

Integrating the Healthcare Enterprise



5

IHE Pharmacy Technical Framework Supplement

10

Community Medication Prescription and Dispense (CMPD)

15

Trial Implementation

20

Date: December 31, 2011

Author: Jürgen Brandstätter

25 Email: pharmacy@ihe.net

Foreword

30 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 submitted for Trial Implementation as of December 31, 2011 and will 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
35 <http://www.ihe.net/pharmacy/pharmacycomments.cfm> or by email to pharmacy@ihe.net.

This supplement introduces a new forthcoming technical framework and where indicated amends text by addition (**bold underline**) or removal (**~~bold strikethrough~~**), as well as addition of large new sections introduced by editor’s instructions to “add new text” or similar, which for readability are not bolded or underlined. “Boxed” instructions like the sample below indicate to
40 the Volume Editor how to integrate the relevant section(s) into the relevant Technical Framework volume:

<i>Replace Section X.X by the following:</i>
--

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

Information about the IHE Pharmacy domain can be found at:
<http://www.ihe.net/Domains/index.cfm>

Information about the structure of IHE Technical Frameworks and Supplements can be found at:
<http://www.ihe.net/About/process.cfm> and <http://www.ihe.net/profiles/index.cfm>

50 The current version of IHE Technical Frameworks can be found at:
http://www.ihe.net/Technical_Framework/index.cfm

CONTENTS

55

INTRODUCTION5

 OPEN ISSUES AND QUESTIONS6

 CLOSED ISSUES.....6

VOLUME 1 – INTEGRATION PROFILES8

60 1.N COPYRIGHT PERMISSION8

2.1 DEPENDENCIES AMONG INTEGRATION PROFILES8

 2.2.4 *Community Medication Prescription and Dispense Integration Profile*.....8

4 COMMUNITY MEDICATION PRESCRIPTION AND DISPENSE INTEGRATION PROFILE.....9

4.1 ACTORS/ TRANSACTIONS.....10

65 4.1.1 *Actors*12

 4.1.1.1 Community Pharmacy Manager12

 4.1.1.2 Prescription placer.....13

 4.1.1.3 Pharmaceutical adviser13

 4.1.1.4 Medication dispenser.....13

70 4.1.1.5 Registry/Repository actors13

4.1.2 *Transactions*.....14

 4.1.2.1 Query Pharmacy Documents14

 4.1.2.2 Registry Stored Query15

 4.1.2.3 Provide and Register Document Set-b.....15

75 4.1.2.4 Retrieve Document Set.....15

4.2 CMPD INTEGRATION PROFILE OPTIONS.....16

 4.2.1 *Workflow Management Option*16

4.3 CMPD ACTOR GROUPINGS AND PROFILE INTERACTIONS16

4.4 CMPD PROCESS FLOW17

80 4.4.1 *Use Case community pharmacy-active substance, publish & pull (Scenario 1: “Including validation step”)*17

 4.4.1.1 Story Board19

 4.4.1.2 Sequence Diagram20

4.4.2 *Use Case community pharmacy-active substance, publish & pull (Scenario 2: “Not including validation step”)*21

85 4.4.2.1 Story Board22

4.4.2.2 Sequence Diagram23

4.5 CMPD SECURITY CONSIDERATIONS24

4.6 CMPD IMPLEMENTATION SCENARIOS.....25

90 4.6.1 *Example scenario (simple)*.....25

 4.6.1.1 Demonstration of use-case 1 in example scenario (simple).....26

4.6.2 *Example scenario (complex)*.....33

 4.6.2.1 Demonstration of use-case 1 in example scenario (complex).....34

APPENDIX A ACTOR SUMMARY DEFINITIONS.....41

95 **APPENDIX B TRANSACTION SUMMARY DEFINITIONS41**

VOLUME 2 – TRANSACTIONS42

3.0 IHE TRANSACTIONS43

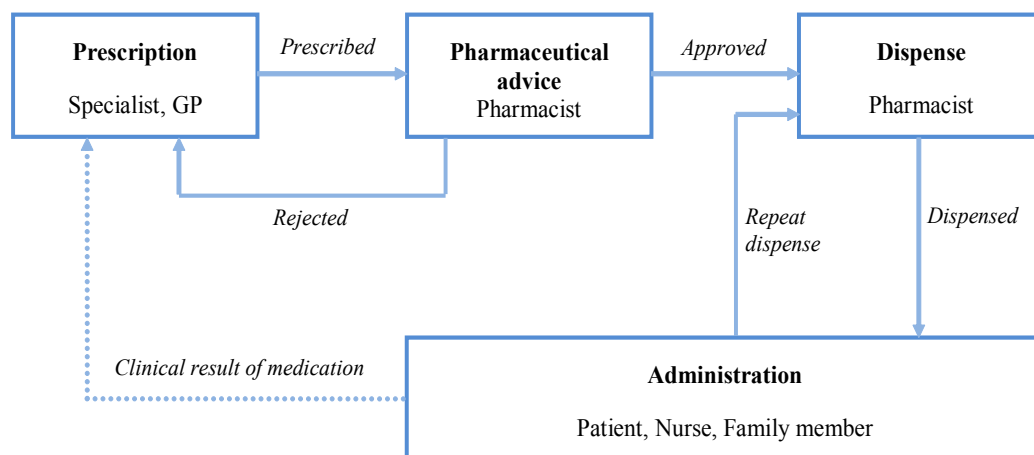
	3.1 QUERY PHARMACY DOCUMENTS [PHARM-1].....	43
100	3.1.1 <i>Scope</i>	43
	3.1.2 <i>Use Case Roles</i>	44
	3.1.3 <i>Referenced Standard</i>	44
	3.1.4 <i>Interaction Diagram</i>	44
105	3.1.4.1 Query Pharmacy Documents.....	44
	3.1.4.1.1 Trigger Events.....	45
	3.1.4.1.2 Message Semantics	45
	3.1.4.1.2.1 Required Queries	45
	3.1.4.1.2.2 Stored Query IDs	54
	3.1.4.1.2.3 Web Services Transport.....	54
110	3.1.4.1.3 Expected Actions	57
	3.1.4.1.3.1 Sample Query Request.....	57
	3.1.4.1.3.2 Sample Query Response	58
	3.1.5 <i>Security Considerations</i>	61
115	3.1.5.1 Security Audit Considerations.....	61
	3.1.5.1.1 Querying actor audit message:.....	61
	3.1.5.1.2 Community Pharmacy Manager audit message:	63
	3.1.5.1.(z) Actor Specific Security Considerations	65
	4 WORKFLOW DEFINITIONS.....	65
	4.1 COMMUNITY MEDICATION PRESCRIPTION AND DISPENSE WORKFLOW DEFINITION (CMPD-WD).....	65
120	4.1.1 <i>Actors and Grouping</i>	66
	4.1.2 <i>XDW Workflow Document – Common Attributes</i>	66
	4.1.3 <i>Workflow Task Definition</i>	67
	4.1.1.1 Task: Ordering	70
	4.1.1.2 Task: Validation	73
	4.1.1.3 Task: Dispensing.....	77
125	GLOSSARY	81

130

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 document¹.

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:



140 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 Profiles²:

- 145 • Pharmacy Prescription Supplement (PRE)

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

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.

150 This supplement also references other documents³. 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](#)
- 155 4. [IT Infrastructure Technical Framework Volume 3](#)
5. HL7 and other standards documents referenced in this document

Open Issues and Questions

- 160 • 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

- 165 • 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.
- 170 • 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://www.ihe.net/Technical_Framework/index.cfm. The remaining documents can be obtained from their respective publishers.

- 175
- 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

180 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.

185 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	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

190 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 Profiles⁴:

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

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

- Pharmacy Dispense Supplement (DIS)

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.

200 *Add Section 4*

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.

205 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:

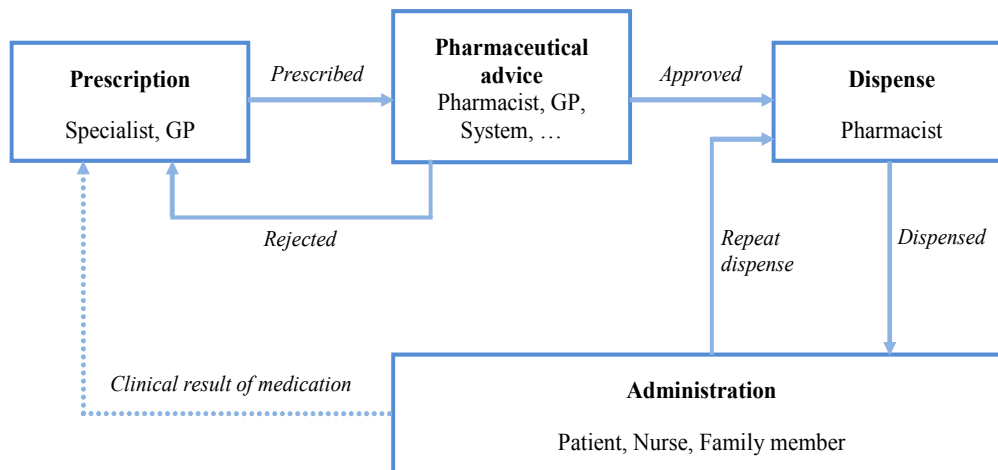


Figure 4-1 Medication Prescription and Dispense Process

210 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 Profiles⁵:

215

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

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.

220

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.

225

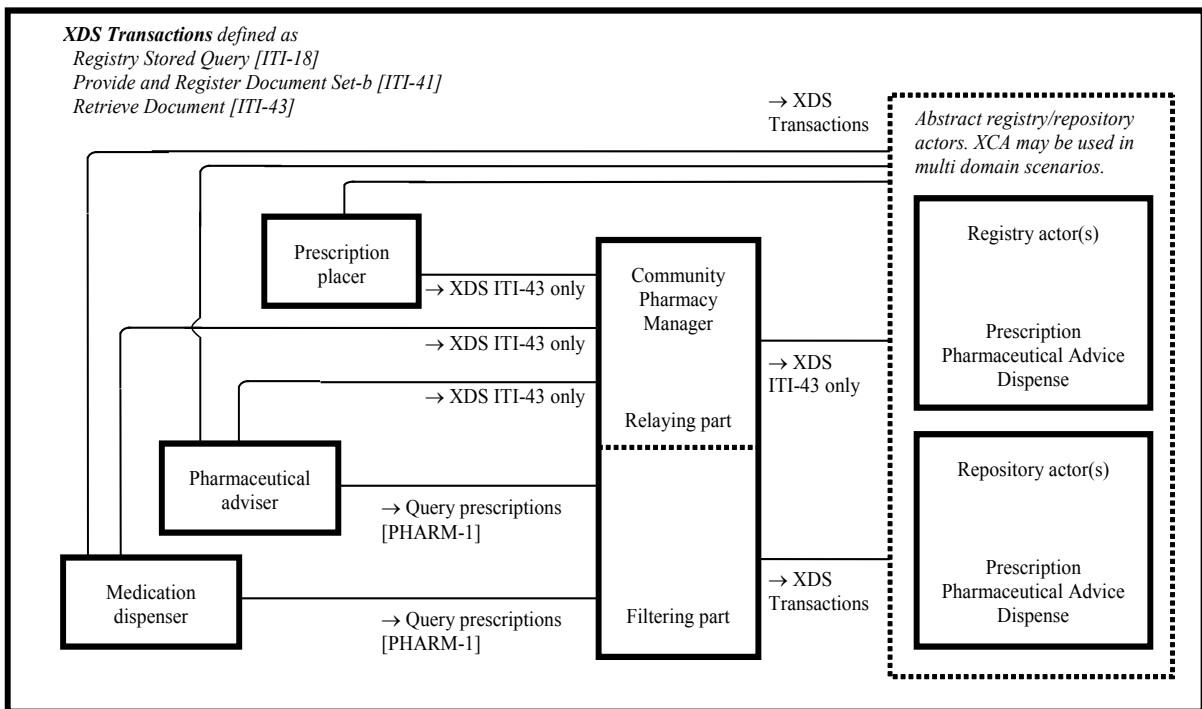


Figure 4.1-1 Community Medication Prescription and Dispense Actor Diagram

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

230

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 places, Pharmaceutical Adviser, Medication Dispenser) and the Registry/Repository actors (Prescription, Pharmaceutical Advice, Dispense repositories).

Especially the fact that the **ITI-41** “Provide and Register Document Set-b” transaction is not relayed by the CPM actor has the implication that the CPM actor has no possibility to perform any kind of status handling *at the point of submission* of documents to reduce the amount of documents to analyse at the next PHARM-1 query (e.g., by deprecating a prescription, if the last prescription item has been dispensed). This subsequently leads to the situation, that implementers of the “filtering” part of the CPM might have to deal with the challenge to master performance issues when delivering the query result (due to the fact that large amounts of documents have to be retrieved, parsed and assembled to determine the status of each prescription).

The topic of relaying the ITI-41 transaction by the CPM has been extensively discussed and many pro’s and contra’s have been discovered. The overall decision on this point was not to make any anticipations or decisions regarding status-management (not even in topology or architectural decisions like relaying submissions by the CPM) but rather to refer as a whole to the upcoming ITI profile for “Status-Management and Workflow” for solving the topic of status-management thoroughly.

235

240

245

250

255

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” 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.

260

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)	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
	Query Pharmacy Documents (PHARM-1)	R	PHARM-TF-2:3.1

Actors	Transactions	Optionality	Section in Vol. 2
Prescription placer	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
Pharmaceutical adviser	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
	Query Pharmacy Documents (PHARM-1)	R	PHARM-TF-2:3.1
Medication dispenser	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
	Query Pharmacy Documents (PHARM-1)	R	PHARM-TF-2:3.1
Registry actors Prescription Pharmaceutical Advice Dispense	Registry Stored Query (ITI-18)	R	ITI-TF-2a:3.18
Repository actors Prescription Pharmaceutical Advice Dispense	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

265 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. 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.

270 It provides special query-transactions which consuming actors (prescription placer, pharmaceutical adviser or medication dispenser) used for reducing the amount of data flowing to them. They return just “relevant” information for specific purposes (e.g., returning just all “active” prescriptions ready for being validated or dispensed together with all related documents). This actor is usually a system actor without human participation.

275 **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 or invalidation, for example). It sends the cancellation 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.

280 **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.

285 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

290 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
295 usually a pharmacist or a pharmacist assistant.

4.1.1.5 Registry/Repository actors

300 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
- 305 • This repository contains the medication prescribed to the patient from the Prescription placer and may receive updates to the current treatment (cancellations, changes, etc.). It also provides information about the current prescribed medication to other actors such as the Community Pharmacy Manager.
- Pharmaceutical repository

- 310
- 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
- 315
- 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.

- 320
- 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).

- 325
- 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

- 330
- 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:

- Pharmaceutical Adviser
- Medication Dispenser

- 335
- Specialized queries allow the finding of prescriptions and their related documents for specific purposes (e.g., for validation).

These are:

- 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
- 340

4.2 CMPD Integration Profile Options

380 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	<i>No options defined</i>	--
Prescription placer	<i>Workflow Management</i>	see Vol 2: 4
Pharmaceutical adviser	<i>Workflow Management</i>	see Vol 2: 4
Medication dispenser	<i>Workflow Management</i>	see Vol 2: 4
Registry actor Prescription Pharmaceutical advice Dispense	<i>No options defined</i>	--
Repository actor Prescription Pharmaceutical advice Dispense	<i>No options defined</i>	--

385 4.2.1 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	The prescription placer actor shall create Prescription documents according to the PRE content profile.
Pharmaceutical adviser	Content Creator: PADV Content Consumer: PRE, PADV, DIS	The prescription placer actor shall create Pharmaceutical Advice documents according to the PADV content profile.
Medication dispenser	Content Creator: DIS Content Consumer: PRE, PADV, DIS	The prescription placer actor shall create Medication Dispense documents according to the DIS content profile.

390 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.

395 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
400 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.

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

Workflow scenarios

410 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**

415 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.

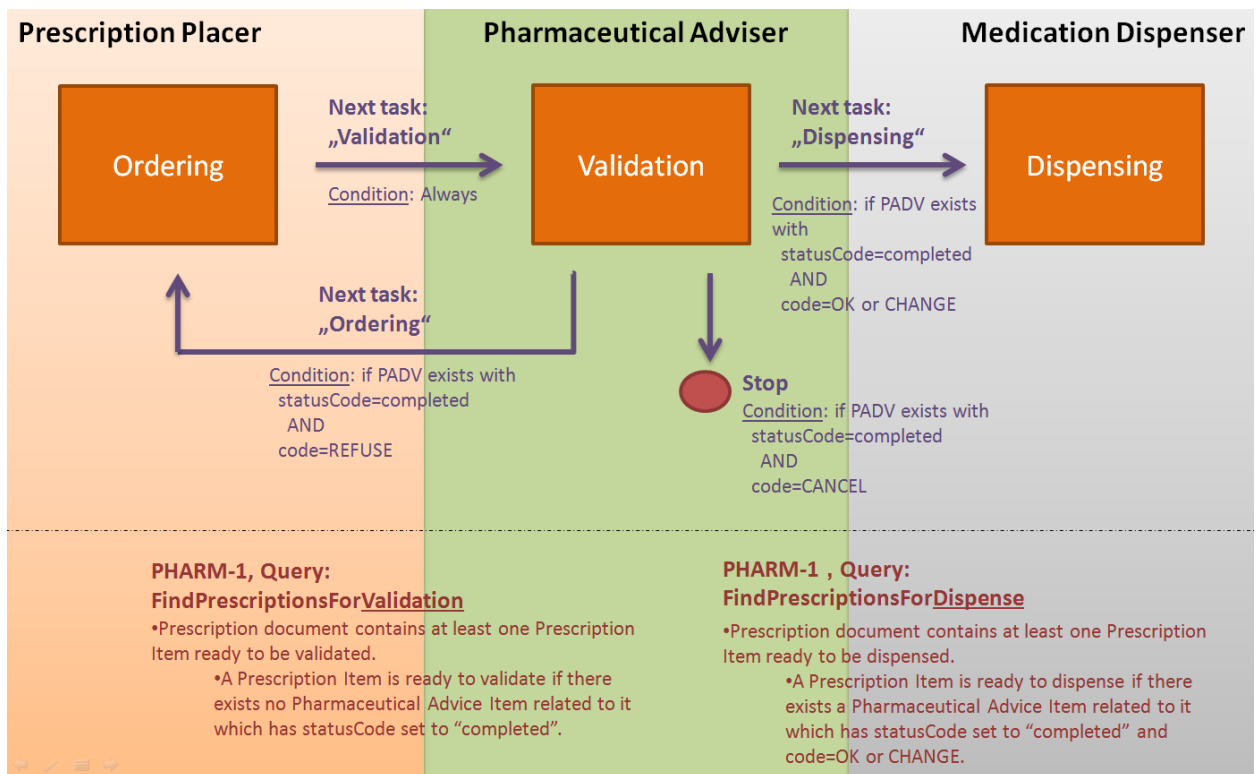
4.4.1 Use Case community pharmacy-active substance, publish & pull (Scenario 420 1: “Including validation step”)

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.

425 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)
- ... 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)



440 **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.

445 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 3.1.4.1.2.1.1.1 FindPrescriptionsForValidation
 - 450 • 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

455 **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”.

460 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 pharmacist checks for interactions and finds nothing outstanding. The information on the pharmaceutical advice is electronically sent to the
465 “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”.

470

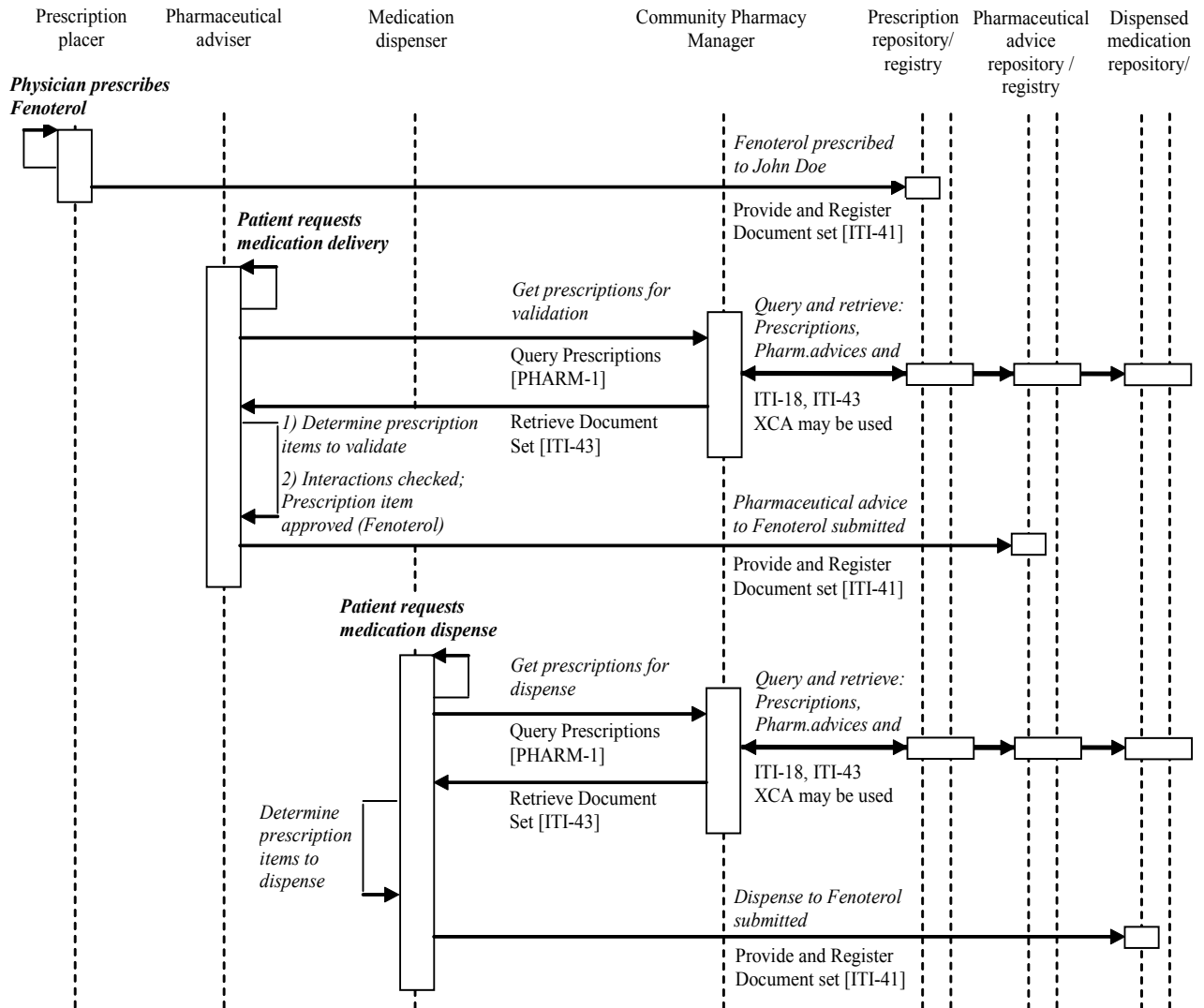
Note

The use-case does not yet include the process step of getting the “current medication”, which is needed for checking interactions (ICAs) to the prescribed item. This is an open issue and will be covered in future versions of the profile.

475

4.4.1.2 Sequence Diagram

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



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”)

490 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

- 495 • ... (workflow) tasks
 - Ordering, Dispensing
- ... which actor performs the task
 - Prescription Placer, Medication Dispenser
- ... the conditions leading to the next task
- 500 • ... 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)

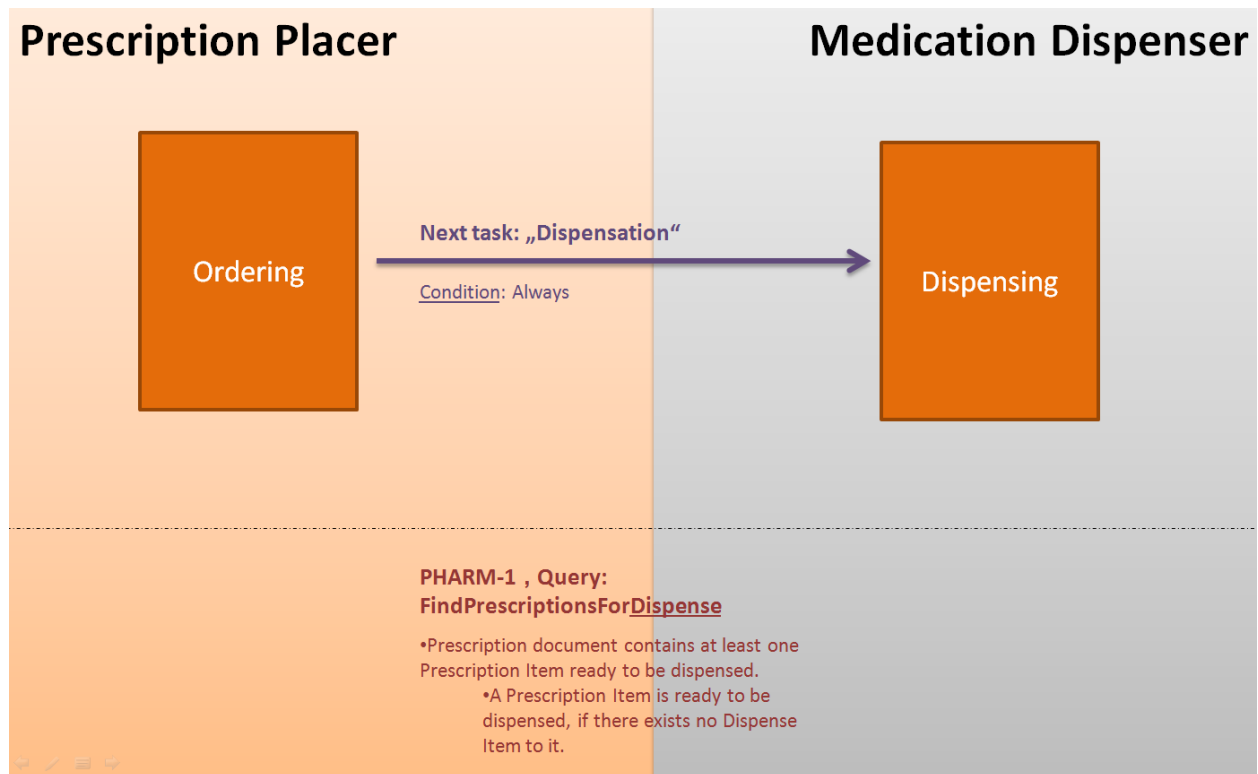


Figure 4.4.2-1 Scenario 2: Overall context of the workflow

505 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:

- 510
- 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

515 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”.

520 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).

525 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”.

Note

The use-case does not yet include the process step of getting the “current medication”, which is needed for checking interactions (ICAs) to the prescribed item.

530 This is an open issue and will be covered in future versions of the profile.

4.4.2.2 Sequence Diagram

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

535

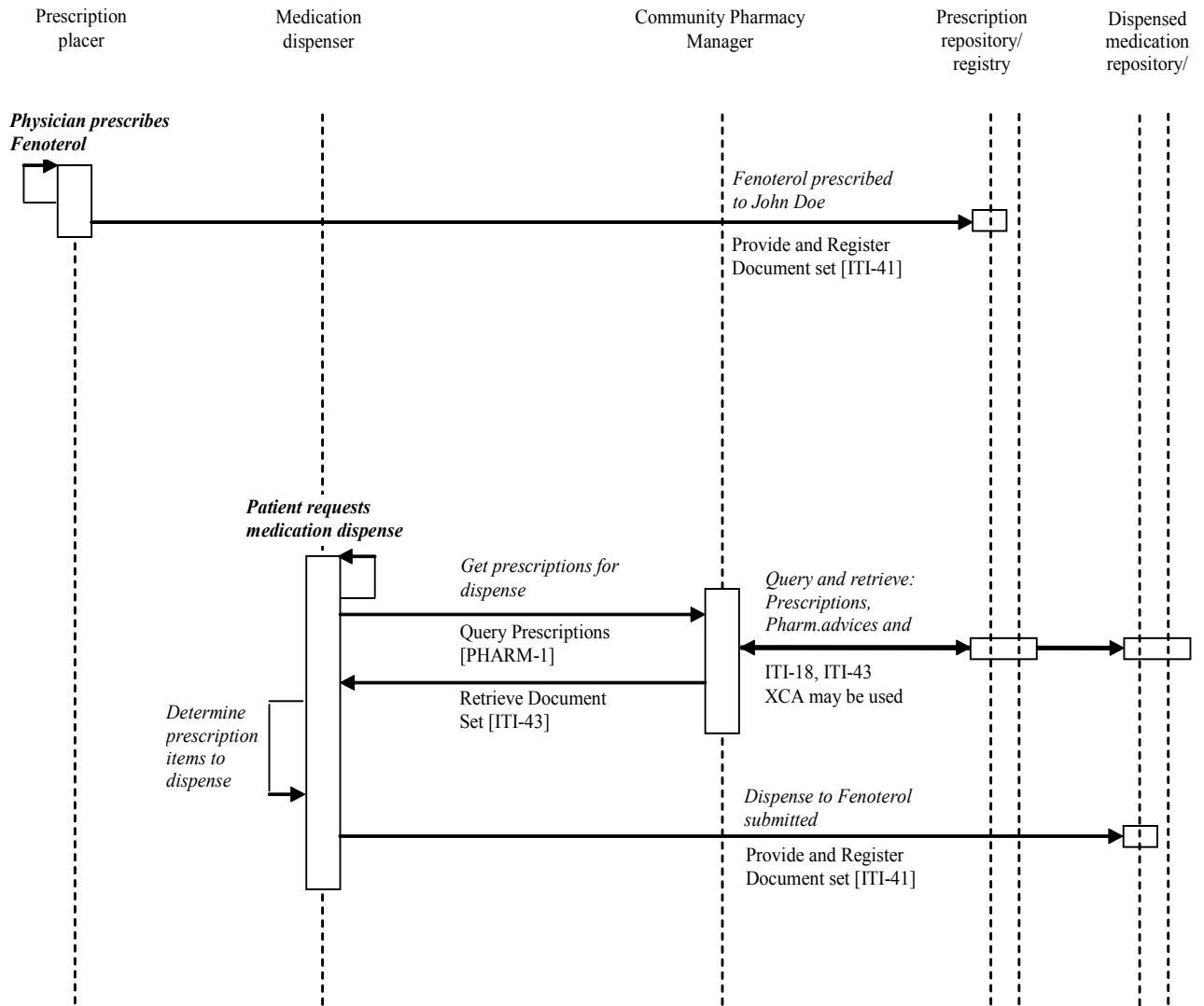


Figure 4.3.1.2-1 Use Case community pharmacy-active substance, publish & pull - Process Flow (Scenario 2: “Not including validation step”)

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

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).

545

4.6 CMPD Implementation Scenarios

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

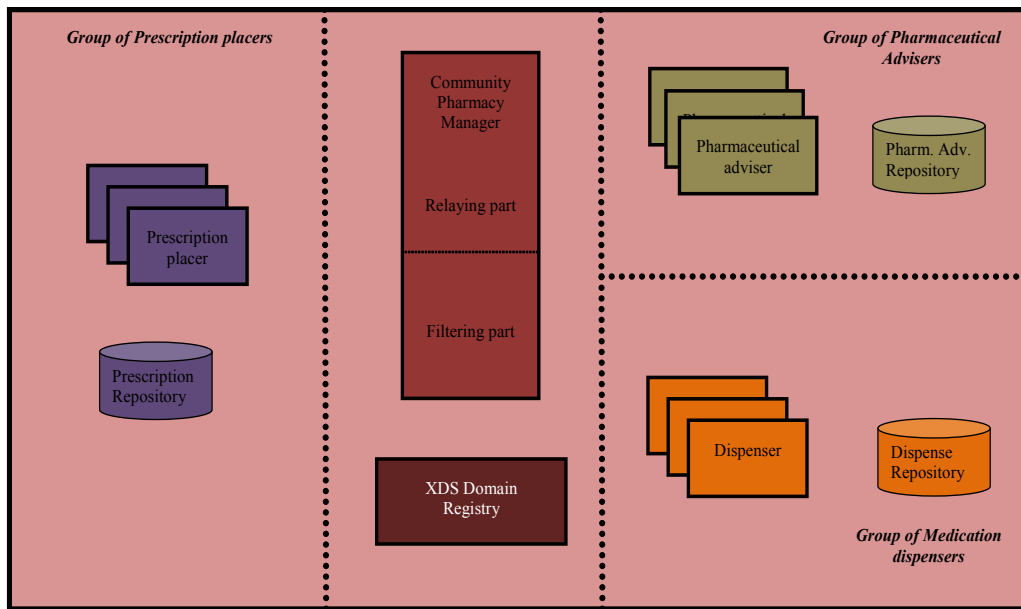
550 The prescription and dispense process of real-world projects involves several parties acting in the different abstract roles (Prescription placer, Pharmaceutical adviser, Medication dispenser). The Prescription placer role is usually taken by physicians, the Pharmaceutical adviser and Medication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

555 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.

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

4.6.1 Example scenario (simple)

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



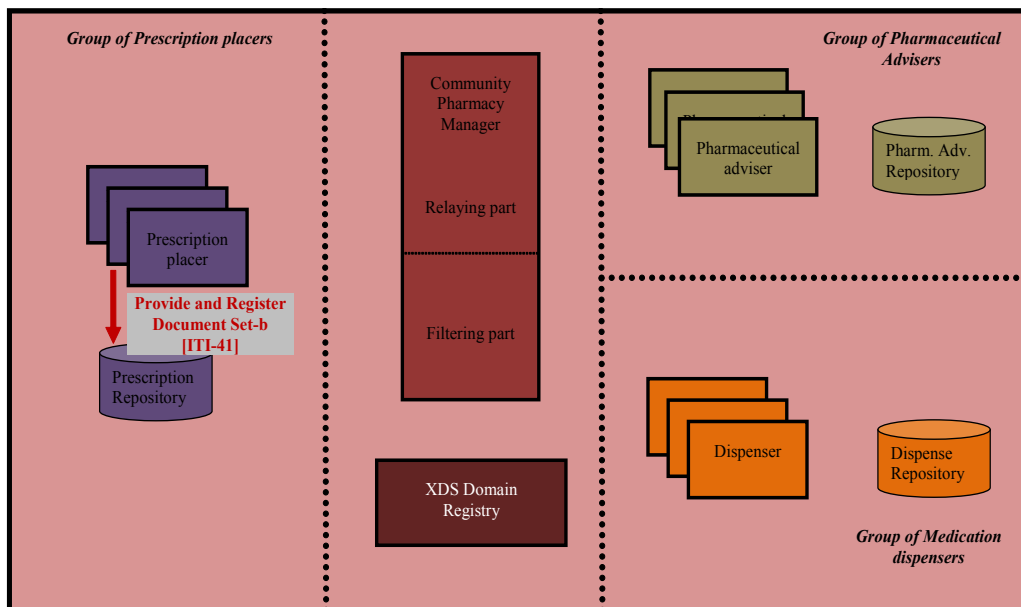
Description of the example scenario

570 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.⁶

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

Step 1: Prescription placer creates a prescription

575 The Prescription document is submitted to the appropriate Prescription repository.



Step 2: Pharmaceutical adviser queries the prescription

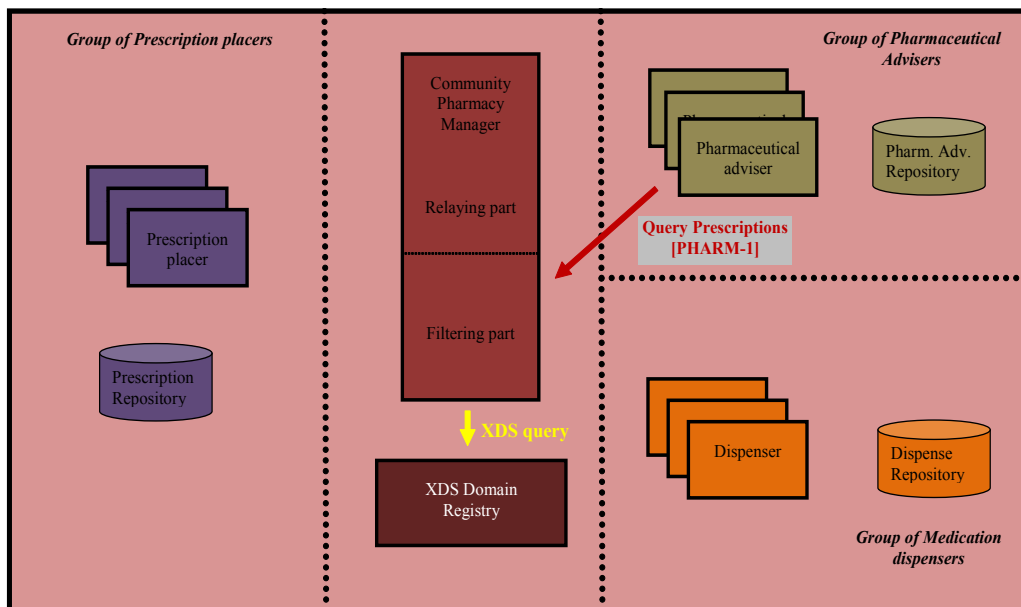
580 The Pharmaceutical adviser queries the prescription by using transaction PHARM-1, query “**FindPrescriptionsForValidation**”.

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.

⁶ 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.

585 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.

590 Future versions of this profile will define additional transactions, like “Query current medication”⁷, which might be important for the pharmacist to do validation. Such additional queries and their related business logic will be also provided by the CPM.



595

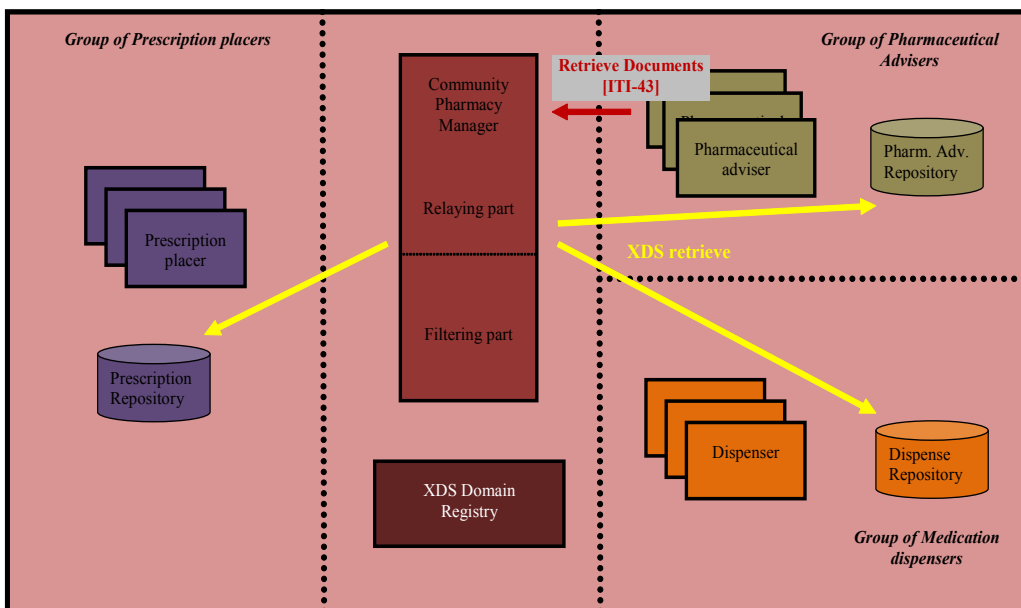
⁷ Exact definition of this query has not yet been discussed (e.g., query all dispense repositories for dispensed medication to a patient and filter this data by validity and range calculations to determine the current medication-status of the patient).

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.

600 The CPM acts as a relaying entity and accesses all requested repositories for retrieving the documents. Then it returns them to the calling client.

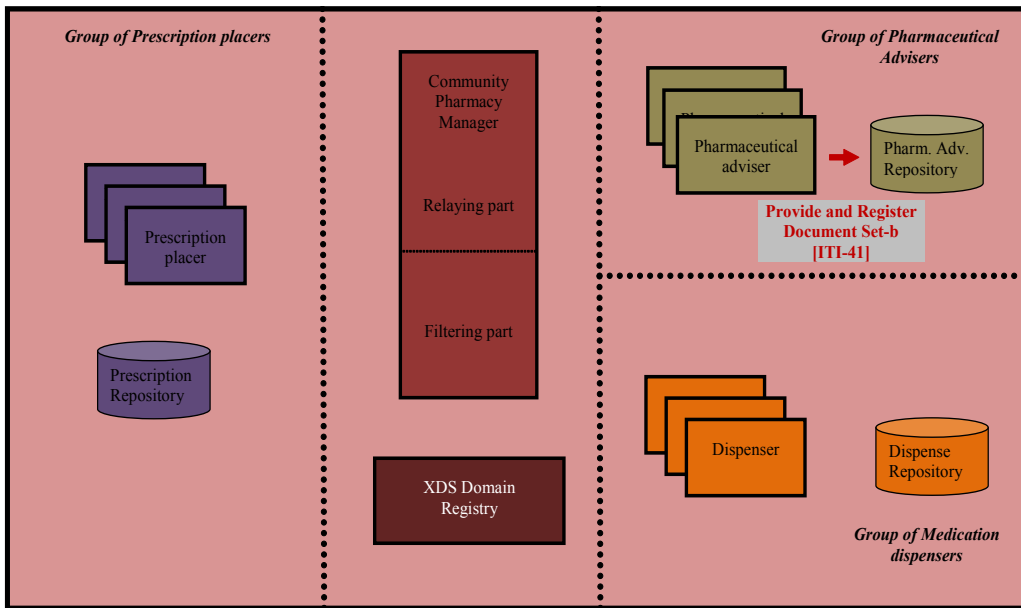
The Pharmaceutical adviser actor (as a machine) parses and relinks the returned documents by their document IDs. Then the system or the human operator performs validation and proceeds with step 4.



605

Step 4: Pharmaceutical adviser submits a pharmaceutical advice

610 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.



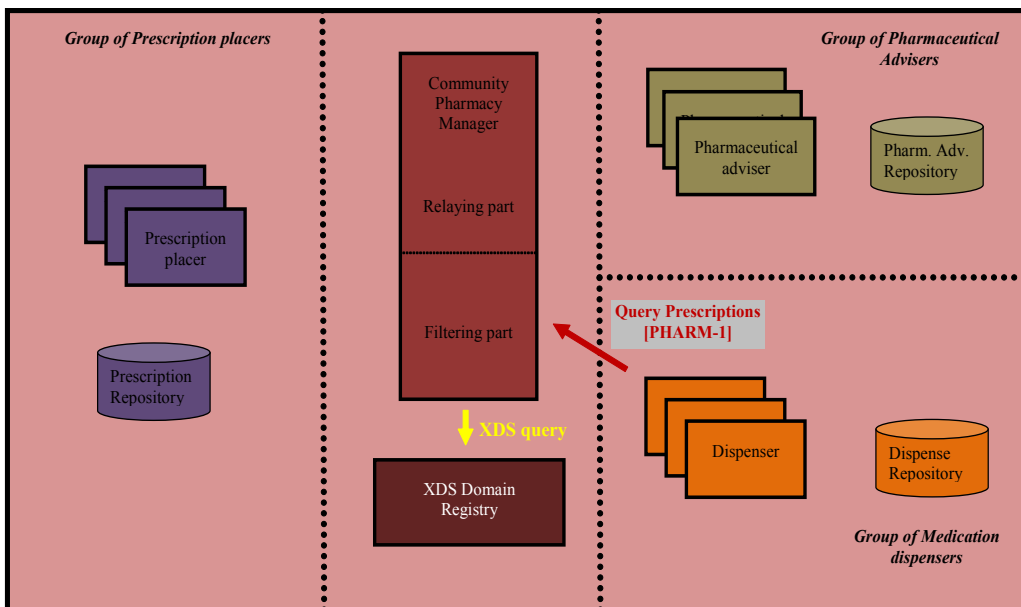
615

Step 5: Medication dispenser queries the prescription

The Medication dispenser queries the prescription by using transaction PHARM-1, query “**FindPrescriptionsForDispense**”.

620 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.

625 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.

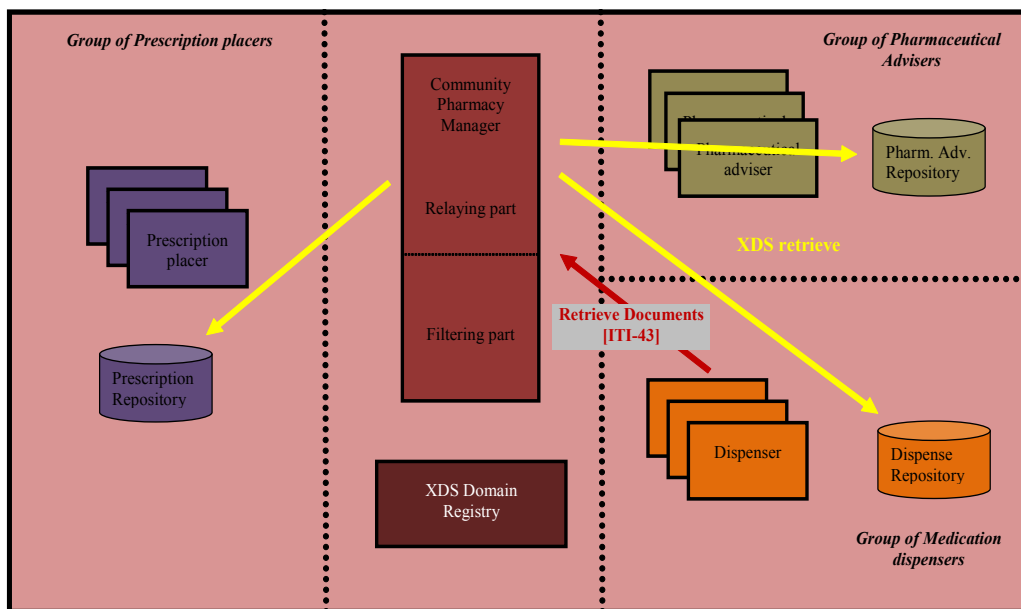


630 Step 6: Medication dispenser retrieves the documents of the query result

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. Then it returns them to the calling client.

635 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.

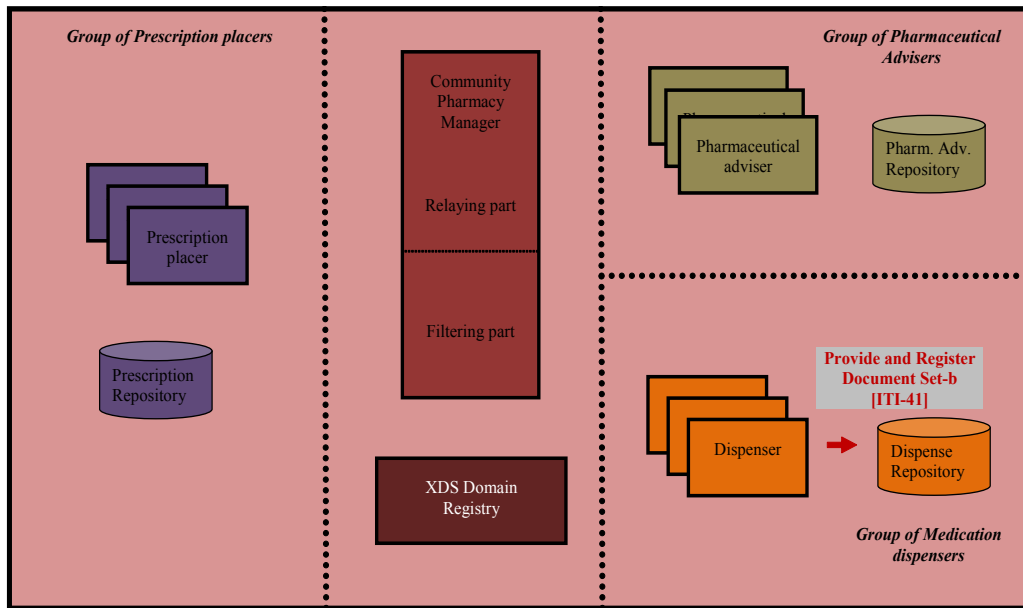


640

Step 7: Medication dispenser submits a dispense

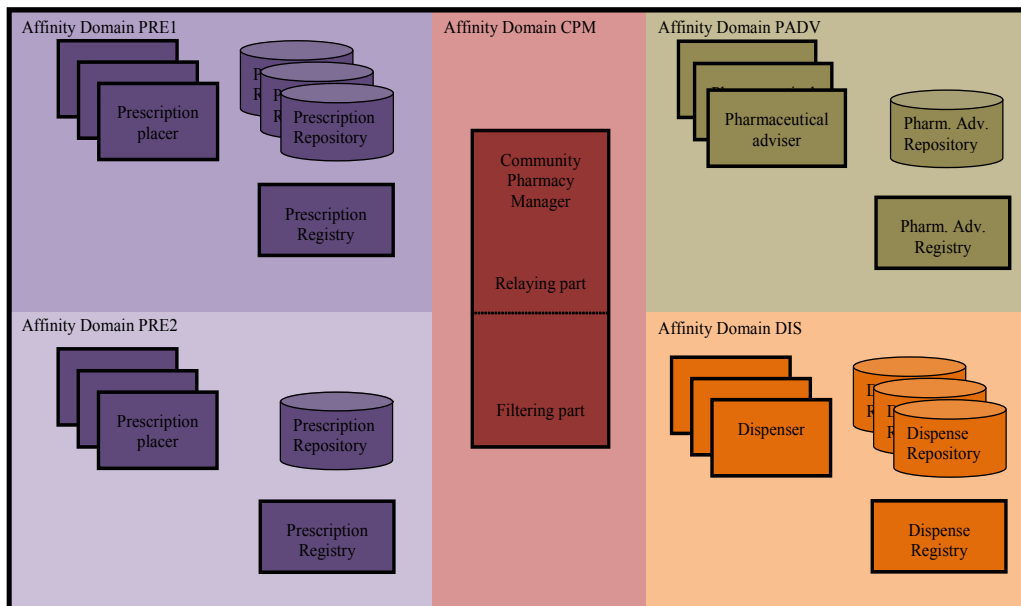
After the dispense has taken place it is documented in a Dispense document. This document is submitted to the appropriate Medication dispensed repository.

645



4.6.2 Example scenario (complex)

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



655

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.

660 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).

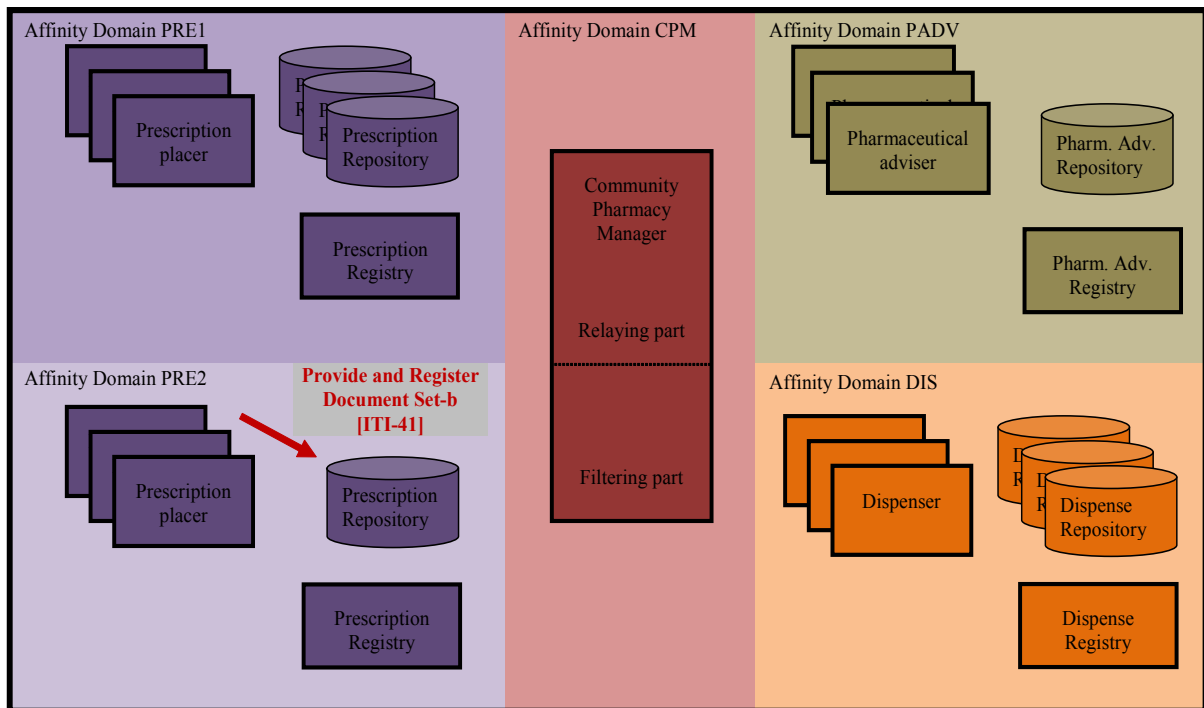
665

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.

670



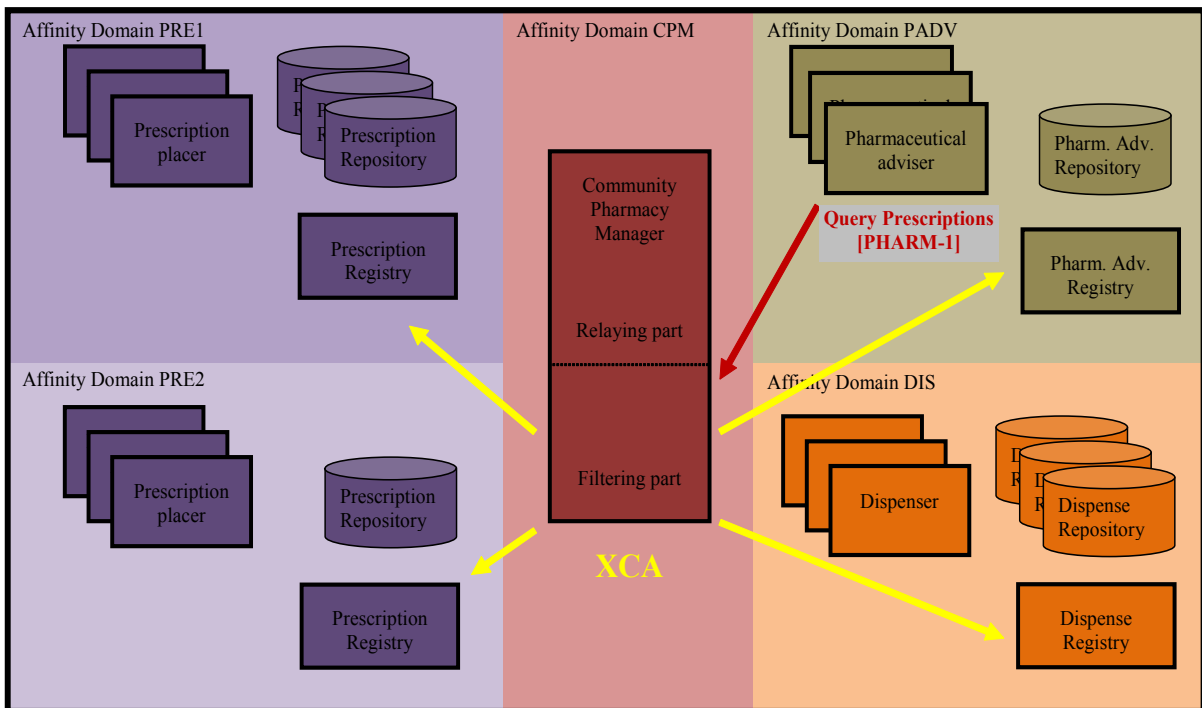
Step 2: Pharmaceutical adviser queries the prescription

675 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.

680 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.

685 Future versions of this profile will define additional transactions, like “Query current medication”⁸, which might be important for the pharmacist to do validation. Such additional queries and their related business logic will be also provided by the CPM.



⁸ Exact definition of this query has not yet been discussed (e.g., query all dispense repositories for dispensed medication to a patient and filter this data by validity and range calculations to determine the current medication-status of the patient.

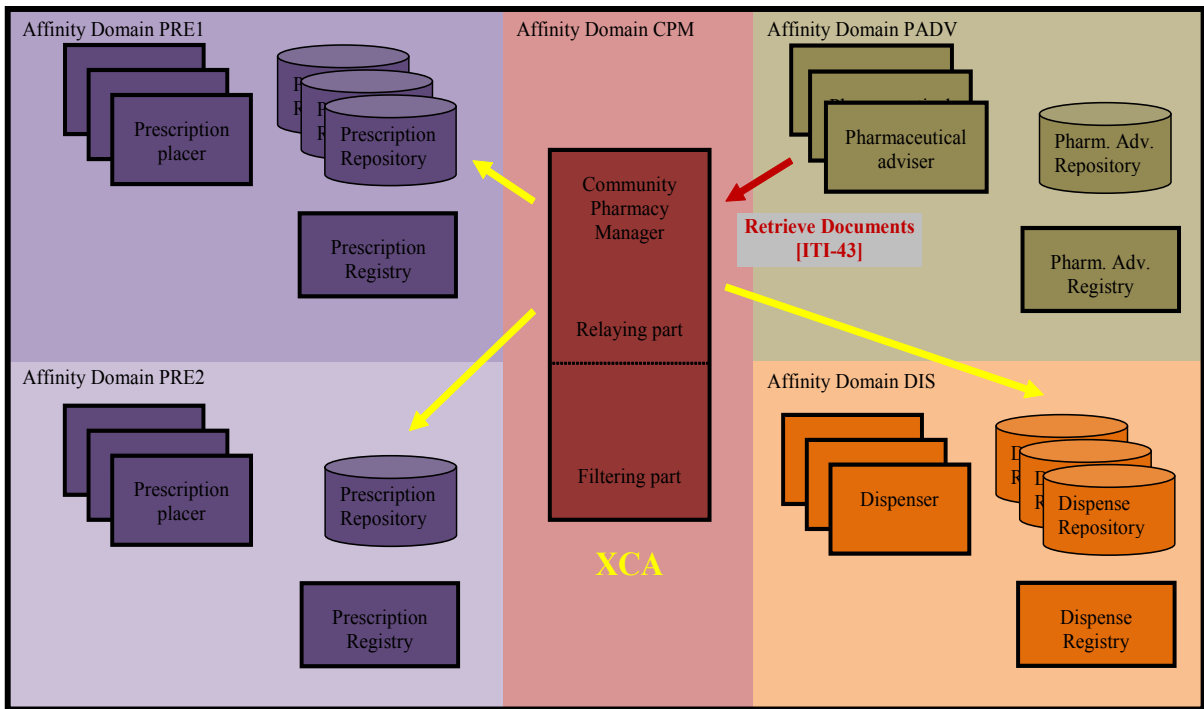
Step 3: Pharmaceutical adviser retrieves the documents of the query result

690 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).

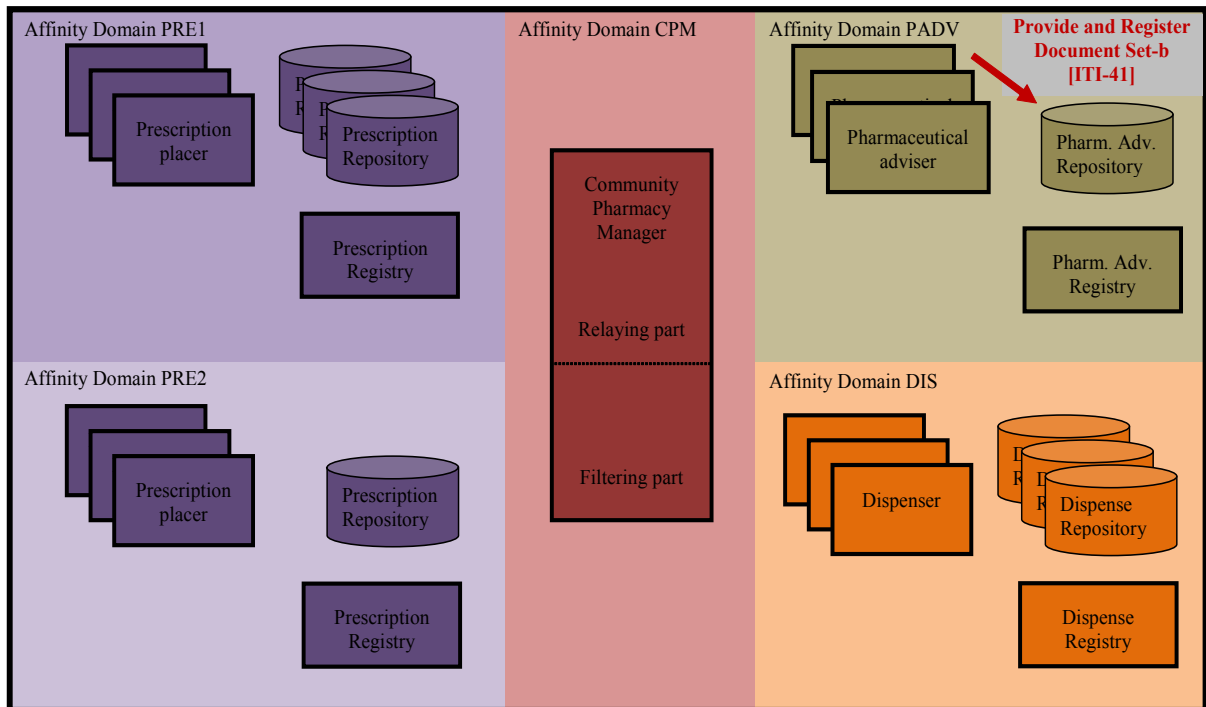
695

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.



Step 4: Pharmaceutical adviser submits a pharmaceutical advice

705 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.



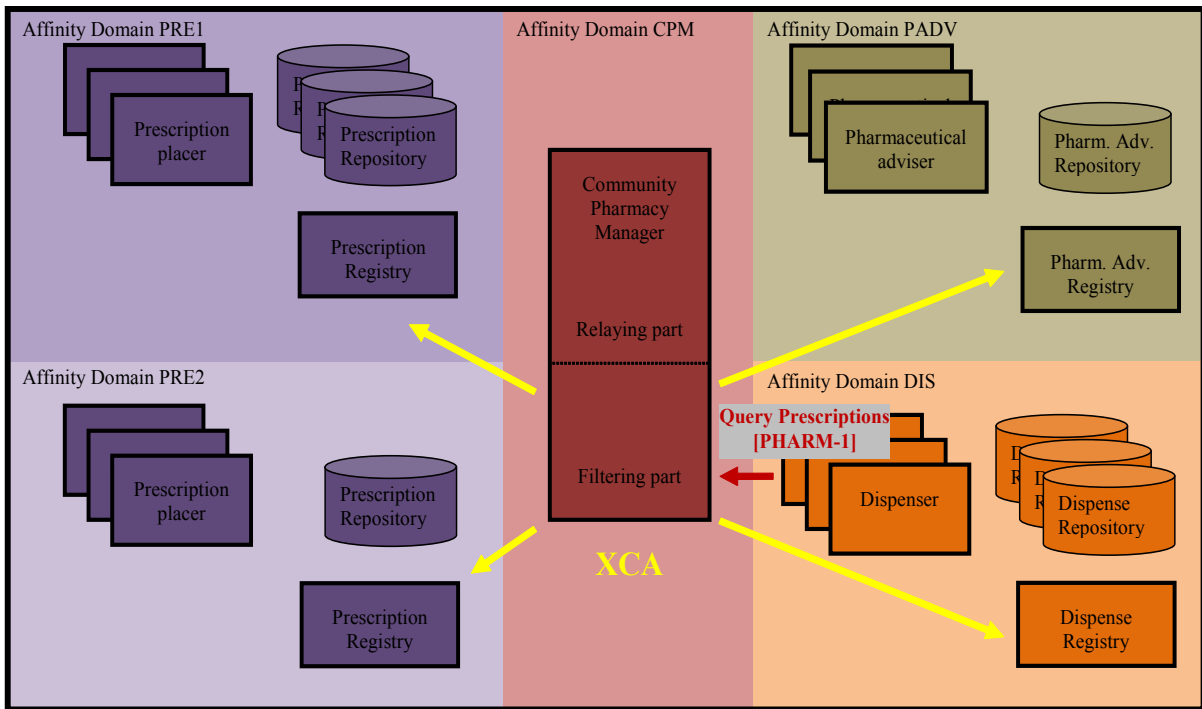
710 Step 5: Medication dispenser queries the prescription

The Medication dispenser queries the prescription by using transaction PHARM-1, query “**FindPrescriptionsForDispense**”.

715 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.

720



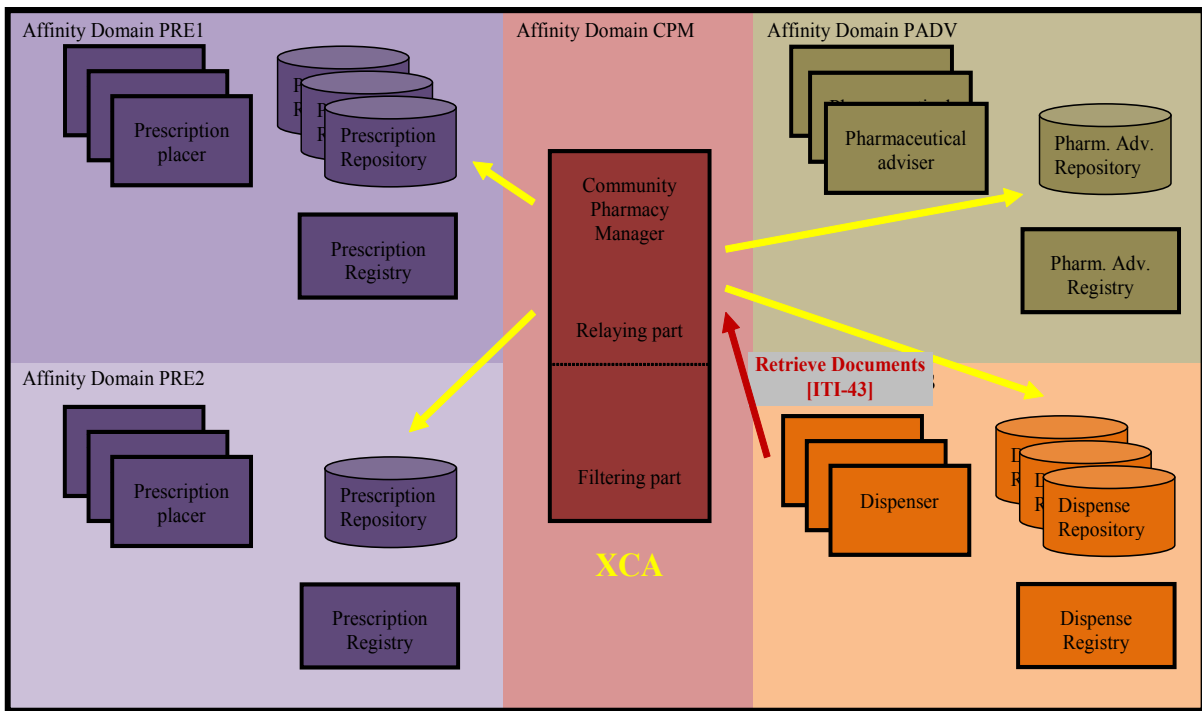
Step 6: Medication dispenser retrieves the documents of the query result

725 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).

730

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.

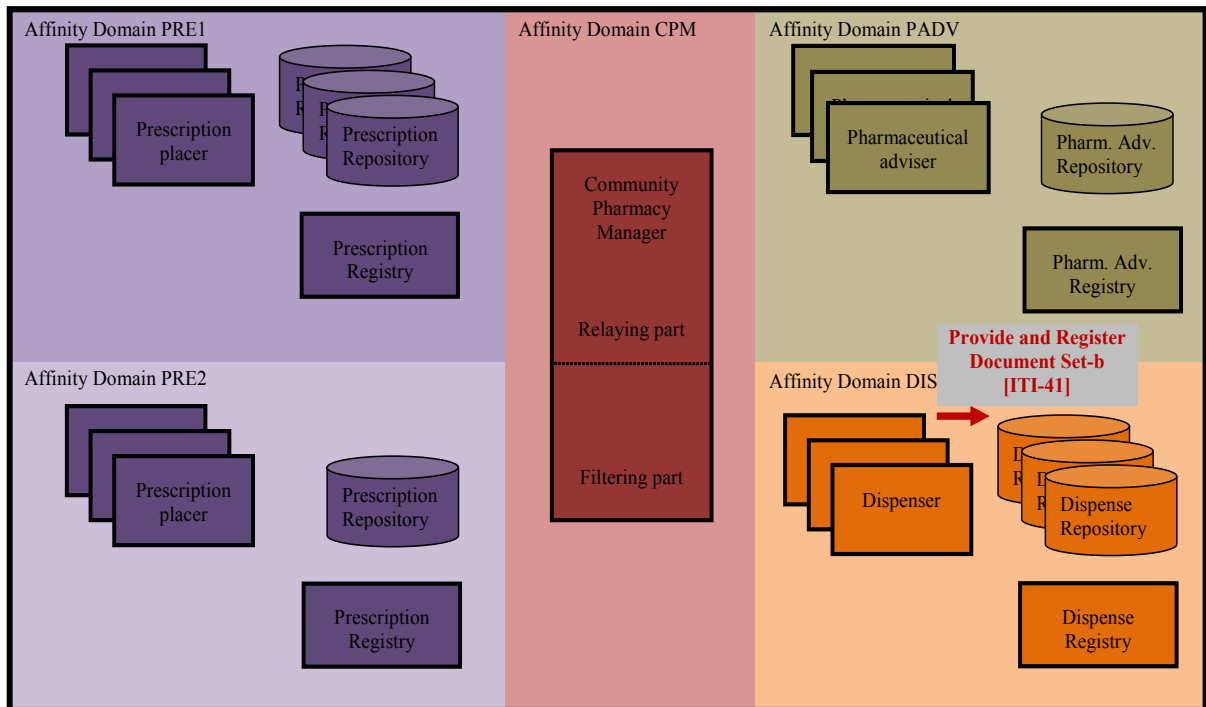


735

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 Medication dispensed repository.

740



745

750

Appendix A Actor Summary Definitions

755 **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.

760 **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.

765 **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.

770 **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

775 **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”).

780 **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)

785 **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

790

Add section 3.1

3.1 Query Pharmacy Documents [PHARM-1]

795 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
- 800 • Medication Dispenser

805 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 two specialized queries:

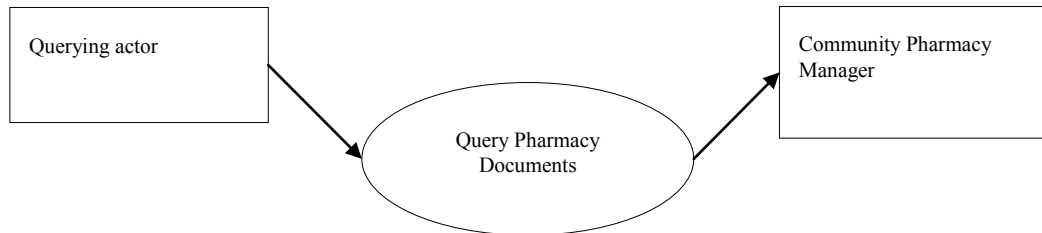
- 810 • **FindPrescriptionsForValidation**
 - Find prescriptions and their related documents containing prescription items ready to be validated
- **FindPrescriptionsForDispense**
 - 815 • Find prescriptions and their related documents containing prescription items ready to be dispensed

All queries return:

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

820

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.

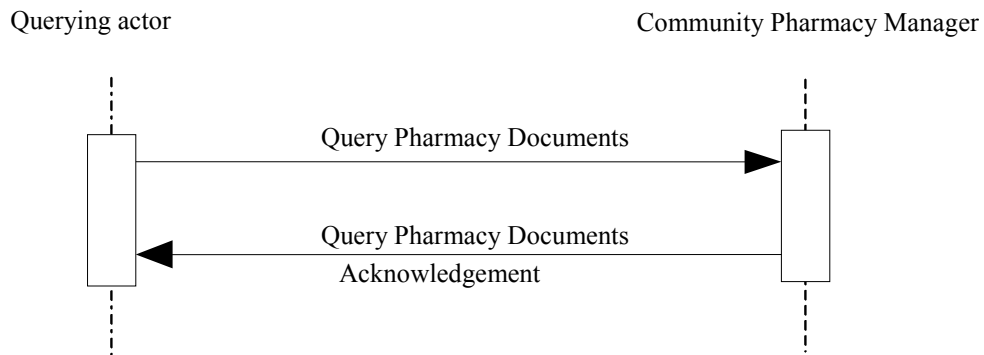
Actor: Community Pharmacy Manager

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

3.1.3 Referenced Standard

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

3.1.4 Interaction Diagram



3.1.4.1 Query Pharmacy Documents

835 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.

840 **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 Pharmaceutical adviser actor wants to find active prescriptions (and their related pharmaceutical advices and dispenses) ready to validate.
- 845 2. A Medication dispenser wants to find active prescriptions (and their related pharmaceutical advices and dispenses) which are already validated and ready for dispense.

3.1.4.1.2 Message Semantics

850 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.).

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

The provided Stored Queries are:

- **FindPrescriptionsForValidation**
 - Find prescriptions and their related documents containing prescription items ready to be validated
- 860 • **FindPrescriptionsForDispense**
 - Find prescriptions and their related documents containing prescription items ready to be dispensed

3.1.4.1.2.1.1 Parameters for Required Queries

865 This chapter defines the parameters for the Required Queries.

3.1.4.1.2.1.1.1 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.

870 **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).

This query is used in scenario 1 “Including validation step” only, so just one set of business rules is defined.

875

Scenario 1 “Including validation step”:

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

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- 880 • 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”⁹ and the result code equals OK or CHANGE¹⁰.
- 885

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**
- 890 • Pharmaceutical Advice document contains a Pharmaceutical Advice Entry Item related to a Prescription Item of the found Prescription documents.¹¹

⁹ See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details about the statusCode element (chapter “Status Code”)

¹⁰ See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details about the code element (chapter “Observation Code”)

¹¹ 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)

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**
- 895 • Dispense document contains a Dispense Entry Item related to a Prescription Item of the found Prescription documents.¹²

Explanation

900 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.

905 **Scenario 2 “Not including validation step”:**

Business rule 2.1: Returns *Prescription* documents matching the query parameters:

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- XDSDocumentEntry matches all other optional query parameters
- 910 • 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.

Business rule 2.2: Returns related *Dispense* documents to the Prescriptions found

- 915 • 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.¹³

920

¹² 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)

¹³ 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)

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.

925

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	XSDocumentEntry. patientId	R	--
\$XDSDocumentEntryEntryUUID	XSDocumentEntry. entryUUID	O ¹	M
\$XDSDocumentEntryUniqueId	XSDocumentEntry. uniqueId	O ¹	M
\$XDSDocumentEntryPracticeSettingCode ³	XSDocumentEntry. practiceSettingCode	O	M
\$XDSDocumentEntryCreationTimeFrom	Lower value of XSDocumentEntry. creationTime	O	--
\$XDSDocumentEntryCreationTimeTo	Upper value of XSDocumentEntry. creationTime	O	--
\$XDSDocumentEntryServiceStartTimeFrom	Lower value of XSDocumentEntry. serviceStartTime	O	--
\$XDSDocumentEntryServiceStartTimeTo	Upper value of XSDocumentEntry. serviceStartTime	O	--
\$XDSDocumentEntryServiceStopTimeFrom	Lower value of XSDocumentEntry. serviceStopTime	O	--
\$XDSDocumentEntryServiceStopTimeTo	Upper value of XSDocumentEntry. serviceStopTime	O	--
\$XDSDocumentEntryHealthcareFacilityTypeCode ³	XSDocumentEntry. healthcareFacilityTypeCode	O	M
\$XDSDocumentEntryEventCodeList ³	XSDocumentEntry. eventCodeList ³	O	M
\$XDSDocumentEntryConfidentialityCode ³	XSDocumentEntry. confidentialityCode ³	O	M
\$XDSDocumentEntryAuthorPerson ⁴	XSDocumentEntry. Author	O	M
\$XDSDocumentEntryStatus	XSDocumentEntry. Status	R	M

930

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

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

935 ⁴The 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 “FindPrescriptionsForValidation” query

940 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			This item is already validated and ready for dispense
	PRE Item 2-3	PADV 4		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:

- 945
- Patient ID
 - Document Status

This is what should be returned by the query:

Returned XSDDocumentEntries		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 XSDocumentEntries		Explanation
Prescriptions	Related documents	
		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.

950

Example 2: Search for a specific prescription

Query Parameters set:

- Patient ID
- Document Status
- 955 • 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 XSDocumentEntries		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.

960

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.

3.1.4.1.2.1.1.2 FindPrescriptionsForDispense

965 Find prescriptions and their related documents (XSDocumentEntry 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 XSDocumentEntry objects returned.

Returns: XSDocumentEntry objects according to the following business rules:

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

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

- FormatCode matches **urn:ihe:pharm:pre:2010**
 - Prescription document contains at least one Prescription Item ready to dispense
- 975 • A Prescription Item is ready to dispense if there exists a Pharmaceutical Advice Item related to it which has statusCode set to “completed”.¹⁴

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

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- 980 • 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.¹⁵

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.¹⁶

Explanation

990 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.

995

Query parameters:

Parameter Name	Attribute	Opt	Mult
\$XDSDocumentEntryPatientId	XDSDocumentEntry. patientId	R	--
\$XDSDocumentEntryEntryUUID	XDSDocumentEntry. entryUUID	O ¹	M

¹⁴ See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details about the statusCode element

¹⁵ 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)

¹⁶ 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)

IHE Pharmacy Technical Framework Supplement – Community Medication Prescription and Dispense (CMPD)

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

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

1000 ³Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

1005 ⁴The 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 “FindPrescriptionsForDispense” query

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

1010

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			This item is already validated and ready for dispense
	PRE Item 2-3	PADV 4		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 XSDDocumentEntries		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.

1020 **Example 2: Search for a specific prescription**

Query Parameters set:

- Patient ID
- Document Status

- 1025 • 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 XSDDocumentEntries		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.

- 1030 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.

3.1.4.1.2.2 Stored Query IDs

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

- 1035 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
FindPrescriptionsForValidation	urn:uuid: c1a43b20-0254-102e-8469-a6af440562e8
FindPrescriptionsForDispense	urn:uuid: c875eb9c-0254-102e-8469-a6af440562e8

- 1040

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.

- 1045 **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"
```

1050 query response -> "QueryPharmacyDocuments_Response_Message"
 portType -> "CommunityPharmacyManager_PortType"
 operation -> "QueryPharmacyDocuments"
 SOAP 1.2 binding -> "CommunityPharmacyManager_Binding_Soap12"
 SOAP 1.2 port -> "CommunityPharmacyManager_Port_Soap12"

1055 **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",
 - 1060 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”
- 1065 • 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”
- 1070 • 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:

```
1075 <?xml version="1.0" encoding="utf-8"?>
     <definitions ...>
       ...
       <types>
1080       <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>
1085 </message name="QueryPharmacyDocuments_Message">
     <documentation>Query Pharmacy Documents</documentation>
     <part name="body" element="query:AdhocQueryRequest"/>
     </message>
1090 <message name="QueryPharmacyDocumentsResponse_Message">
     <documentation>Query Pharmacy Documents Response</documentation>
     <part name="body" element="query:AdhocQueryResponse"/>
     </message>
     ...
1095 <portType name="CommunityPharmacyManager_PortType">
     <operation name="CommunityPharmacyManager_QueryPharmacyDocuments">
```

```
1100 <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>
      ...
1105 </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.

1110 3.1.4.1.2.3.1.1 Sample Query Prescription SOAP Request

```
1110 <s:Envelope xmlns:s=http://www.w3.org/2003/05/soap-envelope
      xmlns:a="http://www.w3.org/2005/08/addressing">
      <s:Header>
1115 <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>
1120 <a:To>http://localhost/service/CommunityPharmacyManager.svc</a:To>
      </s:Header>
      <s:Body>
      <query:AdhocQueryRequest>
      :
1125 see Sample Query Request below
      :
      </query:AdhocQueryRequest>
      </s:Body>
</s:Envelope>
```

1130 3.1.4.1.2.3.1.2 Sample Query Prescription SOAP Response

```
1135 <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
1135 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>
1140 <query:AdhocQueryResponse>
      :
      see Sample Query Response below
      :
      </query:AdhocQueryResponse>
      </s:Body>
1145 </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.

1150 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
- 1155 • 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.

```
1160 <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">
1165 <query:ResponseOption returnComposedObjects="true" returnType="LeafClass"/>
      <rim:AdhocQuery id="urn:uuid:c1a43b20-0254-102e-8469-a6af440562e8">
        <rim:Slot name="$XDSDocumentEntryPatientId">
          <rim:ValueList>
1170           <rim:Value>'st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&ISO'</rim:Value>
          </rim:ValueList>
        </rim:Slot>
        <rim:Slot name="$XDSDocumentEntryStatus">
          <rim:ValueList>
1175           <rim:Value>('urn:oasis:names:tc:ebxml-regrep:StatusType:Approved')</rim:Value>
          </rim:ValueList>
        </rim:Slot>
        <rim:Slot name="$XDSDocumentEntryCreationTimeFrom">
          <rim:ValueList>
1180           <rim:Value>200412252300</rim:Value>
          </rim:ValueList>
        </rim:Slot>
        <rim:Slot name="$XDSDocumentEntryCreationTimeTo">
          <rim:ValueList>
1185           <rim:Value>200501010800</rim:Value>
          </rim:ValueList>
        </rim:Slot>
      </rim:AdhocQuery>
    </query:AdhocQueryRequest>
```

3.1.4.1.3.2 Sample Query Response

1190 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.

1195

```
<?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-
1215 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>
1220 </rim:ValueList>
        </rim:Slot>
      <rim:Slot name="creationTime">
        <rim:ValueList>
          <rim:Value>200412261119</rim:Value>
1225 </rim:ValueList>
        </rim:Slot>
      <rim:Slot name="hash">
        <rim:ValueList>
          <rim:Value>4cf4f82d78b5e2aac35c31bca8cb79fe6bd6a41e</rim:Value>
1230 </rim:ValueList>
        </rim:Slot>
      <rim:Slot name="languageCode">
        <rim:ValueList>
          <rim:Value>en-us</rim:Value>
1235 </rim:ValueList>
        </rim:Slot>
      <rim:Slot name="serviceStartTime">
        <rim:ValueList>
          <rim:Value>200412230800</rim:Value>
1240 </rim:ValueList>
        </rim:Slot>
      </rim:Slot>
    </rim:ExtrinsicObject>
  </rim:RegistryObjectList>
</AdhocQueryResponse>
```

IHE Pharmacy Technical Framework Supplement – Community Medication Prescription and Dispense (CMPD)

```
1245     <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^I1^44130^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
1270 classificationScheme="urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a"
classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
1275 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>
1280 <rim:LocalizedString charset="UTF-8" value="Prescriptions" xml:lang="en-us"/>
      </rim:Name>
      <rim:Description/>
      </rim:Classification>
      <rim:Classification
1285 classificationScheme="urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f"
classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
1290 id="urn:uuid:fla8c8e4-3593-4777-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>
1295 <rim:LocalizedString charset="UTF-8" value="Normal" xml:lang="en-us"/>
      </rim:Name>
```

```

        </rim:Name>
        <rim:Description/>
        </rim:Classification>
        <rim:Classification
1305 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"
1310 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>
1315 <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="Pharmacy Prescription"
xml:lang="en-us"/>
        </rim:Name>
        <rim:Description/>
1320 </rim:Classification>
        <rim:Classification
classificationScheme="urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1"
classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
1325 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>
1330 </rim:ValueList>
        </rim:Slot>
        <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="Emergency Department"
xml:lang="en-us"/>
1335 </rim:Name>
        <rim:Description/>
        </rim:Classification>
        <rim:Classification
1340 classificationScheme="urn:uuid:ccc5598-8b07-4b77-a05e-ae952c785ead"
classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
id="urn:uuid:fb7677c5-c42f-485d-9010-dce0f3cd4ad5"
nodeRepresentation="Cardiology"
1345 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>
1350 <rim:LocalizedString charset="UTF-8" value="Cardiology" xml:lang="en-us"/>
        </rim:Name>
        <rim:Description/>
        </rim:Classification>
        <rim:Classification
1355 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"

```

```

1360 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>
1365 <rim:Name>
      <rim:LocalizedString charset="UTF-8" value="Prescriptions" xml:lang="en-us"/>
    </rim:Name>
    <rim:Description/>
  </rim:Classification>
1370 <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"
1375 value="st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&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"
1385 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>
1390 </AdhocQueryResponse>
1395

```

3.1.5 Security Considerations

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

1400 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/	EventID	M	EV(110112, DCM, "Query")
	EventActionCode	M	"E" (Execute)

IHE Pharmacy Technical Framework Supplement – Community Medication Prescription and Dispense (CMPD)

	Field Name	Opt	Value Constraints
EventIdentification	<i>EventDateTime</i>	M	<i>not specialized</i>
	<i>EventOutcomeIndicator</i>	M	<i>not specialized</i>
	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.
	<i>UserName</i>	U	<i>not specialized</i>
	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.
	<i>AlternativeUserID</i>	U	<i>not specialized</i>
	<i>UserName</i>	U	<i>not specialized</i>
	UserIsRequestor	M	“true”
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	<i>NetworkAccessPointTypeCode</i>	NA	
	<i>NetworkAccessPointID</i>	NA	

Destination AuditMessage/ ActiveParticipant	UserID	M	SOAP endpoint URI.
	<i>AlternativeUserID</i>	U	<i>not specialized</i>
	<i>UserName</i>	U	<i>not specialized</i>
	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.

1405

IHE Pharmacy Technical Framework Supplement – Community Medication Prescription and Dispense (CMPD)

Audit Source AuditMessage/ AuditSourceIdentification	<i>AuditSourceID</i>	<i>U</i>	<i>Not specialized.</i>
	<i>AuditEnterpriseSiteID</i>	<i>U</i>	<i>not specialized</i>
	<i>AuditSourceTypeCode</i>	<i>U</i>	<i>not specialized</i>

Patient (AuditMessage/ ParticipantObjectIdentification)	ParticipantObjectTypeCode	M	“1” (Person)
	ParticipantObjectTypeCodeRole	M	“1” (Patient)
	<i>ParticipantObjectDataLifeCycle</i>	<i>U</i>	<i>not specialized</i>
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, “Patient Number”)
	ParticipantObjectSensitivity	<i>U</i>	<i>not specialized</i>
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	<i>ParticipantObjectName</i>	<i>U</i>	<i>not specialized</i>
	<i>ParticipantObjectQuery</i>	<i>U</i>	<i>not specialized</i>
Query Parameters (AuditMessage/ ParticipantObjectIdentification)	ParticipantObjectTypeCode	M	“2” (system object)
	ParticipantObjectTypeCodeRole	M	“24” (query)
	<i>ParticipantObjectDataLifeCycle</i>	<i>U</i>	<i>not specialized</i>
	ParticipantObjectIDTypeCode	M	EV(“PHARM-1”, “IHE Transactions”, “ Query Pharmacy Documents ”)
	<i>ParticipantObjectSensitivity</i>	<i>U</i>	<i>not specialized</i>
	ParticipantObjectID	M	Stored Query ID (UUID)
	<i>ParticipantObjectName</i>	C	If known the value of <ihe:HomeCommunityId/>
	<i>ParticipantObjectQuery</i>	M	the AdhocQueryRequest, base64 encoded.
	<i>ParticipantObjectDetail</i>	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/	EventID	M	EV(110112, DCM, “Query”)
	EventActionCode	M	“E” (Execute)

IHE Pharmacy Technical Framework Supplement – Community Medication Prescription and Dispense (CMPD)

	Field Name	Opt	Value Constraints
EventIdentification	<i>EventDateTime</i>	M	<i>not specialized</i>
	<i>EventOutcomeIndicator</i>	M	<i>not specialized</i>
	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)			

Where:

Source	Field Name	Opt	Value Constraints
AuditMessage/ ActiveParticipant	UserID	M	The content of the <wsa:ReplyTo/> element.
	AlternativeUserID	U	<i>not specialized</i>
	<i>UserName</i>	U	<i>not specialized</i>
	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.

1410

Destination	Field Name	Opt	Value Constraints
AuditMessage/ ActiveParticipant	UserID	M	SOAP endpoint URI.
	<i>AlternativeUserID</i>	M	the process ID as used within the local operating system in the local system logs.
	<i>UserName</i>	U	<i>not specialized</i>
	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	Field Name	Opt	Value Constraints
AuditMessage/ AuditSourceIdentification	<i>AuditSourceID</i>	U	<i>not specialized</i>
	<i>AuditEnterpriseSiteID</i>	U	<i>not specialized</i>
	<i>AuditSourceTypeCode</i>	U	<i>not specialized</i>

Patient	Field Name	Opt	Value Constraints
(AuditMessage/ ParticipantObjectIdentification)	ParticipantObjectTypeCode	M	“1” (Person)
	ParticipantObjectTypeCodeRole	M	“1” (Patient)
	<i>ParticipantObjectDataLifeCycle</i>	U	<i>not specialized</i>
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, “Patient Number”)

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

1415 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.

1420 4.1 Community Medication Prescription and Dispense Workflow Definition (CMPD-WD)

The management of the workflow related to the CMPD profile is involved in many 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

1425 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.

1430 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 an other in a consistent way.

1440 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).

1445

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.
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

1450 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. In this section only the use of some specific metadata for the use of XDW in the CMPD context is specified.

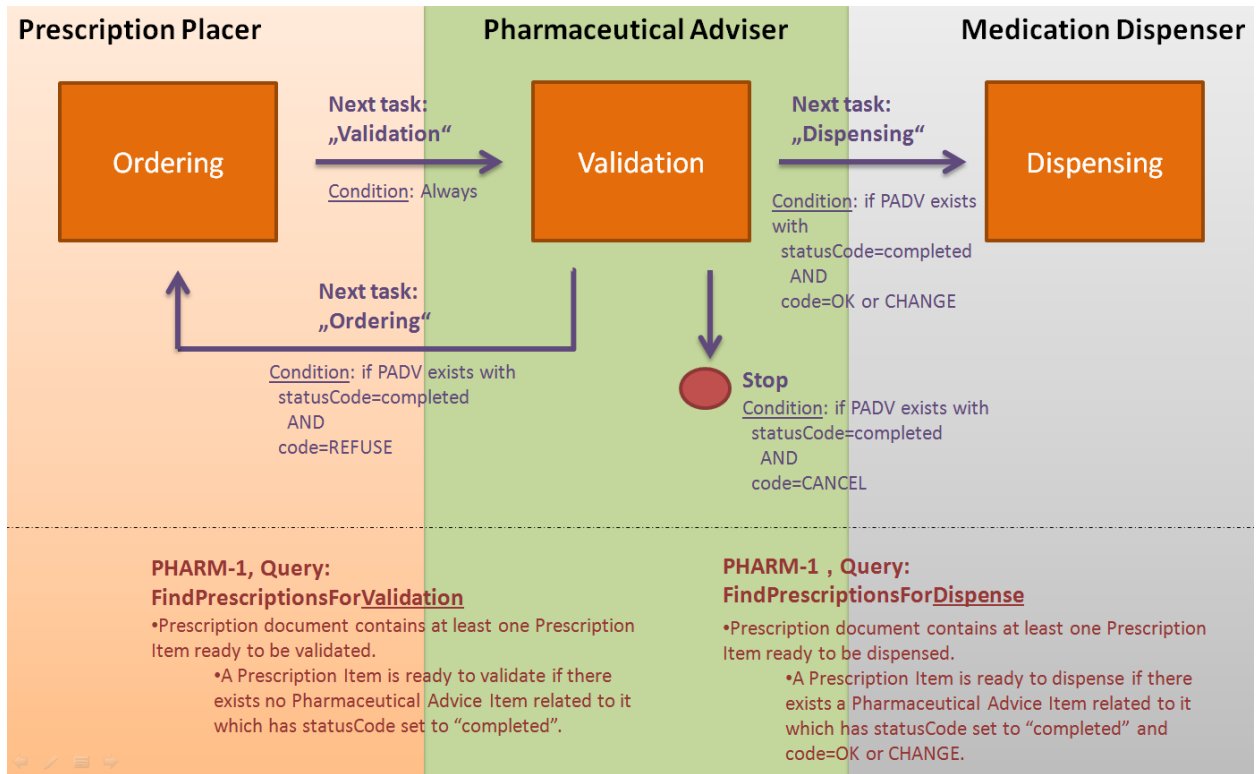
XSDDocumentEntry Attribute	Definition
typeCode	<p>For the Workflow Document which tracks the CMPD process the code for the typeCode shall be:</p> <p>Scenario 1: urn:ihe:pharm:cmpdwd1:2011</p> <p>Scenario 2: urn:ihe:pharm:cmpdwd2:2011</p> <p>Note: see chapter 4.1.3 for a description of the two workflow scenarios.</p> <p>This code is the same code that shall be use in the element workflowDefinitionReference inside the Workflow Document</p>
eventCodeList	<p>Rule 1:</p> <p>A CMPD workflow shall be created with code OPEN and shall remain in this status until it is set to CLOSE.</p> <p>Rule 2:</p> <p>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 cancellation, etc.).</p> <p>See ITI TF-3: 5.4.5.7 for a general description of this attribute.</p>
serviceStartTime	<p>It is the time at which work began on the earliest task for this workflow.</p>
serviceStopTime	<p>It is the time at which the status of the overall Workflow is changed from OPEN to CLOSE.</p> <p>It shall be empty when the workflow is still in OPEN state.</p>

4.1.3 Workflow Task Definition

1455 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:

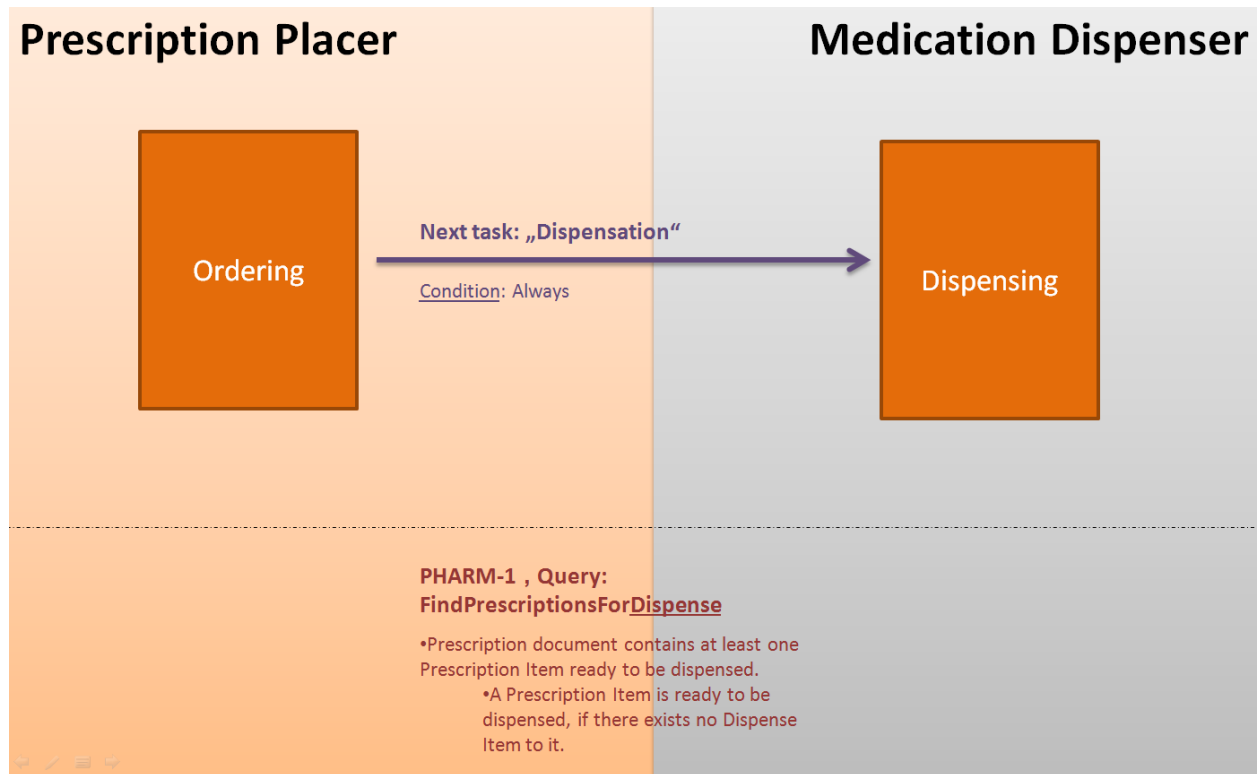
Scenario 1: Including validation step



1460

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

Scenario 2: Not including validation step



1465

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.

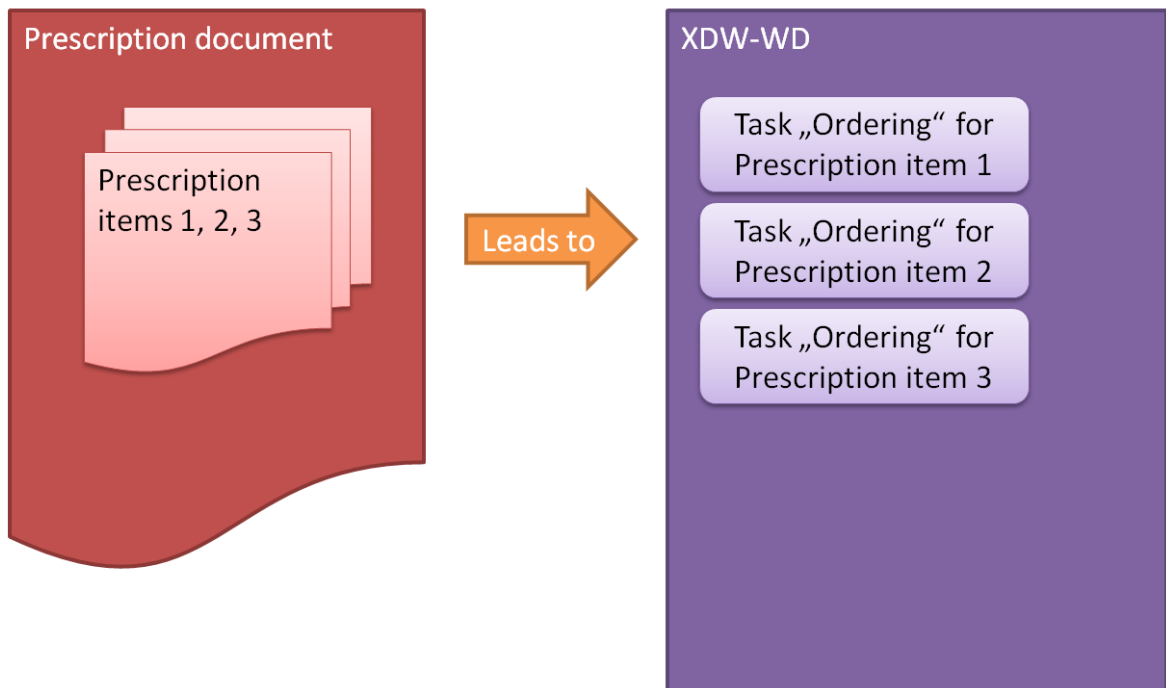
1470 The following workflow tasks are defined and specified in detail in the following chapters:

- Ordering
- Validation
- Dispensing

1475 **Important note:** When referencing input and output documents in tasks, both document uniqueId and homeCommunityId of the document shall be used.

4.1.1.1 Task: Ordering

1480 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 a single Prescription item (within a Prescription), hence separate tasks have to be created for each Prescription item of the Prescription.



1485

Figure 4.1.3.1-1 Prescription containing Prescription items leading to workflow tasks “Ordering”

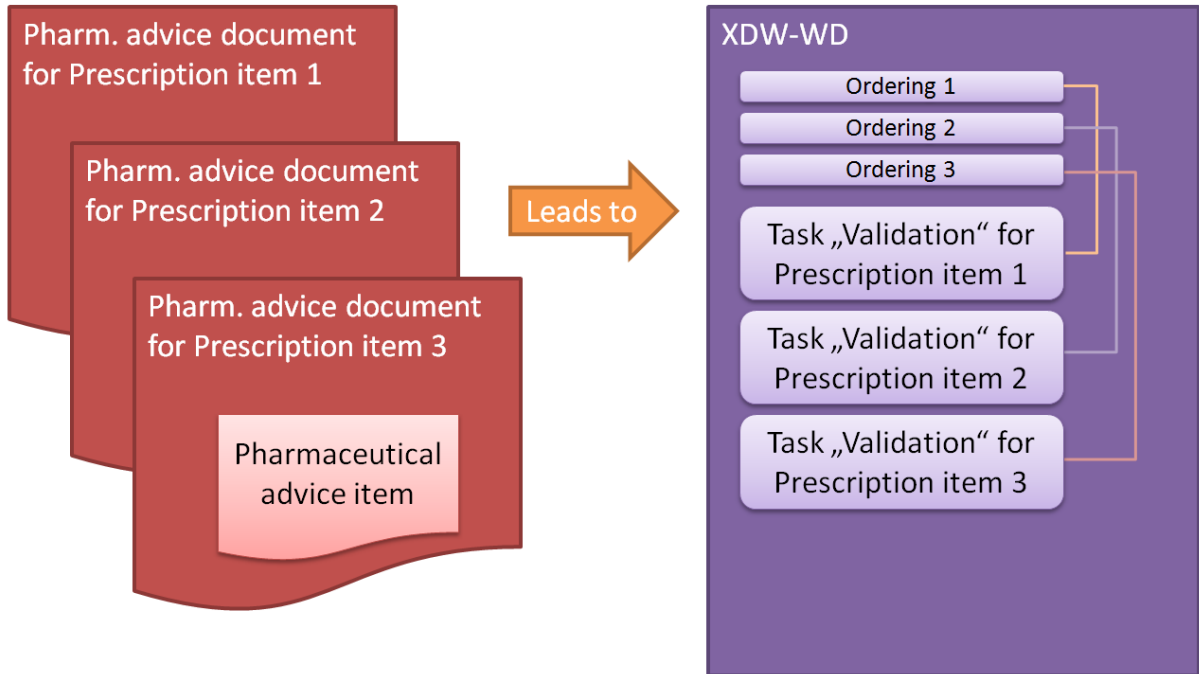
Table 4.1.3.1-1 Ordering Task Rules

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).


```
1500 <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:createdTime>2006-05-04T18:13:51.0Z</ns2:createdTime>
      <ns2:lastModifiedTime>2006-05-04T18:13:51.0Z</ns2:lastModifiedTime>
      <ns2:renderingMethodExists>>false</ns2:renderingMethodExists>
1505 </ns2:taskDetails>
      <!--
      The description element shall contain the PrescriptionItemId, this task is
      referring to (substanceAdministration/id element of the prescription item)
1510 -->
      <ns2:description>4711^^^&1.2.3.4.5.6.7.8.9&ISO</ns2:description>
      <!-- input documents -->
      <ns2:input>
1515   <ns2:part name="document">
      <!-- Document usefull 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>
1520 </ns2:input>
      <!-- output documents -->
      <ns2:output>
      <ns2:part name="Prescription Document">
1525   <!-- 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>
1530 </ns3:taskData>
      :
```

1535 **4.1.1.2 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.



1540

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). <u>Format compliant to the HL7 v2 CX datatype:</u>

Task attributes	Rules for the task “Validation”
	<p><i>Variant 1: Only id/@root is given</i> \$desc = substanceAdministration/id/@root</p> <p><i>Variant 2: id/@root and id/@extension is given</i> \$desc = concat(substanceAdministration/id/@extension, "^^^&", substanceAdministration/id/@root, "&ISO")</p>
Task dependencies	<p>Workflow scenario 1 Ancestors: Ordering Successors: Dispensing, Ordering</p> <p>Workflow scenario 2 Not applicable</p>
Status allowed	<p>IN PROGRESS A Validation task shall be set to IN PROGRESS, when the outcome of the task is a <i>preliminary</i> validation result. A Pharmaceutical Advice document is preliminary, if the element statusCode is set to <i>active</i>.</p> <p>COMPLETED A Validation task shall be set to COMPLETED, when the outcome of the task is a <i>final</i> validation result. A Pharmaceutical Advice document is final, if the element statusCode is set to <i>completed</i>.</p> <p>Note: See Pharmacy Pharmaceutical Advice (PADV) profile, Vol 2, chapter “6.3.4.3.3.6 Status Code”.</p>
Status transactions	All the states of the task (in progress, completed) are performed in sequence
input	<ul style="list-style-type: none"> • Required <ul style="list-style-type: none"> ○ 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. ○ The ancestor task shall be referenced. • Optional <ul style="list-style-type: none"> ○ All additional documents useful to the understand the reason for the outcome of the Pharmaceutical advice may be referenced.
output	<ul style="list-style-type: none"> • Required <ul style="list-style-type: none"> ○ The Pharmaceutical advice document produced shall be referenced.

Task attributes	Rules for the task "Validation"
owner	Same Physician or organization that creates the Pharmaceutical advice document
changes of task owner	Yes 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

1545 Example XML for this XDW task:

```

:
<ns3:taskData>
1550   <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>
1555   <ns2:createdTime>2006-05-04T18:13:51.OZ</ns2:createdTime>
      <ns2:lastModifiedTime>2006-05-04T18:13:51.OZ</ns2:lastModifiedTime>
      <ns2:renderingMethodExists>>false</ns2:renderingMethodExists>
      </ns2:taskDetails>
1560   <!--
      The description element shall contain the PrescriptionItemId, this task is
      referring to (substanceAdministration/id element of the prescription item)
      -->
      <ns2:description>4711^^^&1.2.3.4.5.6.7.8.9&ISO</ns2:description>
1565   <!-- 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>
1575   <!-- output documents -->
      <ns2:output>
        <ns2:part name="Pharmaceutical Advice Document">
          <!-- Pharmaceutical advice document according to PADV profile -->

```

1585

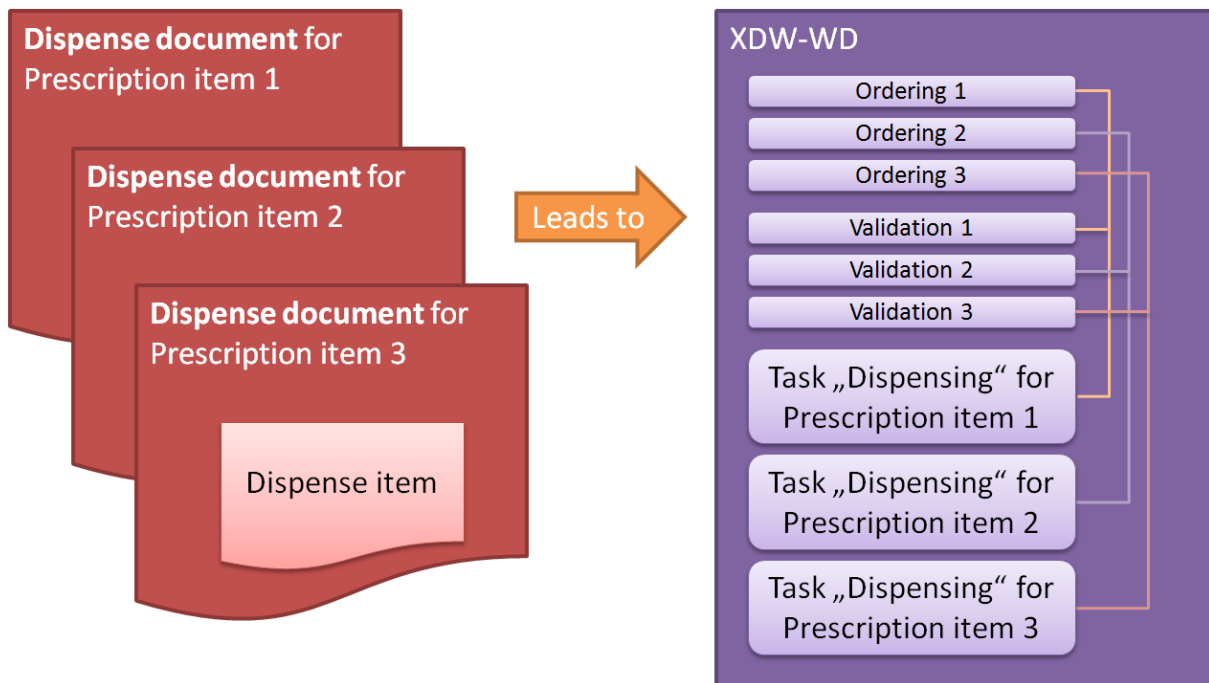
```
<!-- uid: the document uniqueId, home: the homeCommunityId -->
<reference uid="urn:oid:1.2.3.4.4.5" home="urn:oid:1.2.3"/>
</ns2:part>
</ns2:output>

</ns3:taskData>
:
```

1590

4.1.1.3 Task: Dispensing

1595 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 dispensed.



1600

Figure 4.1.3.3-1: Prescription items leading to workflow tasks “Dispensing”

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). <u>Format compliant to the HL7 v2 CX datatype:</u>

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

1605 Example XML for this XDW task:

```

:
<ns3:taskData>
1610   <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>
1615   <ns2:createdTime>2006-05-04T18:13:51.OZ</ns2:createdTime>
      <ns2:lastModifiedTime>2006-05-04T18:13:51.OZ</ns2:lastModifiedTime>
      <ns2:renderingMethodExists>>false</ns2:renderingMethodExists>
      </ns2:taskDetails>
1620   <!--
      The description element shall contain the PrescriptionItemId, this task is
      referring to (substanceAdministration/id element of the prescription item)
      -->
      <ns2:description>4711^^^&1.2.3.4.5.6.7.8.9&ISO</ns2:description>
1625   <!-- input documents -->
      <ns2:input>
        <ns2:part name="Prescription Document">
          <!-- Prescription document according to PRE profile -->
1630          <!-- 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 -->
1635            <!-- 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 -->
1640              <reference taskId="urn:oid:2.2.2.2.2"/>
              </ns2:part>
            </ns2:input>

```

1645

```
<!-- 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>
:
```

1650

1655

Glossary

Add the following terms to the Glossary:

1660 No new terms