QueryPharmacyDocuments_Message"
query response   -> "QueryPharmacyDocuments_Response_Message"
portType         -> "CommunityPharmacyManager_PortType"
operation        -> "QueryPharmacyDocuments"
SOAP 1.2 binding -> "CommunityPharmacyManager_Binding_Soap12"
SOAP 1.2 port    -> "CommunityPharmacyManager_Port_Soap12"
```

IHE-WSP202) The targetNamespace of the WSDL shall be “urn:ihe:iti:xds-b:2007”

These are the requirements for the Registry Stored Query transaction presented in the order in which they would appear in the WSDL definition:

- The following types shall be imported (xsd:import) in the /definitions/types section:
  - namespace=" urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0", schemaLocation="query.xsd"
- The /definitions/message/part/@element attribute of the Find Prescriptions Request message shall be defined as “query:AdhocQueryRequest”
- The /definitions/message/part/@element attribute of the Find Prescriptions Response message shall be defined as “query:AdhocQueryResponse”
- The /definitions/portType/operation/input/@wsaw:Action attribute for the Find Prescriptions Request message shall be defined as “urn:ihe:pharm:cmpd:2010:QueryPharmacyDocuments”
- The /definitions/portType/operation/output/@wsaw:Action attribute for the Find Prescriptions Response message shall be defined as “urn:ihe:pharm:cmpd:2010:QueryPharmacyDocumentsResponse”
- The /definitions/binding/operation/soap12:operation/@soapAction attribute should be defined as “urn:ihe:pharm:cmpd:2010:QueryPharmacyDocuments”

The following WSDL fragment shows an example of Find Prescription transaction definition:

```
<?xml version="1.0" encoding="utf-8"?>
<definitions ...>
  ...
  <types>
    <xsd:schema elementFormDefault="qualified" targetNamespace="urn:ihe:iti:xds-b:2007">
      <xsd:import
       namespace="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
       schemaLocation="schema\query.xsd"/>
      ...
    </xsd:schema>
  </types>
  <message name="QueryPharmacyDocuments_Message">
    <documentation>Query Pharmacy Documents</documentation>
    <part name="body" element="query:AdhocQueryRequest"/>
  </message>
  <message name="QueryPharmacyDocumentsResponse_Message">
    <documentation>Query Pharmacy Documents Response</documentation>
    <part name="body" element="query:AdhocQueryResponse"/>
  </message>
  ...
  <portType name="CommunityPharmacyManager_PortType">
    <operation name="CommunityPharmacyManager_QueryPharmacyDocuments">
      <input message="ihe:QueryPharmacyDocuments_Message"
         wsaw:Action="urn:ihe:pharm:cmpd:2010:QueryPharmacyDocuments"/>
      <output message="ihe:RegistryStoredQueryResponse_Message"
         wsaw:Action="urn:ihe:pharm:cmpd:2010:QueryPharmacyDocumentsResponse"/>
    </operation>
    ...
  </portType>
  ...
</definitions>
```

3.1.4.1.2.3.1 Sample SOAP Messages

The samples in the following two sections show a typical SOAP request and its relative SOAP response.

3.1.4.1.2.3.1.1 Sample Query Prescription SOAP Request

```
<s:Envelope xmlns:s=http://www.w3.org/2003/05/soap-envelope
            xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">urn:ihe:pharm:cmpd:2010:QueryPharmacyDocuments</a:Action>
    <a:MessageID>urn:uuid:def119ad-dc13-49c1-a3c7-e3742531f9b3</a:MessageID>
    <a:ReplyTo s:mustUnderstand="1">
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To>http://localhost/service/CommunityPharmacyManager.svc</a:To>
  </s:Header>
  <s:Body>
    <query:AdhocQueryRequest>
         :
       see Sample Query Request below
         :
    </query:AdhocQueryRequest>
  </s:Body>
```

```
</s:Envelope>
```

3.1.4.1.2.3.1.2 Sample Query Prescription SOAP Response

```
<s:Envelope xmlns:s=http://www.w3.org/2003/05/soap-envelope
            xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action
s:mustUnderstand="1">urn:ihe:pharm:cmpd:2010:QueryPharmacyDocumentsResponse</a:Action>
    <a:RelatesTo>urn:uuid:def119ad-dc13-49c1-a3c7-e3742531f9b3</a:RelatesTo>
  </s:Header>
  <s:Body>
    <query:AdhocQueryResponse>
         :
      see Sample Query Response below
         :
    </query:AdhocQueryResponse>
  </s:Body>
</s:Envelope>
```

3.1.4.1.3 Expected Actions

The Community Pharmacy Manager Actor shall do the same actions as described for the Document Registry Actor in the Registry Stored Query (ITI-18) transaction.

3.1.4.1.3.1 Sample Query Request

This example query specifies:

- The FindPrescriptionsForValidation query (id attribute of AdhocQuery element)
- patientID st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&ISO
- Return Approved documents only
- Time range (creation time) 200412252300 to 200501010800

Note that ebRS 3.0 specifies the use of Slot to specify name/value(s) pairs as parameters to a Stored Query.

Note: query parameter names are highlighted for readability.

```
<query:AdhocQueryRequest
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
    xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
    xmlns:rs="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0">
  <query:ResponseOption returnComposedObjects="true" returnType="LeafClass"/>
  <rim:AdhocQuery id="urn:uuid:c1a43b20-0254-102e-8469-a6af440562e8">
    <rim:Slot name="$XDSDocumentEntryPatientId">
      <rim:ValueList>
        <rim:Value>'st3498702^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp;ISO'</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="$XDSDocumentEntryStatus">
      <rim:ValueList>
        <rim:Value>('urn:oasis:names:tc:ebxml-regrep:StatusType:Approved')</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="$XDSDocumentEntryCreationTimeFrom">
      <rim:ValueList>
        <rim:Value>200412252300</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="$XDSDocumentEntryCreationTimeTo">
      <rim:ValueList>
        <rim:Value>200501010800</rim:Value>
      </rim:ValueList>
    </rim:Slot>
  </rim:AdhocQuery>
</query:AdhocQueryRequest>
```

3.1.4.1.3.2 Sample Query Response

This sample query response corresponds to the above query. Note that the query response message is coded in version 3.0 ebRIM and ebRS. This sample response and the ebXML Registry version 3.0 schema files are available online. The Implementation Guide found at http://wiki.ihe.net/index.php?title=ITI_Implementation_Guide contains such supplemental material.

```
<?xml version="1.0" encoding="UTF-8"?>
<AdhocQueryResponse
xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0 file:/query.xsd"
status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success">
  <rim:RegistryObjectList>
    <rim:ExtrinsicObject
xmlns:q="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
id="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
isOpaque="false"
mimeType="text/xml"
objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"
status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved">
      <rim:Slot name="URI">
        <rim:ValueList>
          <rim:Value>http://localhost:8080/XDS/Repository/08a15a6f-5b4a-42de-8f95-
89474f83abdf.xml</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="authorInstitution">
        <rim:ValueList>
          <rim:Value>Some Hospital^^^^^^^^^1.2.3.4.5.6.7.8.9.1789.45</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="creationTime">
        <rim:ValueList>
          <rim:Value>200412261119</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="hash">
        <rim:ValueList>
          <rim:Value>4cf4f82d78b5e2aac35c31bca8cb79fe6bd6a41e</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="languageCode">
        <rim:ValueList>
          <rim:Value>en-us</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="serviceStartTime">
        <rim:ValueList>
          <rim:Value>200412230800</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="serviceStopTime">
        <rim:ValueList>
          <rim:Value>200412230801</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="size">
        <rim:ValueList>
          <rim:Value>54449</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="sourcePatientId">
        <rim:ValueList>
          <rim:Value>jd12323^^^wsh</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="sourcePatientInfo">
        <rim:ValueList>
          <rim:Value>PID-3|pid1^^^domain</rim:Value>
          <rim:Value>PID-5|Doe^John^^^</rim:Value>
          <rim:Value>PID-7|19560527</rim:Value>
          <rim:Value>PID-8|M</rim:Value>
          <rim:Value>PID-11|100 Main St^^Metropolis^Il^44130^USA</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="Pharmacy Prescription" xml:lang="en-
us"/>
      </rim:Name>
      <rim:Description/>
      <rim:Classification
classificationScheme="urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a"
classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
id="urn:uuid:ac872fc0-1c6e-439f-84d1-f76770a0ccdf"
nodeRepresentation="57833-6"
objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
        <rim:Slot name="codingScheme">
          <rim:ValueList>
            <rim:Value>Connect-a-thon classCodes</rim:Value>
          </rim:ValueList>
        </rim:Slot>
        <rim:Name>
          <rim:LocalizedString charset="UTF-8" value="Prescriptions" xml:lang="en-us"/>
        </rim:Name>
        <rim:Description/>
      </rim:Classification>
      <rim:Classification
classificationScheme="urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f"
classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
id="urn:uuid:f1a8c8e4-3593-4777-b7e0-8b0773378705"
nodeRepresentation="N"
objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
        <rim:Slot name="codingScheme">
          <rim:ValueList>
            <rim:Value>Connect-a-thon confidentialityCodes</rim:Value>
          </rim:ValueList>
        </rim:Slot>
        <rim:Name>
          <rim:LocalizedString charset="UTF-8" value="Normal" xml:lang="en-us"/>
        </rim:Name>
        <rim:Description/>
      </rim:Classification>
      <rim:Classification
classificationScheme="urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d"
classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
id="urn:uuid:b6e49c73-96c8-4058-8c95-914d83bd262a"
nodeRepresentation="urn:ihe:pharm:pre:2010"
objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
        <rim:Slot name="codingScheme">
          <rim:ValueList>
            <rim:Value>Connect-a-thon formatCodes</rim:Value>
          </rim:ValueList>
        </rim:Slot>
        <rim:Name>
          <rim:LocalizedString charset="UTF-8" value="Pharmacy Prescription"
xml:lang="en-us"/>
        </rim:Name>
        <rim:Description/>
      </rim:Classification>
      <rim:Classification
classificationScheme="urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1"
classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
id="urn:uuid:61e2b376-d74a-4984-ac21-dcd0b8890f9d"
nodeRepresentation="Emergency Department"
objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
        <rim:Slot name="codingScheme">
          <rim:ValueList>
            <rim:Value>Connect-a-thon healthcareFacilityTypeCodes</rim:Value>
          </rim:ValueList>
        </rim:Slot>
        <rim:Name>
          <rim:LocalizedString charset="UTF-8" value="Emergency Department"
xml:lang="en-us"/>
        </rim:Name>
        <rim:Description/>
      </rim:Classification>
      <rim:Classification
classificationScheme="urn:uuid:cccf5598-8b07-4b77-a05e-ae952c785ead"
```

```
classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
id="urn:uuid:fb7677c5-c42f-485d-9010-dce0f3cd4ad5"
nodeRepresentation="Cardiology"
objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
      <rim:Slot name="codingScheme">
        <rim:ValueList>
          <rim:Value>Connect-a-thon practiceSettingCodes</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="Cardiology" xml:lang="en-us"/>
      </rim:Name>
      <rim:Description/>
     </rim:Classification>
     <rim:Classification
classificationScheme="urn:uuid:f0306f51-975f-434e-a61c-c59651d33983"
classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
id="urn:uuid:0a8a8ed9-8be5-4a63-9b68-a511adee8ed5"
nodeRepresentation="57833-6"
objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
      <rim:Slot name="codingScheme">
        <rim:ValueList>
          <rim:Value>LOINC</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="Prescriptions" xml:lang="en-us"/>
      </rim:Name>
      <rim:Description/>
     </rim:Classification>
     <rim:ExternalIdentifier
id="urn:uuid:db9f4438-ffff-435f-9d34-d76190728637"
registryObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
identificationScheme="urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427"
objectType="ExternalIdentifier"
value="st3498702^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp;ISO">
      <rim:Name>
        <rim:LocalizedString charset="UTF-8"
value="XDSDocumentEntry.patientId" xml:lang="en-us"/>
      </rim:Name>
      <rim:Description/>
     </rim:ExternalIdentifier>
     <rim:ExternalIdentifier
id="urn:uuid:c3fcbf0e-9765-4f5b-abaa-b37ac8ff05a5"
registryObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
identificationScheme="urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab"
objectType="ExternalIdentifier" value="1.3.6.1.4.1.21367.2005.3.99.1.1010">
      <rim:Name>
        <rim:LocalizedString charset="UTF-8"
value="XDSDocumentEntry.uniqueId" xml:lang="en-us"/>
      </rim:Name>
      <rim:Description/>
     </rim:ExternalIdentifier>
   </rim:ExtrinsicObject>
  </rim:RegistryObjectList>
</AdhocQueryResponse>
```

3.1.5 Security Considerations

Relevant XDS Affinity Domain Security background is discussed in the XDS Security
1650 Considerations Section (see ITI TF-1: 10.7).

3.1.5.1 Security Audit Considerations

The Actors involved shall record audit events according to the following:

3.1.5.1.1 Querying actor audit message:

	Field Name	Opt	Value Constraints
Event AuditMessage/ EventIdentification	EventID	M	EV(110112, DCM, "Query")
	EventActionCode	M	"E" (Execute)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("**PHARM-1**", "IHE Transactions", "**Query Pharmacy Documents**")
Source (Document Consumer) (1)			
Human Requestor (0..n)			
Destination (Document Registry) (1)			
Audit Source (Document Consumer) (1)			
Patient (0..1)			
Query Parameters(1)			

Where:

Source AuditMessage/ ActiveParticipant	UserID	M	The content of the <wsa:ReplyTo/> element.
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.
Human Requestor (if known) AuditMessage/ ActiveParticipant	UserID	M	Identity of the human that initiated the transaction.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

Destination AuditMessage/ ActiveParticipant	UserID	M	SOAP endpoint URI.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	“false”
	RoleIDCode	M	EV(110152, DCM, “Destination”)
	NetworkAccessPointTypeCode	M	“1” for machine (DNS) name, “2” for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source AuditMessage/ AuditSourceIdentification	*AuditSourceID*	*U*	*not specialized*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient (AuditMessage/ ParticipantObjectIdentification)	ParticipantObjectTypeCode	M	“1” (Person)
	ParticipantObjectTypeCodeRole	M	“1” (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, “Patient Number”)
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
Query Parameters (AuditMessage/ ParticipantObjectIdentification)	ParticipantObjectTypeCode	M	“2” (system object)
	ParticipantObjectTypeCodeRole	M	“24” (query)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(“**PHARM-1**”, “IHE Transactions”, “**Query Pharmacy Documents**”)
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	Stored Query ID (UUID)
	ParticipantObjectName	C	If known the value of <ihe:HomeCommunityId/>
	ParticipantObjectQuery	M	the AdhocQueryRequest, base64 encoded.

	ParticipantObjectDetail	C	The ParticipantObjectDetail element may occur more than once. In one element, set "QueryEncoding" as the value of the attribute type, Set the attribute value to the character encoding, such as "UTF-8", used to encode the ParticipantObjectQuery before base64 encoding. In another element, set "urn:ihe:iti:xca:2010:homeCommunityId" as the value of the attribute type and the value of the homeCommunityID as the value of the attribute value, if known.

3.1.5.1.2 Community Pharmacy Manager audit message:

	Field Name	Opt	Value Constraints
Event AuditMessage/ EventIdentification	EventID	M	EV(110112, DCM, "Query")
	EventActionCode	M	"E" (Execute)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("**PHARM-1**", "IHE Transactions", "**Query Pharmacy Documents**")
Source (Document Consumer) (1)			
Destination (Document Registry) (1)			
Audit Source (Document Registry) (1)			
Patient (0..1)			
Query Parameters(1)			

1660 Where:

Source AuditMessage/ ActiveParticipant	UserID	M	The content of the <wsa:ReplyTo/> element.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Destination AuditMessage/ ActiveParticipant	UserID	M	SOAP endpoint URI.
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address

	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source AuditMessage/ AuditSourceIdentification	*AuditSourceID*	*U*	*not specialized*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient (AuditMessage/ ParticipantObjectIdentification)	ParticipantObjectTypeCode	M	“1” (Person)
	ParticipantObjectTypeCodeRole	M	“1” (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, “Patient Number”)
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
Query Parameters (AuditMessage/ ParticipantObjectIdentification)	ParticipantObjectTypeCode	M	“2” (system object)
	ParticipantObjectTypeCodeRole	M	“24” (query)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(“PHARM-1”, “IHE Transactions”, “Query Pharmacy Documents”)
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	Stored Query ID (UUID)
	ParticipantObjectName	C	If known the value of <ihe:HomeCommunityId/>
	ParticipantObjectQuery	M	the AdhocQueryRequest, base64 encoded.
	ParticipantObjectDetail	C	The ParticipantObjectDetail element may occur more than once. In one element, set “QueryEncoding” as the value of the attribute type, Set the attribute value to the character encoding, such as “UTF-8”, used to encode the ParticipantObjectQuery before base64 encoding. In another element, set “urn:ihe:iti:xca:2010:homeCommunityId” as the value of the attribute type and the value of the homeCommunityID as the value of the attribute value, if known.

3.1.5.1.(z) Actor Specific Security Considerations

No information available yet.

4 Workflow Definitions

The management of the workflow related to clinical process has becoming a fundamental topic with the increasing of the use by different sectors of document sharing related IHE profiles with their different types of document and information.

4.1 Community Medication Prescription and Dispense Workflow Definition (CMPD-WD)

The management of the workflow related to the CMPD Profile is involved in much clinical and organizational process for its important role in the process of digitalization. The lack of a workflow management blocks the use of the Prescription in an extended way. The definition of a workflow with defined rules and tasks is needed in a scenario cross enterprise in which many actors are involved in the same process.

In this chapter a set of rules which defines the workflow of the CMPD process and the relationship with the actors involved are described. If real-world scenarios need a technical workflow management the actors involved in the process can use the “Workflow Management” Option which groups the CMPD actors with the XDW actors.

The ITI XDW Profile 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 by the different domains. For the definition of the CMPD workflow it is possible to use the ITI XDW Profile as an infrastructure layer to define a set logical or clinical tasks definitions and rules to apply. The rules in the workflow definition ensure that the different participants in a workflow operate jointly to advance within tasks and to move from one task to another in a consistent way.

To integrate the CMPD Profile with ITI XDW Profile it is necessary to introduce the integrations described in the follow paragraphs.

4.1.1 Actors and Grouping

If the “Workflow Management” Option is supported the following CMPD actors shall be grouped with XDW actors to allow access and manipulation of the XDW-WD (XDW Workflow document).

Actor	Groups with	Note
Prescription Placer	XDW: Content Creator XDW: Content Consumer XDW: Content Updater	The Prescription Placer Actor shall create the XDW-WD to start the process. It also consumes and maybe updates the XDW-WD document in case of modification to the Prescription.
Pharmaceutical Adviser	XDW: Content Consumer XDW: Content Updater	The Pharmaceutical Adviser Actor consumes and updates the XDW-WD after validation of a Prescription Item.

Actor	Groups with	Note
Medication Dispenser	XDW: Content Consumer XDW: Content Updater	The Medication Dispenser Actor consumes and updates the XDW-WD after dispensing a Prescription Item.

4.1.2 XDW Workflow Document – Common Attributes

The CMPD Workflow Definition does not introduce new metadata and all the metadata elements used are the common XDS document metadata specified in ITI TF-3:4.1.5 and in ITI TF-3:5.4.6. 1700 In this section only the use of some specific metadata for the use of XDW in the CMPD context is specified.

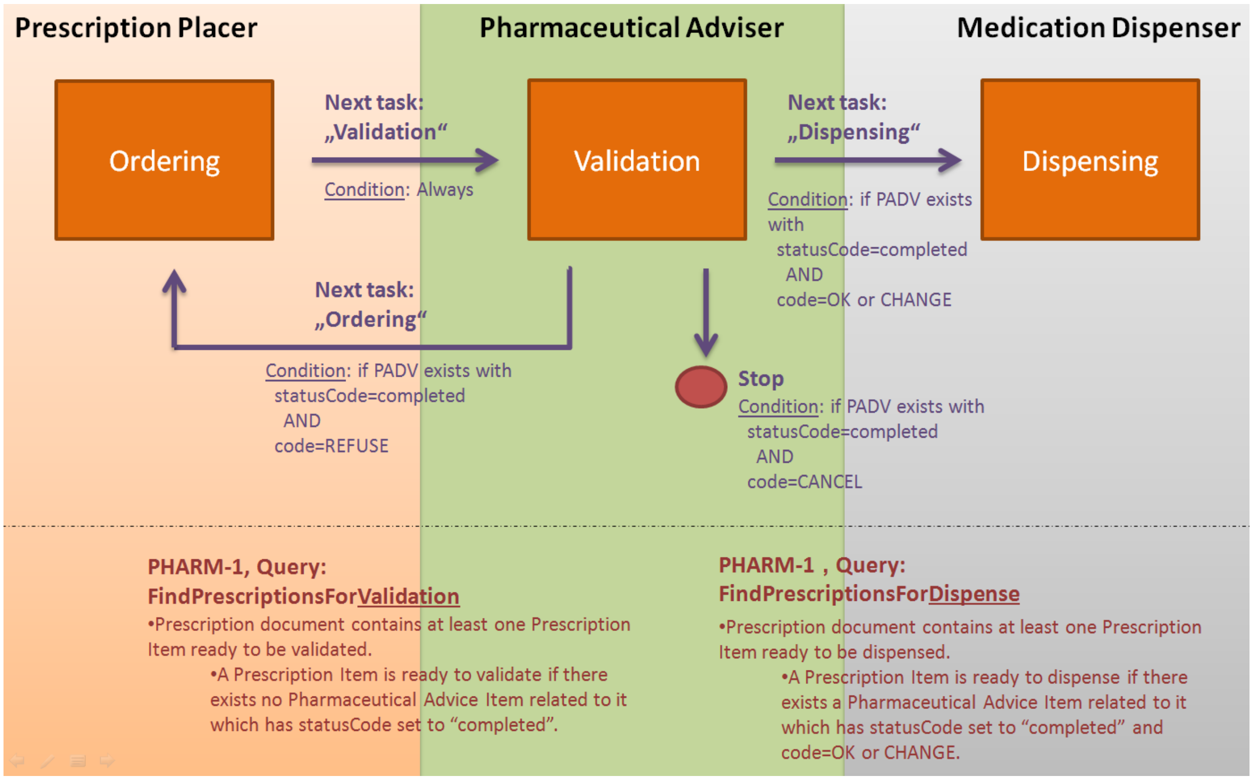
XDSDocumentEntry Attribute	Definition
typeCode	For the Workflow Document which tracks the CMPD process the code for the typeCode shall be: **Scenario 1: urn:ihe:pharm:cmpdwd1:2011** **Scenario 2: urn:ihe:pharm:cmpdwd2:2011** Note: see chapter 4.1.3 for a description of the two workflow scenarios. This code is the same code that shall be used in the element **workflowDefinitionReference** inside the Workflow Document
eventCodeList	Rule 1: A CMPD workflow shall be created with code OPEN and shall remain in this status until it is set to CLOSE. Rule 2: A CMPD workflow should be set to CLOSE by any actor which discovers that the partial workflow of each Prescription Item within (for which a task “Ordering” has been created) has ended either by a complete dispense of the item or any other way (e.g., a cancelation, etc.). See ITI TF-3: 5.4.5.7 for a general description of this attribute.
serviceStartTime	It is the time at which work began on the earliest task for this workflow.
serviceStopTime	It is the time at which the status of the overall Workflow is changed from OPEN to CLOSE. It shall be empty when the workflow is still in OPEN state.

4.1.3 Workflow Task Definition

1705 This chapter describes Workflow Tasks which are used in the XDW Workflow document to express a Community Pharmacy workflow.

The Community Pharmacy workflow can be divided in two different scenarios:

Scenario 1: Including validation step



Prescription Placer

Pharmaceutical Adviser

Medication Dispenser

Ordering

Validation

Dispensing

Next task: „Validation"

Condition: Always

Next task: „Dispensing"

Condition: if PADV exists with statusCode=completed AND code=OK or CHANGE

Next task: „Ordering"

Condition: if PADV exists with statusCode=completed AND code=REFUSE

Stop

Condition: if PADV exists with statusCode=completed AND code=CANCEL

PHARM-1, Query: FindPrescriptionsForValidation

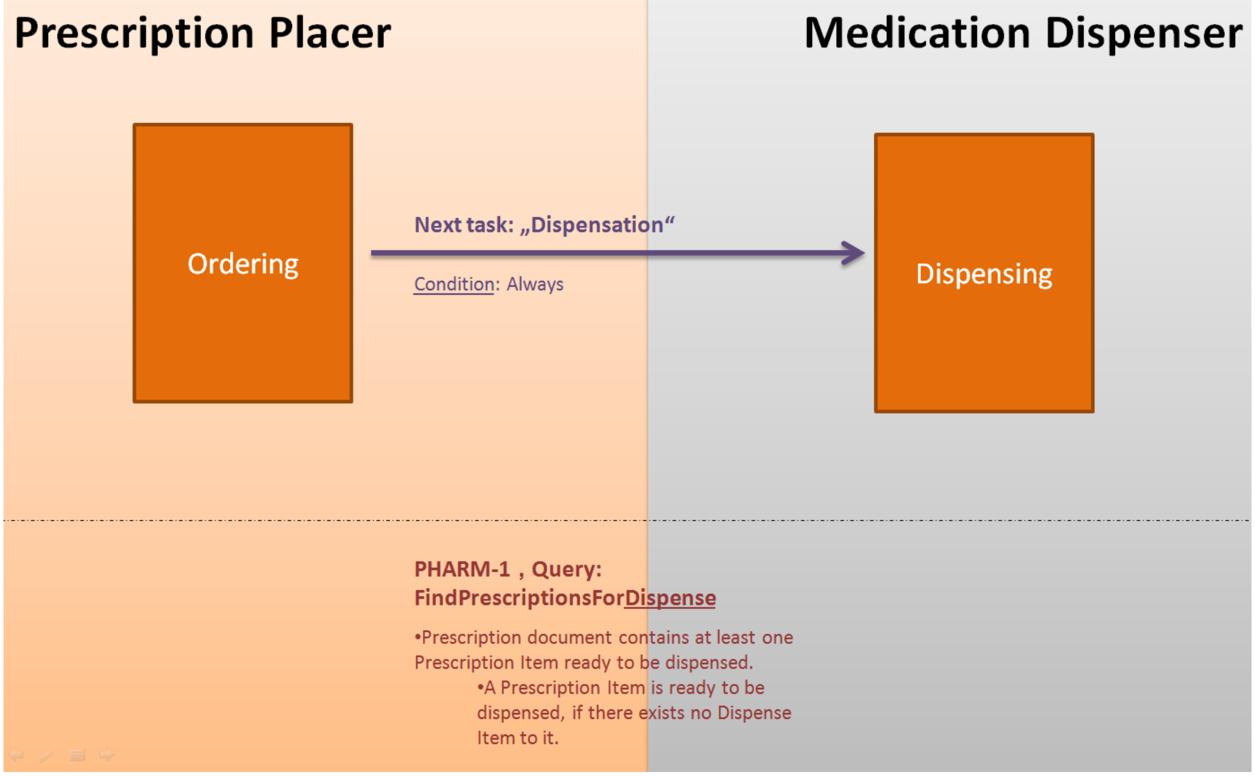
- Prescription document contains at least one Prescription Item ready to be validated.
  - A Prescription Item is ready to validate if there exists no Pharmaceutical Advice Item related to it which has statusCode set to "completed".

PHARM-1 , Query: FindPrescriptionsForDispense

- Prescription document contains at least one Prescription Item ready to be dispensed.
  - A Prescription Item is ready to dispense if there exists a Pharmaceutical Advice Item related to it which has statusCode set to "completed" and code=OK or CHANGE.

Figure 4.1.3-1: Scenario 1: Overall context of the workflow

Scenario 2: Not including validation step



PHARM-1 , Query: FindPrescriptionsForDispense

•Prescription document contains at least one Prescription Item ready to be dispensed.

- •A Prescription Item is ready to be dispensed, if there exists no Dispense Item to it.

Figure 4.1.3-2: Scenario 2: Overall context of the workflow

Please refer to Volume 1, chapter 4.4 CMPD Process Flow for a detailed explanation of the Community Pharmacy workflow scenarios.

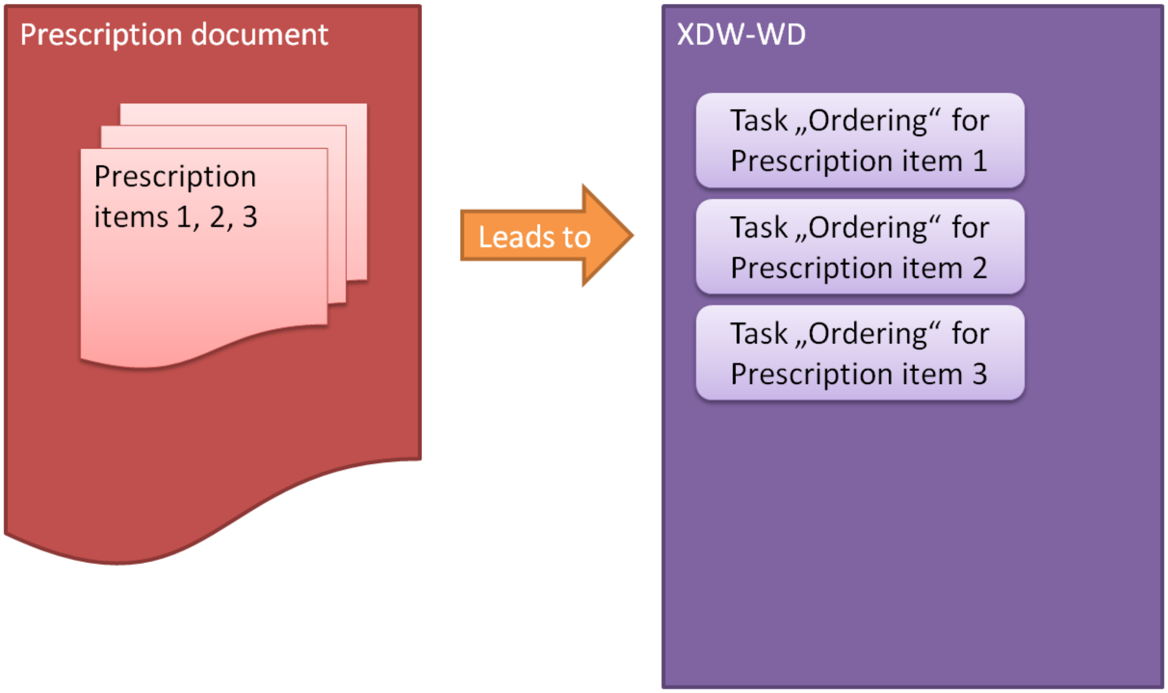
The following workflow tasks are defined and specified in detail in the following chapters:

- Ordering
- Validation
- Dispensing

Important Note: When referencing input and output documents in tasks, both document uniqueId and homeCommunityId of the document shall be used.

4.1.3.1 Task: Ordering

The task "Ordering" starts the Community Pharmacy workflow by creation of the Workflow document together with a single Prescription document. The task is able to record the creation of 1730 a single Prescription Item (within a Prescription), hence separate tasks have to be created for each Prescription Item of the Prescription.


Prescription document
Prescription items 1, 2, 3
Leads to
XDW-WD
Task „Ordering" for Prescription item 1
Task „Ordering" for Prescription item 2
Task „Ordering" for Prescription item 3

1735 **Figure 4.1.3.1-1: Prescription containing Prescription Items leading to workflow tasks "Ordering"**

Table 4.1.3.1-1: Ordering Task Rules

Task attributes	Rules for the task "Ordering"
Task id	Unique id of the instance of the task
Task type	**Ordering**
Task name	Order_of_Prescription_Item
Task description	The description element shall contain the PrescriptionItemId, this task is referring to (substanceAdministration/id element of the Prescription Item).

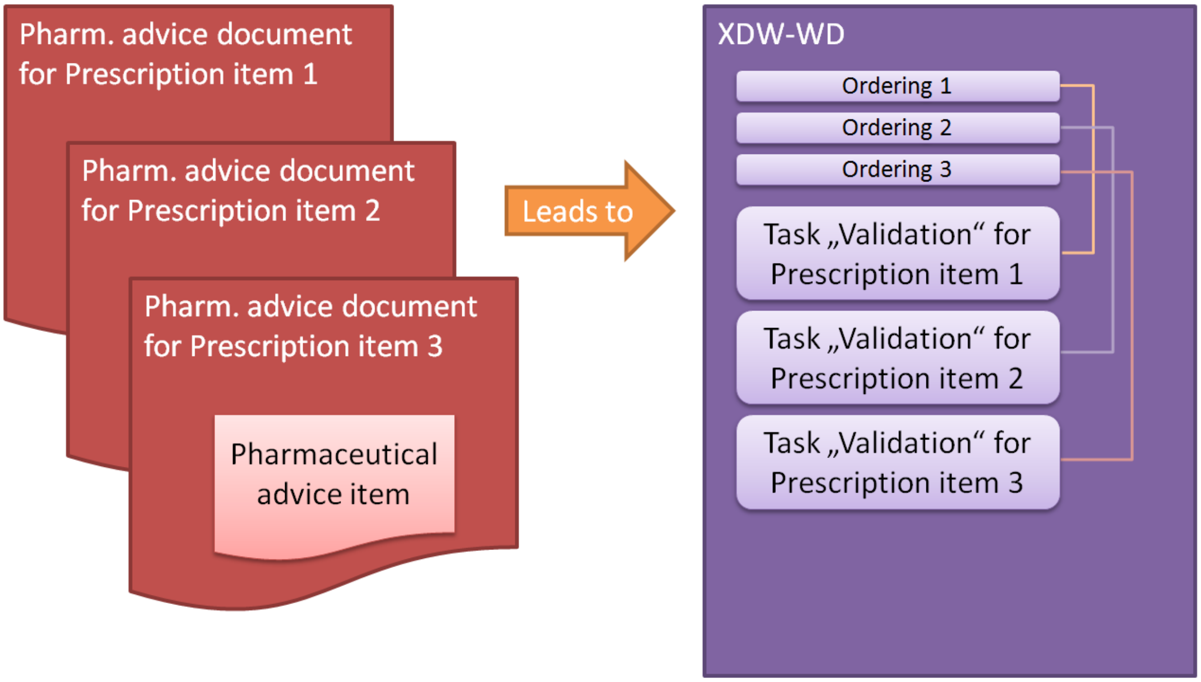
Task attributes	Rules for the task “Ordering”
	Format compliant to the HL7 v2 CX datatype: *Variant 1: Only id/@root is given* $desc = substanceAdministration/id/@root *Variant 2: id/@root and id/@extension is given* $desc = concat(substanceAdministration/id/@extension, "^^^&", substanceAdministration/id/@root, "&ISO")
Task dependencies	**Workflow scenario 1** Ancestors: None, Validation Successors: Validation **Workflow scenario 2** Ancestors: None Successors: Dispensing
Status allowed	**COMPLETED** An Ordering task is always set to COMPLETED.
Status transactions	None
input	• Optional o All documents useful to understand the reason for the prescription (clinical reports, …) may be referenced.
output	• Required o The Prescription document produced shall be referenced.
owner	Same Physician or organization that creates the Prescription document
owner changes	No
<taskEvent>	Only one

Example XML for this XDW task:

```
  :
<ns3:taskData>


  <ns2:taskDetails>
    <ns2:id>urn:oid:1.1.1.1.1</ns2:id>
    <ns2:taskType>Ordering</ns2:taskType>
    <ns2:name>Order_of_Prescription_Item</ns2:name>
    <ns2:status>COMPLETED</ns2:status>
    <ns2:actualOwner>Dr. Brum</ns2:actualOwner>
    <ns2:createdTime>2006-05-04T18:13:51.0Z</ns2:createdTime>
    <ns2:createdBy>Dr. Brum</ns2:createdBy>
    <ns2:lastModifiedTime>2006-05-04T18:13:51.0Z</ns2:lastModifiedTime>
    <ns2:renderingMethodExists>false</ns2:renderingMethodExists>
  </ns2:taskDetails>

  <!--
    The description element shall contain the PrescriptionItemId, this task is
    referring to (substanceAdministration/id element of the Prescription Item)
  -->
  <ns2:description>4711^^^&amp;1.2.3.4.5.6.7.8.9&amp;ISO</ns2:description>

  <!-- input documents -->
  <ns2:input>
    <ns2:part name="document">
      <!-- Document useful to understand the reason of the prescription -->
      <!-- uid: the document uniqueId, home: the homeCommunityId -->
      <reference uid="urn:oid:1.2.3.4.4.3.2.2.3" home="urn:oid:1.2.3"/>
    </ns2:part>
  </ns2:input>

  <!-- output documents -->
  <ns2:output>
    <ns2:part name="Prescription_Document">
      <!-- Prescription document according to PRE Profile -->
      <!-- uid: the document uniqueId, home: the homeCommunityId -->
      <reference uid="urn:oid:1.2.3.4.4.4" home="urn:oid:1.2.3"/>
    </ns2:part>
  </ns2:output>

</ns3:taskData>
  :
```

1785 4.1.3.2 Task: Validation

The task "Validation" is able to record the validation of a single Prescription Item (within a Prescription). Each validation results in the creation of a Pharmaceutical Advice document which documents the outcome of the validation. Hence separate tasks have to be created for each Prescription Item of the Prescription which is validated.


Pharm. advice document for Prescription item 1
Pharm. advice document for Prescription item 2
Pharm. advice document for Prescription item 3
Pharmaceutical advice item
Leads to
XDW-WD
Ordering 1
Ordering 2
Ordering 3
Task „Validation" for Prescription item 1
Task „Validation" for Prescription item 2
Task „Validation" for Prescription item 3

1790

Figure 4.1.3.2-1: Prescription Items leading to workflow tasks "Validation"

Table 4.1.3.2-1: Validation Task Rules

Task attributes	Rules for the task "Validation"
Task id	Unique id of the instance of the task
Task type	**Validation**
Task name	Validation_of_Prescription_Item
Task description	The description element shall contain the PrescriptionItemId, this task is referring to (substanceAdministration/id element of the Prescription Item). Format compliant to the HL7 v2 CX datatype: *Variant 1: Only id/@root is given*

Task attributes	Rules for the task “Validation”
	$desc = substanceAdministration/id/@root *Variant 2: id/@root and id/@extension is given* $desc = concat(substanceAdministration/id/@extension, "^^^&", substanceAdministration/id/@root, "&ISO")
Task dependencies	**Workflow scenario 1** Ancestors: Ordering Successors: Dispensing, Ordering **Workflow scenario 2** Not applicable
Status allowed	**IN PROGRESS** A Validation task shall be set to IN PROGRESS, when the outcome of the task is a ***preliminary*** validation result. A Pharmaceutical Advice document is preliminary, if the element statusCode is set to ***active***. **COMPLETED** A Validation task shall be set to COMPLETED, when the outcome of the task is a ***final*** validation result. A Pharmaceutical Advice document is final, if the element statusCode is set to ***completed***. Note: See Pharmacy Pharmaceutical Advice (PADV) Profile, Vol. 2, chapter “6.3.4.3.3.6 Status Code”.
Status transactions	All the states of the task (in progress, completed) are performed in sequence
input	• Required o The Prescription document containing the validated Prescription Item shall be referenced. The specific Prescription Item the task is referring to shall also be stated in that structure. o The ancestor task shall be referenced. • Optional o All additional documents useful to understand the reason for the outcome of the Pharmaceutical Advice may be referenced.
output	• Required o The Pharmaceutical Advice document produced shall be referenced.
owner	Same Physician or organization that creates the Pharmaceutical Advice document
changes of task owner	Yes

Task attributes	Rules for the task “Validation”
	The owner may change, if more than one Pharmaceutical Advice documents are created (in case of a multi-step validation using preliminary validation).
<taskEvent>	At least one

Example XML for this XDW task:

```
 :
<ns3:taskData>

 <ns2:taskDetails>
  <ns2:id>urn:oid:2.2.2.2.2</ns2:id>
  <ns2:taskType>Validation</ns2:taskType>
  <ns2:name>Validation_of_Prescription_Item</ns2:name>
  <ns2:status>COMPLETED</ns2:status>
  <ns2:actualOwner>Dr. Brum</ns2:actualOwner>
  <ns2:createdTime>2006-05-04T18:13:51.0Z</ns2:createdTime>
  <ns2:createdBy>Dr. Brum</ns2:createdBy>
  <ns2:lastModifiedTime>2006-05-04T18:13:51.0Z</ns2:lastModifiedTime>
  <ns2:renderingMethodExists>false</ns2:renderingMethodExists>
 </ns2:taskDetails>

 <!--
  The description element shall contain the PrescriptionItemId, this task is
  referring to (substanceAdministration/id element of the Prescription Item)
 -->
 <ns2:description>4711^^^&amp;1.2.3.4.5.6.7.8.9&amp;ISO</ns2:description>

 <!-- input documents -->
 <ns2:input>
  <ns2:part name="Prescription_Document">
   <!-- Prescription document according to PRE Profile -->
   <!-- uid: the document uniqueId, home: the homeCommunityId -->
   <reference uid="urn:oid:1.2.3.4.4.4" home="urn:oid:1.2.3"/>
  </ns2:part>
  <ns2:part name="Ancestor_task">
   <!-- Ancestor task -->
   <reference taskId="urn:oid:1.1.1.1.1"/>
  </ns2:part>
 </ns2:input>

 <!-- output documents -->
 <ns2:output>
  <ns2:part name="Pharmaceutical_Advice_Document">
   <!-- Pharmaceutical Advice document according to PADV Profile -->
   <!-- uid: the document uniqueId, home: the homeCommunityId -->
   <reference uid="urn:oid:1.2.3.4.4.5" home="urn:oid:1.2.3"/>
```

```
    </ns2:part>
  </ns2:output>

</ns3:taskData>
  :
```

4.1.3.3 Task: Dispensing

The task "Dispensing" is able to record the dispense of a single Prescription Item (within a Prescription). Each dispense results in the creation of a Medication Dispense document. Hence separate tasks have to be created for each Prescription Item of the Prescription which is
1850 dispensed.

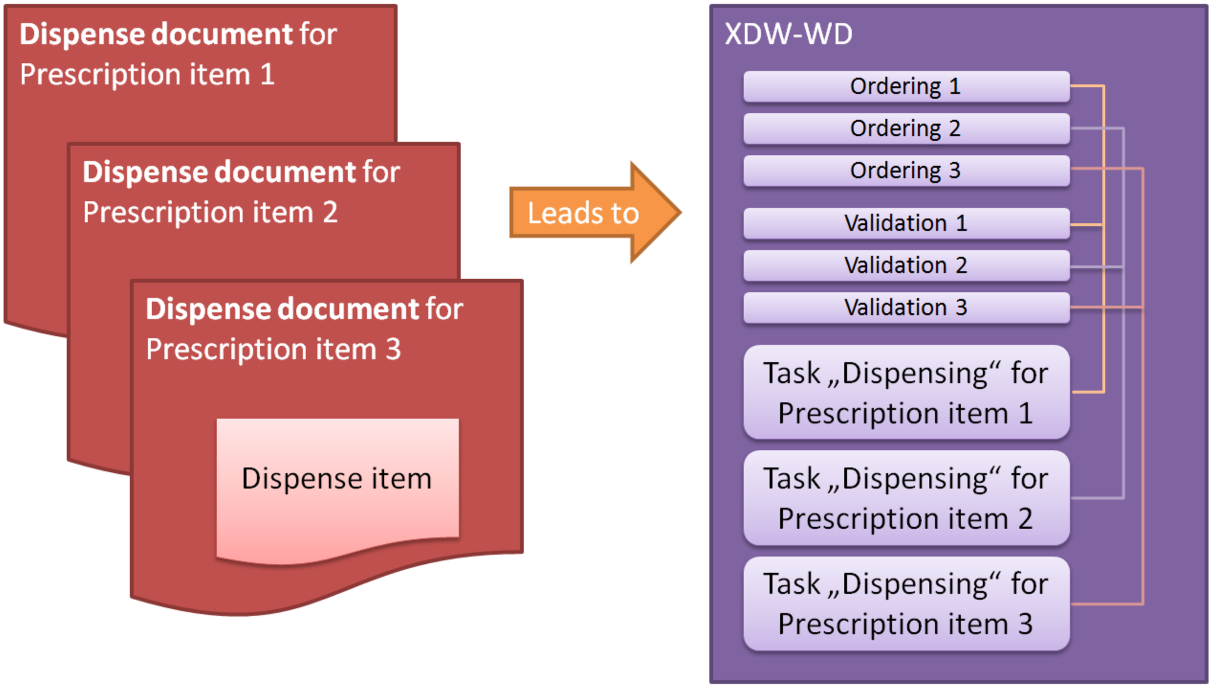

Dispense document for Prescription item 1
Dispense document for Prescription item 2
Dispense document for Prescription item 3
Dispense item
Leads to
XDW-WD
Ordering 1
Ordering 2
Ordering 3
Validation 1
Validation 2
Validation 3
Task „Dispensing" for Prescription item 1
Task „Dispensing" for Prescription item 2
Task „Dispensing" for Prescription item 3

Figure 4.1.3.3-1: Prescription Items leading to workflow tasks "Dispensing"

1855 **Table 4.1.3.3-1: Dispensing Task Rules**

Task attributes	Rules for the task "Dispensing"
Task id	Unique id of the instance of the task
Task type	**Dispensing**
Task name	Dispense_of_Prescription_Item
Task description	The description element shall contain the PrescriptionItemId, this task is referring to (substanceAdministration/id element of the Prescription Item). Format compliant to the HL7 v2 CX datatype:

Task attributes	Rules for the task “Dispensing”
	Variant 1: Only id/@root is given $desc = substanceAdministration/id/@root *Variant 2: id/@root and id/@extension is given* $desc = concat(substanceAdministration/id/@extension, "^^^&", substanceAdministration/id/@root, "&ISO")
Task dependencies	**Workflow scenario 1** Ancestors: Validation Successors: None **Workflow scenario 2** Ancestors: Ordering Successors: None
Status allowed	**IN PROGRESS** A Dispensing task shall be set to IN PROGRESS, when the order is repeatable and the current dispense is ***not the last*** repeat. **COMPLETED** A Dispensing task shall be set to IN PROGRESS, when the order is repeatable and the current dispense is the ***last*** repeat. Note: See Pharmacy Prescription (PRE) Profile, Vol. 2, chapter “6.3.4.1.3.21 Amount of units of the consumable to dispense”.
Status transactions	All the states of the task (in progress, completed) are performed in sequence
input	• Required o The Prescription document containing the dispensed Prescription Item shall be referenced. The specific Prescription Item the task is referring to shall also be stated in that structure. o *Additional in scenario 2:* The Pharmaceutical Advice document related to the Prescription Item which approved the dispensing act shall be referenced. o The ancestor task shall be referenced.
output	• Required o The Medication Dispense document produced shall be referenced in the Output element of this task.
owner	Same Physician or organization that creates the Dispense document.
changes of task owner	Yes The owner may change, if more than one Dispense documents are created (in case of a repeatable order).

Task attributes	Rules for the task “Dispensing”
<taskEvent>	At least one

Example XML for this XDW task:

```
 :
<ns3:taskData>

 <ns2:taskDetails>
  <ns2:id>urn:oid:3.3.3.3.3</ns2:id>
  <ns2:taskType>Dispensing</ns2:taskType>
  <ns2:name>Dispense_of_Prescription_Item</ns2:name>
  <ns2:status>COMPLETED</ns2:status>
  <ns2:actualOwner>Dr. Brum</ns2:actualOwner>
  <ns2:createdTime>2006-05-04T18:13:51.0Z</ns2:createdTime>
  <ns2:createdBy>Dr. Brum</ns2:createdBy>
  <ns2:lastModifiedTime>2006-05-04T18:13:51.0Z</ns2:lastModifiedTime>
  <ns2:renderingMethodExists>false</ns2:renderingMethodExists>
 </ns2:taskDetails>

 <!--
  The description element shall contain the PrescriptionItemId, this task is
  referring to (substanceAdministration/id element of the Prescription Item)
 -->
 <ns2:description>4711^^^&amp;1.2.3.4.5.6.7.8.9&amp;ISO</ns2:description>

 <!-- input documents -->
 <ns2:input>
  <ns2:part name="Prescription_Document">
   <!-- Prescription document according to PRE Profile -->
   <!-- uid: the document uniqueId, home: the homeCommunityId -->
   <reference uid="urn:oid:1.2.3.4.4.4" home="urn:oid:1.2.3"/>
  </ns2:part>
  <ns2:part name="Pharmaceutical_Advice_Document">
   <!-- Pharmaceutical Advice document according to PADV Profile -->
   <!-- uid: the document uniqueId, home: the homeCommunityId -->
   <reference uid="urn:oid:1.2.3.4.4.5" home="urn:oid:1.2.3"/>
  </ns2:part>
  <ns2:part name="Ancestor_task">
   <!-- Ancestor task -->
   <reference taskId="urn:oid:2.2.2.2.2"/>
  </ns2:part>
 </ns2:input>

 <!-- output documents -->
 <ns2:output>
  <ns2:part name="Dispense_Document">
   <!-- Dispense document according to DIS Profile -->
```

```
    <!-- uid: the document uniqueId, home: the homeCommunityId -->
    <reference uid="urn:oid:1.2.3.4.4.6" home="urn:oid:1.2.3"/>
   </ns2:part>
 </ns2:output>

</ns3:taskData>
 :
```

Glossary

Add the following terms to the Glossary:

No new terms