

Integrating the Healthcare Enterprise



**IHE Quality, Research and Public Health
(QRPH)**

Technical Framework Supplement

**Retrieve Process for Execution
(RPE)**

Trial Implementation

Date: August 30, 2010
Author: Mark Arratoon
Email: qrph@ihe.net

Foreword

This is a supplement to the forthcoming IHE Quality, Research, and Public Health 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 August 30, 2010 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 Quality, Research, and Public Health Technical Framework. Comments are invited and may be submitted on the IHE forums at <http://forums.rsna.org/forumdisplay.php?f=371> or by email to qrph@ihe.net.

This supplement describes changes to the existing technical framework supplement documents 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 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>
--

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

Information about IHE Quality, Research, and Public Health 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>

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

Contents

INTRODUCTION	5
OPEN ISSUES AND QUESTIONS.....	5
CLOSED ISSUES.....	5
VOLUME 1 – INTEGRATION PROFILES	6
1.1.5 Copyright Permissions.....	6
2.5 DEPENDENCIES OF THE RPE INTEGRATION PROFILE	6
2.7 HISTORY OF ANNUAL CHANGES	6
X RETRIEVE PROCESS FOR EXECUTION INTEGRATION PROFILE.....	7
X.1 PURPOSE AND SCOPE.....	7
X.2 USE CASES.....	7
X.2.1 Clinical Trials	7
Before RPE.....	7
After RPE	8
X.2.2 Public Health Case Reporting	10
X.2.3 Quality Reporting	14
X.3 ACTORS / TRANSACTIONS	16
X.3.1 RPE Actors	18
X.3.1.1 ProcessDefinitionManager	18
X.3.1.2 ProcessStateManager	18
X.3.1.3 ProcessActivityExecutor	18
X.3.2 RPE Transactions.....	18
X.3.2.1 RetrieveProcessDefinitions	18
X.3.2.2 Subscribe	19
X.3.2.3 ReceiveProcessDefinitions	19
X.3.2.4 InitiateProcess.....	19
X.3.2.6 RetrieveActivities	19
X.3.2.7 UpdateActivity	19
X.3.2.8 ReceiveProcessStateAlert.....	19
X.3.2.9 InitiateActivity	19
X.4 OPTIONS	20
X.5 GROUPINGS	20
X.6 SECURITY CONSIDERATIONS	20
X.6.1 Recommendations	20
X.7 REQUIREMENTS OF ACTORS	21
X.8 CONTENT MODULES.....	22
X.9 PROCESS FLOW	22
APPENDIX A: ACTOR SUMMARY DEFINITIONS	26
APPENDIX B: TRANSACTION SUMMARY DEFINITIONS	26
GLOSSARY	28
VOLUME 2 - TRANSACTIONS	29
3. IHE TRANSACTIONS	30

3.Y1 RETRIEVEPROCESSDEFINITIONS.....	30
3.Y1.1 Scope.....	30
3.Y1.2 Use Case Roles.....	30
3.Y1.3 Referenced Standards.....	30
3.Y1.4 Interaction Diagram.....	31
3.Y1.4.1 RetrieveProcessDefinitions Message.....	31
3.Y1.4.2 RetrieveProcessDefinitionsResponse Message.....	33
3.Y1.5 Security Considerations.....	34
3.Y1.6 Protocol Requirements.....	34
3.Y1.6.1 Sample SOAP Messages.....	35
3.Y2 SUBSCRIBE.....	36
3.Y2.1 Scope.....	36
3.Y2.2 Use Case Roles.....	37
3.Y2.3 Referenced Standards.....	37
3.Y2.4 Interaction Diagram.....	38
3.Y2.4.1 Subscribe Message.....	38
3.Y2.4.2 SubscribeResponse Message.....	39
3.Y2.5 Security Considerations.....	40
3.Y2.6 Protocol Requirements.....	40
3.Y2.6.1 Sample SOAP Messages.....	41
3.Y3 RECEIVEPROCESSDEFINITIONS.....	42
3.Y3.1 Scope.....	42
3.Y3.2 Use Case Roles.....	42
3.Y3.3 Referenced Standards.....	42
3.Y3.4 Interaction Diagram.....	43
3.Y3.4.1 ReceiveProcessDefinitions Message.....	43
3.Y3.4.2 ReceiveProcessDefinitionsResponse Message.....	44
3.Y3.5 Security Considerations.....	45
3.Y3.6 Protocol Requirements.....	45
3.Y3.6.1 Sample SOAP Messages.....	45
3.Y4 INITIATEPROCESS.....	47
3.Y4.1 Scope.....	47
3.Y4.2 Use Case Roles.....	48
3.Y4.3 Referenced Standards.....	48
3.Y4.4 Interaction Diagram.....	48
3.Y4.4.1 InitiateProcess Message.....	49
3.Y4.4.2 InitiateProcessResponse Message.....	50
3.Y4.5 Security Considerations.....	50
3.Y4.6 Protocol Requirements.....	50
3.Y4.6.1 Sample SOAP Messages.....	51
3.Y5 RETRIEVEACTIVITIES.....	52
3.Y5.1 Scope.....	52
3.Y5.2 Use Case Roles.....	52
3.Y5.3 Referenced Standards.....	53
3.Y5.4 Interaction Diagram.....	54
3.Y5.4.1 RetrieveActivities Message.....	54
3.Y5.4.2 RetrieveActivitiesResponse Message.....	55
3.Y5.5 Security Considerations.....	55
3.Y5.6 Protocol Requirements.....	56
3.Y5.6.1 Sample SOAP Messages.....	56
3.Y6 UPDATEACTIVITY.....	58

3.Y6.1 Scope	58
3.Y6.2 Use Case Roles.....	58
3.Y6.3 Referenced Standards	59
3.Y6.4 Interaction Diagram	59
3.Y6.4.1 UpdateActivity Message.....	60
3.Y6.4.1 UpdateActivityResponse Message	61
3.Y6.5 Security Considerations	61
3.Y6.6 Protocol Requirements	61
3.Y6.6.1 Sample SOAP Messages	62
3.Y7 RECEIVEPROCESSSTATEALERT	63
3.Y7.1 Scope	63
3.Y7.2 Use Case Roles.....	64
3.Y7.3 Referenced Standards	64
3.Y7.4 Interaction Diagram	64
3.Y7.4.1 ReceiveProcessStateAlert Message.....	65
3.Y7.4.2 ReceiveProcessStateAlertResponse Message	66
3.Y7.5 Security Considerations	67
3.Y7.6 Protocol Requirements	67
3.Y7.6.1 Sample SOAP Messages	67
3.Y8 INITIATEACTIVITY	69
3.Y8.1 Scope	69
3.Y8.2 Use Case Roles.....	69
3.Y8.3 Referenced Standards	69
3.Y8.4 Interaction Diagram	70
3.Y8.4.1 InitiateActivity Message	70
3.Y8.4.2 InitiateActivityResponse Message.....	71
3.Y8.5 Security Considerations	72
3.Y8.6 Protocol Requirements	72
3.Y8.6.1 Sample SOAP Messages	73
APPENDIX A – RPE MOTIVATIONS.....	75
APPENDIX B – BACKWARDS COMPATIBILITY	79
APPENDIX C – INFORMATIVE EXAMPLE WITH BPEL/WS-HUMANTASK	80

Introduction

Retrieve Process for Execution (RPE) is a proposed IHE standard for *collaborative workflow* or *collaborative process management* involving three main actors: 1/ the manager of process definitions, 2/ the manager of runtime processes and 3/ the performer of process activities.

RPE started as Retrieve *Protocol* for Execution Profile a proposed automated mechanism for an Electronic Health Record (EHR) to retrieve a complex set of clinical research instructions (a study protocol) from a research sponsor's Electronic Data Capture (EDC) system and to execute these instructions within the EHR. Interest has since grown to expand RPE to support processes from other domains involving collaborative workflow like quality, public health and patient care support. RPE now does this consistent with accepted IT standards for process management.

Open Issues and Questions

No open issues at this time.

Closed Issues

No closed issues at this time.

Volume 1 – Integration Profiles

Add the following to section 1.1.5

1.1.5 Copyright Permissions

To be completed

Add the following to section 2.5

2.5 Dependencies of the RPE Integration Profile

Retrieve Process for Execution (RPE) is not strictly dependent on any IHE profile but, for performance of activities, the Retrieve Form for Data-capture (RFD) and Redaction Services profiles can be used very synergistically for form-based submission of appropriately redacted EHR data.

Add the following to section 2.7

2.7 History of Annual Changes

To be completed

Add Section X to the QRPH Technical Framework

X Retrieve Process for Execution Integration Profile

Retrieve Process for Execution (RPE) is a proposed IHE standard for *collaborative workflow* or *collaborative process management* involving three main actors: 1/ the manager of process definitions, 2/ the manager of runtime processes and 3/ the performer of process activities.

X.1 Purpose and Scope

The Retrieve Process for Execution (RPE) profile started as Retrieve *Protocol* for Execution, a proposed automated mechanism for an Electronic Health Record (EHR) to retrieve a complex set of clinical research instructions (a study protocol) from a research sponsor's Electronic Data Capture (EDC) system and to execute these instructions within the EHR. Interest has since grown to expand RPE to support processes from other domains involving collaborative workflow like quality, public health and patient care support. This new version of RPE now does this consistent with accepted IT standards for process management.

X.2 Use Cases

X.2.1 Clinical Trials

In the clinical research uses case below, we describe the before and after effects of implementing the Retrieve Process for Execution (RPE) profile for an investigational new drug clinical trial scenario.

- Scenario
- The setting for the clinical trial use case is a physicians' practice where patient care is delivered side-by-side with clinical research.
- The site, Holbin Medical Group, is a multi-site physician practice, employing over 100 physicians in a variety of specialties.
- Holbin's CEO encourages the physicians to participate as site investigators for pharmaceutical-sponsored clinical trials.

Before RPE

Preconditions

1. A Clinical Research Protocol is defined by a clinical trials expert at PharmaGen, a biopharma research sponsor
2. Holbin provides support for clinical research activities in the form of a Research Department of twelve dedicated study coordinators, mostly RNs, along with clerical and data-entry support personnel

3. Holbin Medical Group uses an Electronic Health Record (EHR) and a number of sponsor-provided Electronic Data Capture (EDC) systems for documenting clinical trial activities

Use Case

1. Clinical Research Site's Involvement:
 - i. Holbin's involvement in a clinical study begins when the Research Department receives a request for proposal (RFP) from PharmaGen
 - ii. A Study Coordinator, Patricia Zone, RN, evaluates the RFP for business viability and clinical appropriateness, provides the requested documentation back to the sponsor, and agrees to participate
 - iii. After being approved as a site for the PharmaGen #1234 trial, the site Holbin Medical Group provides the required regulatory documentation to the sponsor
2. Following trial set up, Patricia contacts Corey Jones, a patient at Holbin, about participating in the trial, and Corey agrees to participate as a subject. A number of tasks deal with this individual patient:
 - i. Obtain proper consent and other documentation from study candidate Corey Jones.
 - ii. Register Corey in the EHR as a candidate in trial #1234, using the EHR's patient index.
 - iii. She also registers Corey as a candidate in the EDC system.
 - iv. She schedules Corey's study screening visits using the EHR scheduling module, and flags the visits as pertaining to the trial #1234.
 - v. Screening results are then examined to confirm continued enrollment in the trial.
3. After screening, Patricia then obtains and schedules the next set of study activities corresponding to the path (arm) thru the trial assigned for Corey.

Postconditions

1. Holbin Medical Group uses an EHR and the EDC system to document the PharmaGen #1234 trial activities.

After RPE

Preconditions

1. A Clinical Research Protocol is defined by the clinical trials expert at PharmaGen using a study design tool.
2. The resulting study design definition document is stored and made available for access by a ProcessDefinitionManager (implemented by an EDC or other system).
3. A ProcessStateManager (implemented by an EDC or other system) is available to deploy, run and manage the execution of the clinical trial process.
4. Holbin's EHR (or other system) can implement the ProcessActivityExecutor role.

Use Case

1. Clinical Research Site's Involvement:
 - i. The ProcessActivityExecutor uses the RetrieveProcessDefinitions transaction to obtain a list of protocols from the ProcessDefinitionManager.
2. Screening:
 - i. ProcessActivityExecutor uses the InitiateProcess transaction to notify the ProcessStateManager that the site wishes to enter a patient into the study.
 - ii. ProcessActivityExecutor uses the RetrieveActivities transaction to obtain from the ProcessStateManager the set of screening activities.
 - iii. ProcessActivityExecutor uses the UpdateActivity transaction to send the ProcessStateManager the screening results. (Note if activities have associated RFD forms then UpdateActivity may be replaced by RFD form retrieval and submission.)
 - iv. ProcessStateManager uses the ReceiveProcessStateAlert transaction to notify the ProcessActivityExecutor that the patient passed screening and has been enrolled in the trial.
3. Treatment:
 - i. ProcessActivityExecutor uses the RetrieveActivities transaction again to obtain from the ProcessStateManager the next set of study activities for this patient.
 - ii. ProcessActivityExecutor uses the UpdateActivity transaction to send the ProcessStateManager updates of activities as they are performed. (Note if activities have associated RFD forms then UpdateActivity may be replaced by RFD form retrieval and submission.)
 - iii. ProcessActivityExecutor can at any time use the ReceiveProcessStateAlert transaction to notify the ProcessStateManager that the patient has withdrawn from the trial.

- iv. ProcessStateManager can at any time use the ReceiveProcessStateAlert transaction to notify the ProcessActivityExecutor that the trial has been placed on hold.

Postconditions

1. Holbin uses the ProcessActivityExecutor audit trail of study activity transactions to document its entire PharmaGen #1234 trial participation.
2. PharmaGen uses the ProcessStateManager audit trail of the study process to document the entire #1234 trial for all sites and patients.

X.2.2 Public Health Case Reporting

From a draft IHE proposal for Public Health Case Reporting Workflow:

Existing problem: In order to execute essential public health functions, local and state public health agencies rely heavily on the reporting of public health cases from providers and other entities at the point of care. The absence of a comprehensive public health case reporting workflow limits providers and other entities to send a public health case report that contains timely and complete public health information in a case investigation. Regardless if a provider uses a paper or electronic form, information as to how to report and what ancillary information to include is uncertain.

How problem could be solved: Presently, gaps exist in describing the protocols involved in sending a public health case report. CDC’s National Center for Public Health Informatics (NCPHI) and the Council of State and Territorial Epidemiologists (CSTE) host the Case Report Standardization Workgroup (CRSWG), which intends to describe the relationships between ongoing public health case report activities aimed at improving the timeliness and completeness of reporting across the spectrum of public health.

Why IHE: Ultimately, the Public Health Case Reporting Workflow will help develop a coordinated approach for developing public health profiles at IHE and could inform the development of the Technical Framework for the Quality, Research and Public Health (QRPH) Technical Committee.

...

The use case to describe public health case reporting has already been described by the American Health Information Community (AHIC) and entitled “Public Health Detailed Used Case”, 2008. URL:

http://healthit.hhs.gov/portal/server.pt/gateway/PTARGS_0_10741_848112_0_0_18/PHCRDetailed.pdf

The use case is also supported by HITSP Interoperability Specification [IS] 11, 2008. URL: <http://wiki.hitsp.org/docs/IS11/IS11-2.html>

Based on the AHIC use case, the extension of [RPE] (and [RFD]) may serve as a conduit to further describe the public health case report workflow along with the systems, tools, and resources used to enable case reporting.

Two key use case scenarios described in HITSP IS11 (referenced above) are shown below.

Note: the figures shown are within the IS11 document and not part of this specification:

Figure 6.4-1 Public Health Case Reporting (PHCR) High Level Sequence Diagram – Part 1 – Provider Perspective (Including Laboratory) – Scenario 1: Reporting from Electronic Health Records

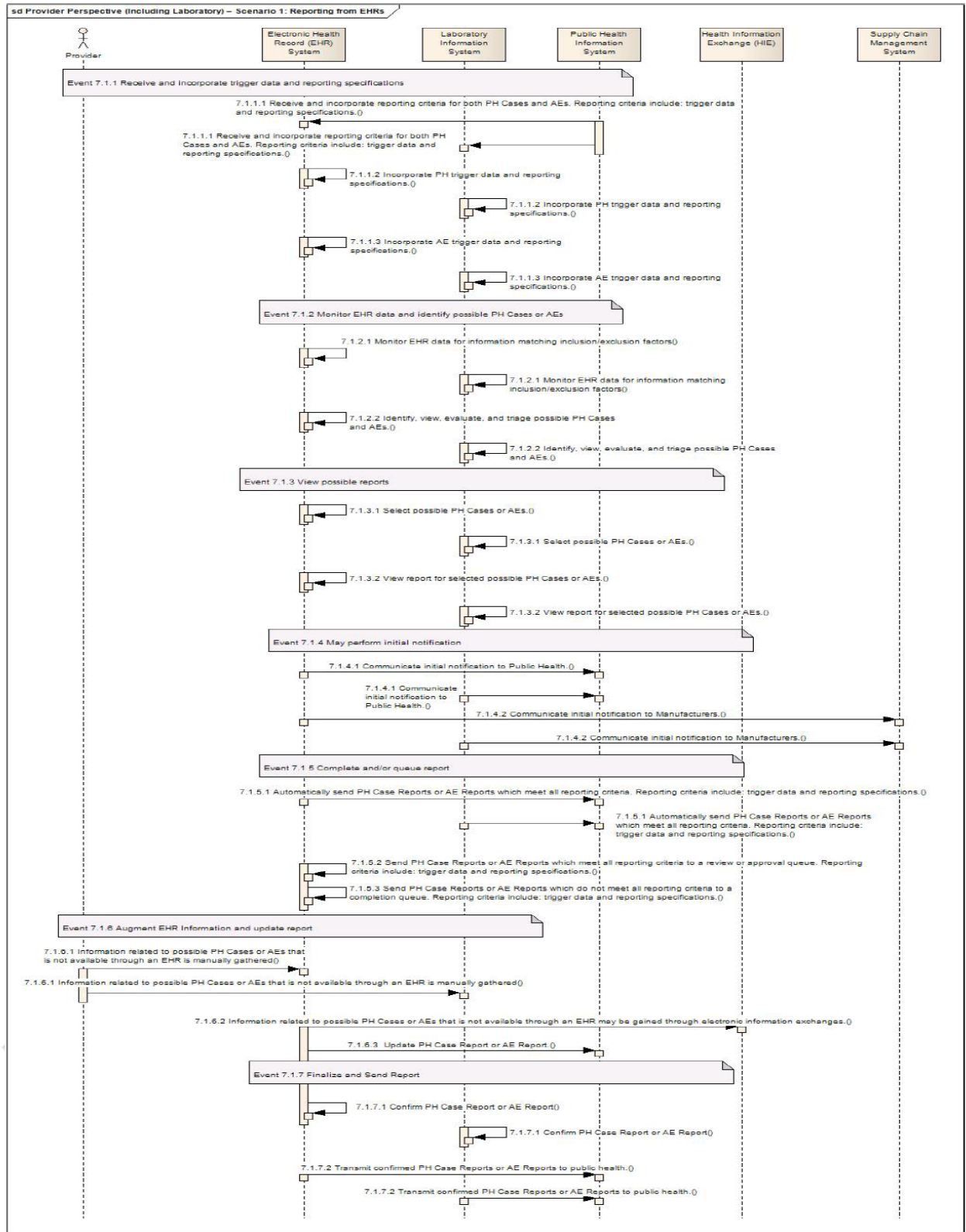
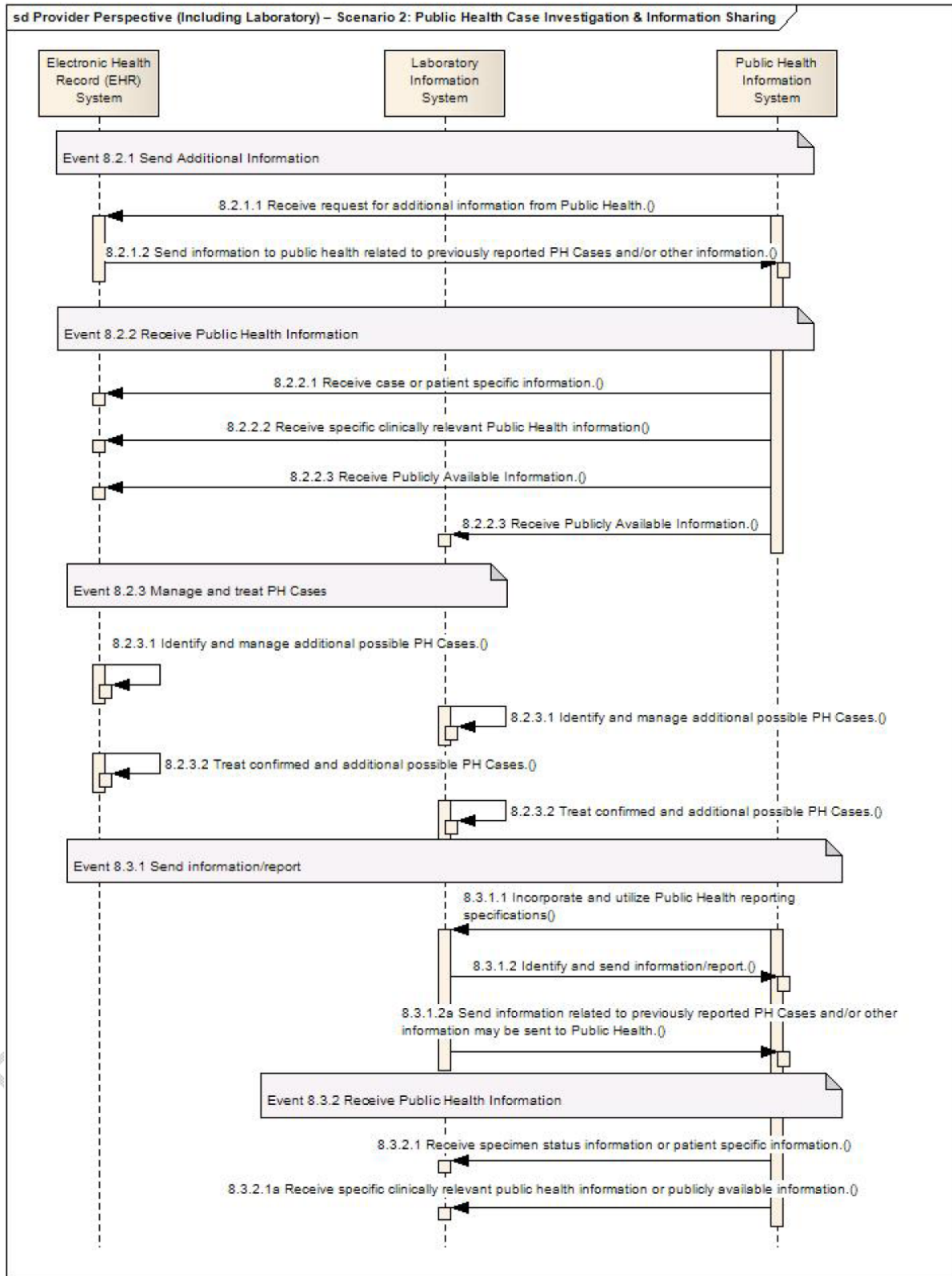


Figure 6.4-2 Public Health Case Reporting (PHCR) High Level Sequence Diagram – Part 2 – Provider Perspective (Including Laboratory) – Scenario 2: Alert Functionality

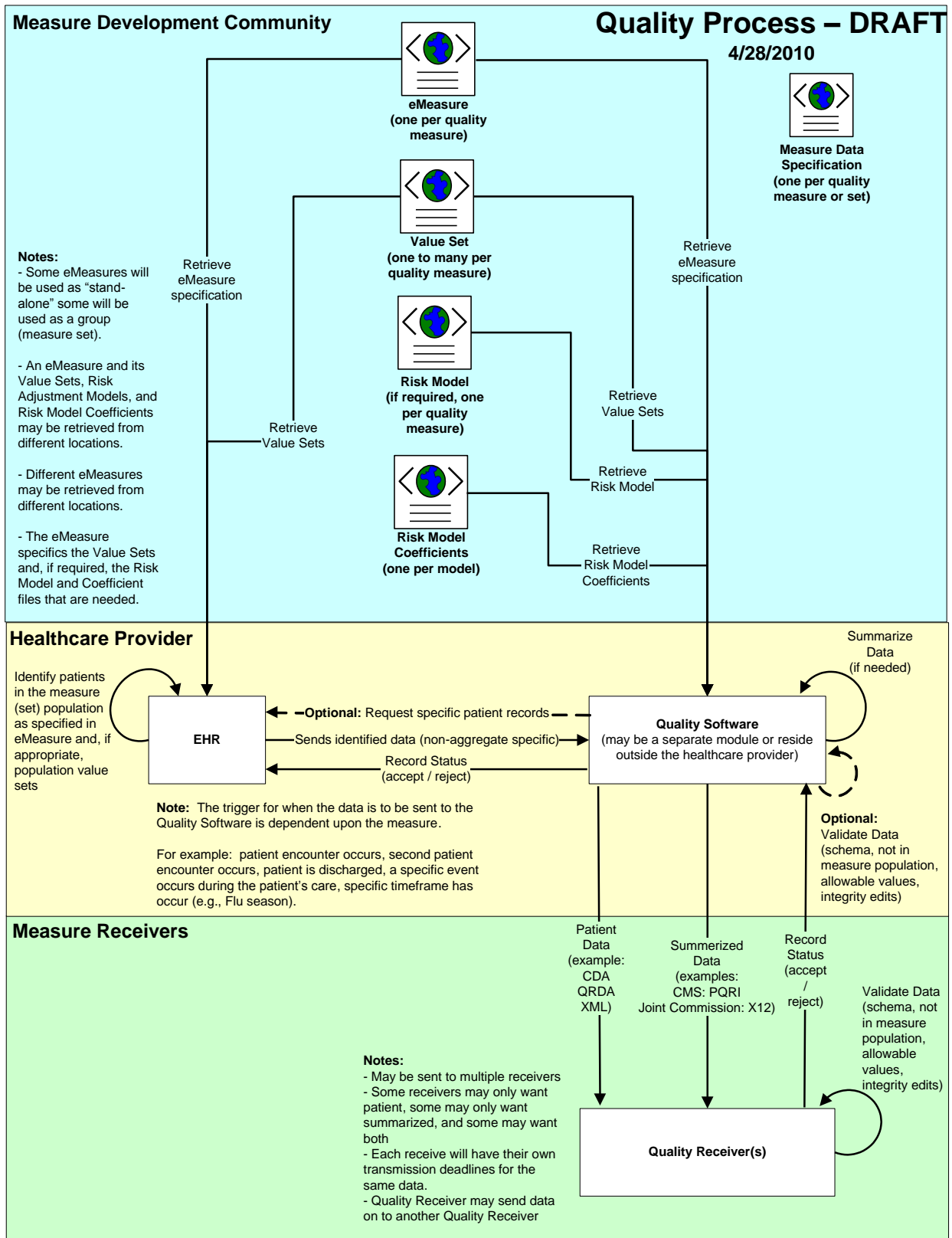


Once the processes and activities that constitute Public Health Case Reporting Workflow have been well-defined and expressed in a process definition format, RPE could be applied to manage the above transactions. The table below shows the potential applicability of RPE (and RFD) actors and transactions to the above use cases:

RPE/RFD Actor & Transactions	Equivalent PH Actor	Description (from IS11)
RPE		
Process Activity Executor	EHR	
ReceiveProcessStateAlert ReceiveProcessDefinitions		7.1.1.1 Receive and incorporate reporting criteria for both PH Cases and AEs. (PH <i>pushes</i> criteria to EHR.) 8.2.2.1 Receive case or patient specific information. 8.2.1.1 Receive request for additional information from PH. 8.2.2.2 Receive specific clinically relevant PH information. 8.2.2.3 Receive publically available information.
Process Definition Manager	Public Health	
RetrieveProcessDefinitions Subscribe		7.1.1.1 Receive and incorporate reporting criteria for both PH Cases and AEs. 7.1.1.1.B <i>Retrieve</i> and incorporate reporting criteria for both PH Cases and AEs. (EHR <i>pulls</i> criteria from PH. Note: this is not yet a part of IS11 but a suggested addition.)
Process State Manager	Public Health	
InitiateProcess RetrieveActivities UpdateActivity ReceiveProcessStateAlert ReceiveProcessDefinitions		7.1.1.1.B <i>Retrieve</i> and incorporate reporting criteria for both PH Cases and AEs. (EHR <i>pulls</i> criteria from PH. Note: this is not yet a part of IS11 but a suggested addition.) 7.1.4.1 Communicate initial notification to PH. 7.1.5.1 Automatically send PH Case Reports or AE Reports which meet all reporting criteria. 7.1.6.3 Update PH Case Report or AE Report. 7.1.7.2 Transmit confirmed PH Case Report or AE Report. 8.2.1.2 Send information to PH related to previously reported PH Cases and/or other information.
RFD		
Form Filler	EHR	
Form Manager	Public Health	
RetrieveForm		7.1.1.1 Receive and incorporate reporting criteria for both PH Cases and AEs.
Form Receiver	Public Health	
SubmitForm		7.1.4.1 Communicate initial notification to PH. 7.1.5.1 Automatically send PH Case Reports or AE Reports which meet all reporting criteria. 7.1.6.3 Update PH Case Report or AE Report. 7.1.7.2 Transmit confirmed PH Case Report or AE Report. 8.2.1.2 Send information to PH related to previously reported PH Cases and/or other information.

X.2.3 Quality Reporting

Below is a draft representation of the overall quality reporting process:



The table below shows the potential applicability of RPE (and RFD) actors and transactions to the above process:

RPE/RFD Actor & Transactions	Equivalent Quality Actor	Description (see above Quality Process flow)
RPE		
Process Activity Executor	Quality Process Executor	Healthcare Provider
ReceiveProcessStateAlert ReceiveProcessDefinitions		Record Status (accept/reject). Optional: Request specific patient records. Receive quality process/eMeasure specifications.
Process Definition Manager	Quality Process Repository	A source of quality reporting process definitions/eMeasure specifications within Measure Development Community.
RetrieveProcessDefinitions Subscribe		Retrieve quality process definitions/eMeasure specifications. Subscribe to quality process definitions/eMeasure specifications.
Process State Manager	Quality Process Manager	Quality Software: manages the runtime quality measure reporting process. The latter includes an eMeasure specification(s) as well as reporting trigger/activity requirements.
InitiateProcess UpdateActivity ReceiveProcessDefinitions		Initiate quality process reporting with Quality Software. Send identified data to Quality Software. Receive quality process/eMeasure specifications.
RFD		
Form Filler	Quality Process Executor	Healthcare Provider
Form Manager	Quality Process Manager	(see above)
RetrieveForm		Retrieve eMeasure form for sending identified data.
Form Receiver	Quality Process Manager	(see above)
SubmitForm		Send identified data to Quality Software.

X.3 Actors / Transactions

Figure X.3-1 shows the actors directly involved in the Retrieve Process for Execution Integration Profile and the relevant transactions between them. Other actors that may be indirectly involved due to their participation in other profiles are not necessarily shown.

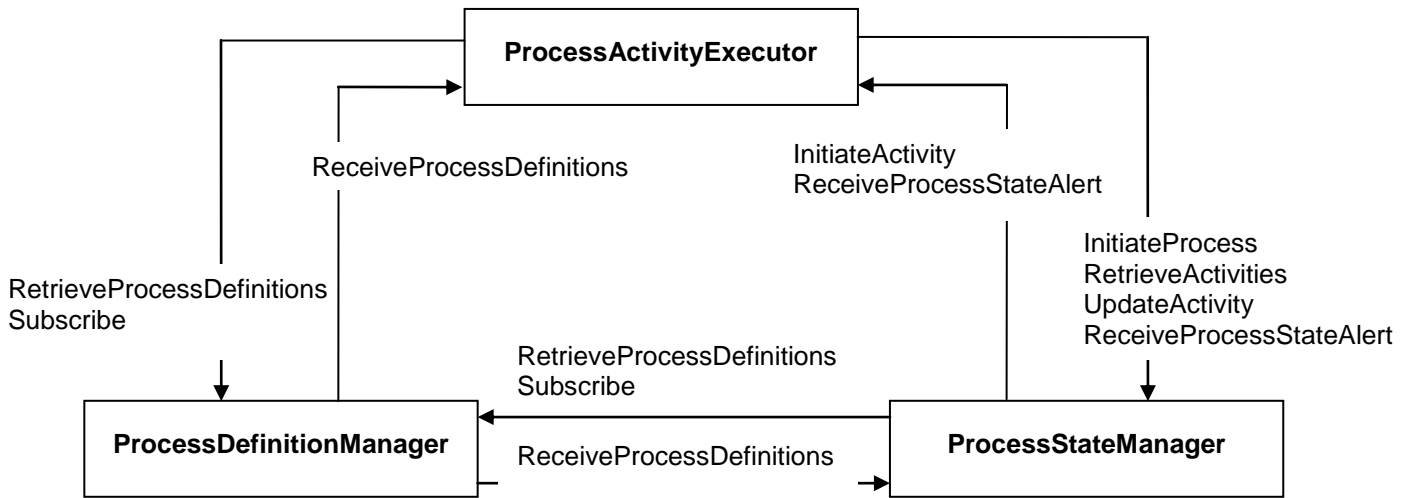


Figure X.3-1 RPE Actor Diagram

Table X.3-1 lists the transactions for each actor directly involved in the RPE 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 I, Section X.2.

Table X.3-1: RPE Actors and Transactions

Actors	Transactions	Optionality	Section in Vol. 2
ProcessDefinitionManager	RetrieveProcessDefinitions	R	QRPH TF-2:3.Y1
	Subscribe	R	QRPH TF-2:3.Y2
	ReceiveProcessDefinitions	R	QRPH TF-2:3.Y3
ProcessStateManager	RetrieveProcessDefinitions	O	QRPH TF-2:3.Y1
	Subscribe	O	QRPH TF-2:3.Y2
	ReceiveProcessDefinitions	O	QRPH TF-2:3.Y3
	InitiateProcess	R	QRPH TF-2:3.Y4
	RetrieveActivities	R	QRPH TF-2:3.Y5
	UpdateActivity	O	QRPH TF-2:3.Y6
	ReceiveProcessStateAlert	R	QRPH TF-2:3.Y7
ProcessActivityExecutor	InitiateActivity	O	QRPH TF-2:3.Y8
	RetrieveProcessDefinitions	R	QRPH TF-2:3.Y1
	Subscribe	O	QRPH TF-2:3.Y2
	ReceiveProcessDefinitions	O	QRPH TF-2:3.Y3
	InitiateProcess	R	QRPH TF-2:3.Y4
	UpdateActivity	O	QRPH TF-2:3.Y6

Actors	Transactions	Optionality	Section in Vol. 2
	ReceiveProcessStateAlert	R	QRPH TF-2:3.Y7
	InitiateActivity	O	QRPH TF-2:3.Y8

X.3.1 RPE Actors

X.3.1.1 ProcessDefinitionManager

The ProcessDefinitionManager manages access to a repository of process definitions allowing for search and retrieval.

An example would be a research sponsor providing access to the clinical research protocols it has developed. Another would be a provider of evidence-based clinical practice guidelines.

X.3.1.2 ProcessStateManager

The ProcessStateManager manages the initiation and state of runtime process instances. The ProcessStateManager typically also supports the initiation and lifecycle management of task activities associated with a process while providing the ability for task performers to retrieve and update activities.

An example would be a research sponsor conducting clinical trials in conjunction with its EHR participants. Another would be an EHR performing clinical care in accord with executable guideline-based processes.

X.3.1.3 ProcessActivityExecutor

The ProcessActivityExecutor performs activities as prescribed by a running process being managed by a ProcessStateManager. The ProcessActivityExecutor retrieves current activity or task lists, works its list, updating the ProcessStateManager on activity state until completion. This cycle is repeated until all process activities have been worked and the process itself completes.

An example would be an EHR performing activities as part of a clinical trial being managed by a research sponsor. Another example would be an EHR performing guideline-based care process activities.

X.3.2 RPE Transactions

X.3.2.1 RetrieveProcessDefinitions

RetrieveProcessDefinitions enables access to one or more process definitions specified by an identifier or other query criteria. This transaction is implemented by the ProcessDefinitionManager and used by both the:

- ProcessActivityExecutor – to examine processes it may be interested in becoming an active participant and,
- ProcessStateManager – to deploy processes it wishes to manage

X.3.2.2 Subscribe

Subscribe allows either a ProcessStateManager or ProcessActivityExecutor to optionally establish a subscription to new or amended process definitions of matching interest, as they become available from a ProcessDefinitionManager.

X.3.2.3 ReceiveProcessDefinitions

ReceiveProcessDefinitions enables either a ProcessStateManager or ProcessActivityExecutor to receive new or amended process definitions of matching interest, as defined by a previous Subscribe transaction, as they become available from a ProcessDefinitionManager.

X.3.2.4 InitiateProcess

InitiateProcess enables a ProcessActivityExecutor to initiate a new process to be managed by a ProcessStateManager, e.g., an EHR entering a new patient candidate in a clinical trial being managed by a research sponsor.

X.3.2.6 RetrieveActivities

RetrieveActivities enables a ProcessActivityExecutor to retrieve the current set of activities it needs to execute as part of a process managed by a ProcessStateManager.

X.3.2.7 UpdateActivity

UpdateActivity allows a ProcessActivityExecutor to provide an update on activity's state or data to a ProcessStateManager for a process it is a participant in.

X.3.2.8 ReceiveProcessStateAlert

ReceiveProcessStateAlert provides either the ProcessStateManager or ProcessActivityExecutor the ability to notify each other of unscheduled events that effect the state of the process, e.g., an EHR patient withdrawing from a clinical trial or, a study being placed on hold.

X.3.2.9 InitiateActivity

InitiateActivity allows the ProcessActivityExecutor to be in control of the lifecycle of activities it performs, e.g., an EHR may want to use its own task processor and interfaces to create and manage the activities it needs to perform as part of a process managed by the ProcessStateManager.

X.4 Options

No options defined.

X.5 Groupings

RPE can make use of both Retrieve Form for Data-capture (RFD) and Redaction Services in a complementary fashion to enact a powerful collaborative workflow framework.

X.6 Security Considerations

The risk analysis for RPE enumerates assets, threats, and mitigations. The complete risk data is stored and available from IHE¹.

The purpose of this risk assessment is to notify vendors of some of the risks that they are advised to consider when implementing RPE actors. For general IHE risks and threats, please see ITI TF-1: Appendix L. The vendor is also advised that many risks cannot be mitigated by the IHE profile and instead responsibility for mitigation is transferred to the vendor, and occasionally to the affinity domains, individual enterprises and implementers. In these instances, IHE fulfills its responsibility to notify affected parties through the use of the following sections.

X.6.1 Recommendations

The high impact risks include:

- mismatch between patient and patient data
- malicious system attack
- inappropriate or premature disclosure of personal health information
- patient and organization identity protection

For risk mitigation, the following profiles from the ITI-TF are referred:

- Audit Trail and Node Authentication (ATNA)
- Basic Patient Privacy Consents (BPPC)
- Cross-Enterprise User Assertion (XUA)
- Consistent Time (CT)
- Enterprise User Authentication (EUA)

¹ The risk analysis data may be found at:
[ftp://ftp.ihe.net/Quality/2010_2011_YR_4/QRPH%20Technical%20Committee/Maintenance%20work/RPE/RPE Risk Analysis 2010-05-21.xls](ftp://ftp.ihe.net/Quality/2010_2011_YR_4/QRPH%20Technical%20Committee/Maintenance%20work/RPE/RPE%20Risk%20Analysis%202010-05-21.xls)

- Document Digital Signature (DSG)

This profile includes the mitigations:

M1: Ensure additional demographics are available to verify patient identity (e.g., address, other identifiers)

M2: Identify actors, ensure secure login and access control to protected data, e.g., using XUA and grouping with EUA for protected applications.

M3: Use secured communication for any patient data e.g., as per ATNA.

M4: Audit access to PHI, e.g., as per ATNA.

M5: Obtain patient consent to release protected data, e.g., using BPPC and DSG.

M6: Use pseudonymized identifiers for organizations in transactions as supported by RPE.

These mitigations are transferred to vendors and clients:

T1: Verify demographics other than patient name, birth date and gender.

T2: Providers evaluate and review activity data before submission to ensure data is entered correctly for the correct patient. Providers are cautioned not to use RPE for unmediated treatment or diagnosis, i.e., a doctor must always intervene prior to treatment or diagnosis to ensure that errors that may occur in submission are checked by a human prior to engaging in any treatment or diagnosis of a patient.

T3: Secure internal networks from unauthorized access.

T4: Ensure strong password use for applications with access to PHI.

T5: Employ a restrictive RBAC scheme for applications providing access to PHI.

T6: Obtain patient consents and ensure patient data requiring increased protection is clearly identified.

T7: Workflow processes should be designed to ensure provider to patient communication occurs prior to sharing results with negative implications.

T8: Assign pseudonymized identifiers in processes when it is required to shield real identities.

T9: Actors are advised to consider the usability of their logging and audit repository implementation.

X.7 Requirements of Actors

No requirements beyond those described in section X.3.

X.8 Content Modules

Not applicable.

X.9 Process Flow

The basic process flow for RPE is shown below:

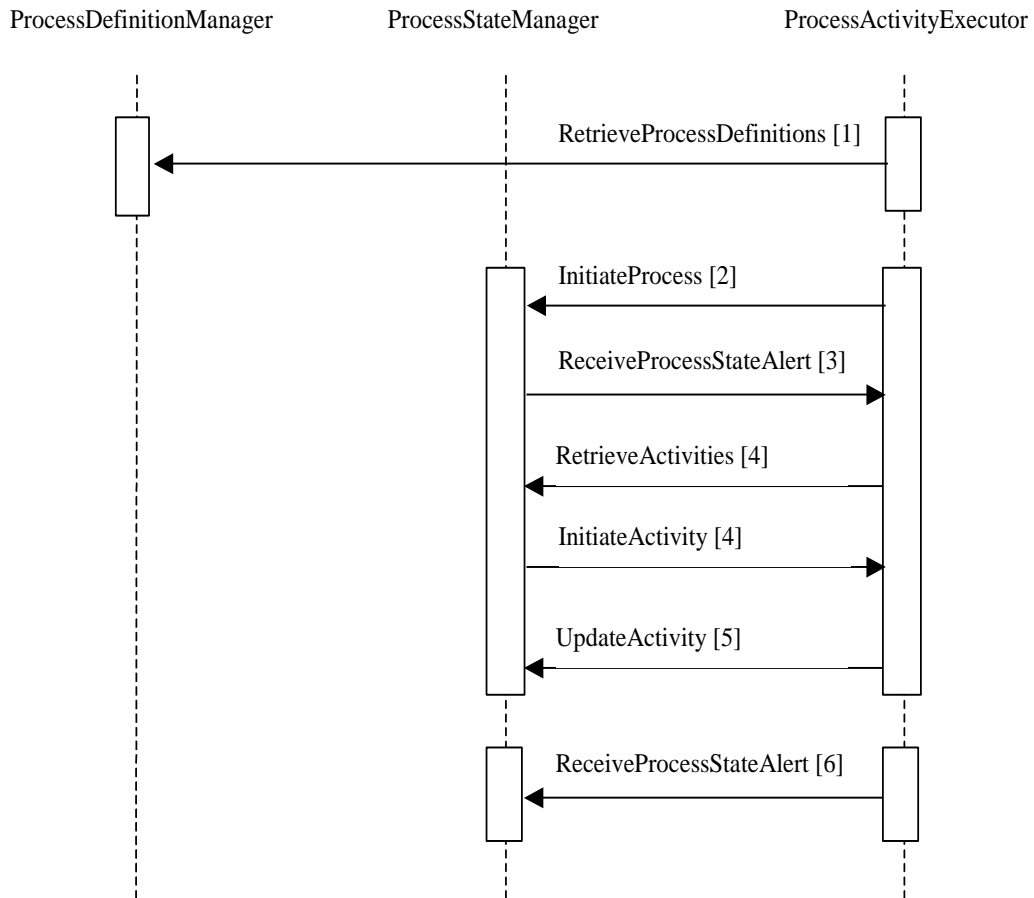


Figure X.9-1: Basic Process Flow in RPE Profile

1. The ProcessActivityExecutor retrieves process definitions of potential interest from the ProcessDefinitionManager.
2. The ProcessActivityExecutor request's process initiation by the ProcessStateManager forwarding a given process definition identifier as well as other required data, e.g., a patient identifier, demographics or eligibility criteria.
3. The ProcessStateManager notifies the ProcessActivityExecutor that the process is proceeding or otherwise.

4. The ProcessActivityExecutor either:
 - Retrieves the current activity(ies) it has to perform when the activity is created and managed by the ProcessStateManager or optionally,
 - Creates the activity itself but only as specified in a definition sent to it by the ProcessStateManager
5. After performance of an activity, the ProcessActivityExecutor sends the ProcessStateManager the updated activity state and data.
6. The ProcessActivityExecutor can always notify the ProcessStateManager of unscheduled events that may affect the process state, e.g., a patient withdrawal from a clinical trial.

An alternate flow when subscription is used by the ProcessActivityExecutor is shown below:

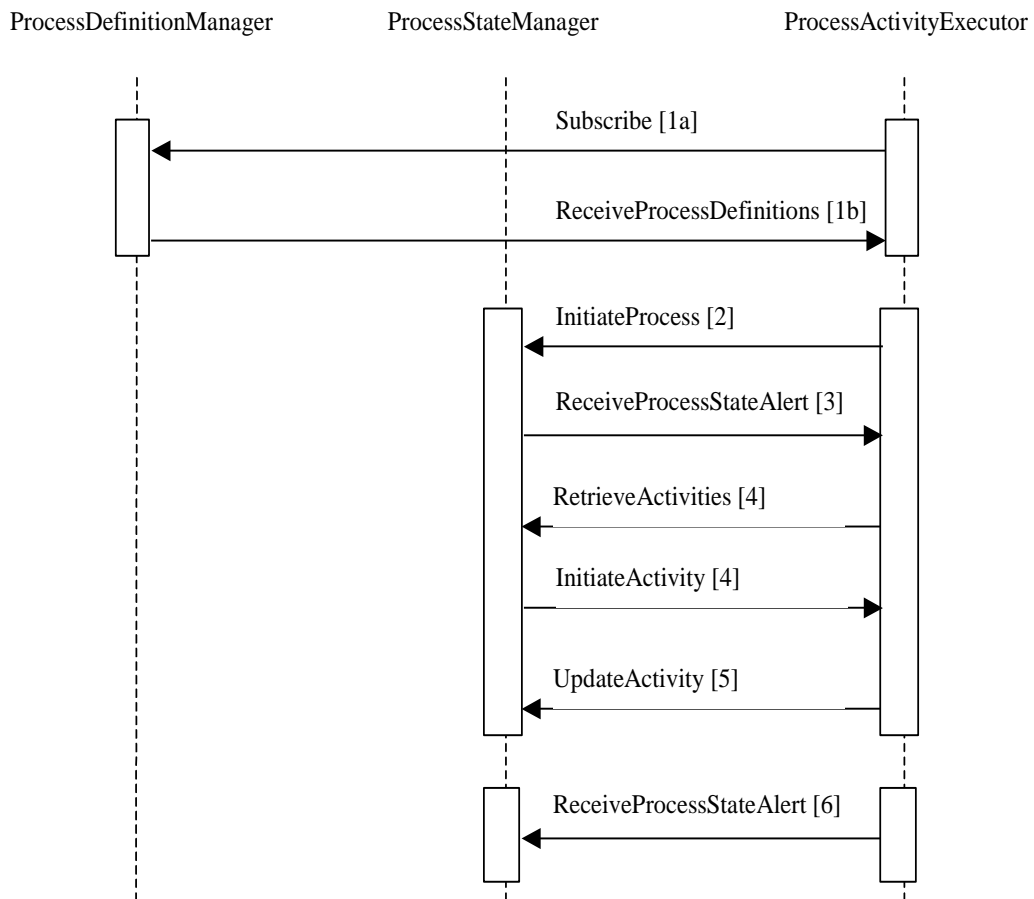


Figure X.9-2: Subscription Process Flow in RPE Profile

1. Replace step 1 in previous sequence with the following two steps:
 - a. The ProcessActivityExecutor sends the ProcessDefinitionManager a subscription request to new or updated process definitions of potential interest.

- b. When process definitions of interest to subscribers are available the ProcessDefinitionManager proactively forwards them to subscribed ProcessActivityExecutors.

Steps 2 thru 6 remain as for the Basic Process Flow.

An alternate flow when activity retrieval and update is accompanied by RFD form retrieval and submission, is shown below:

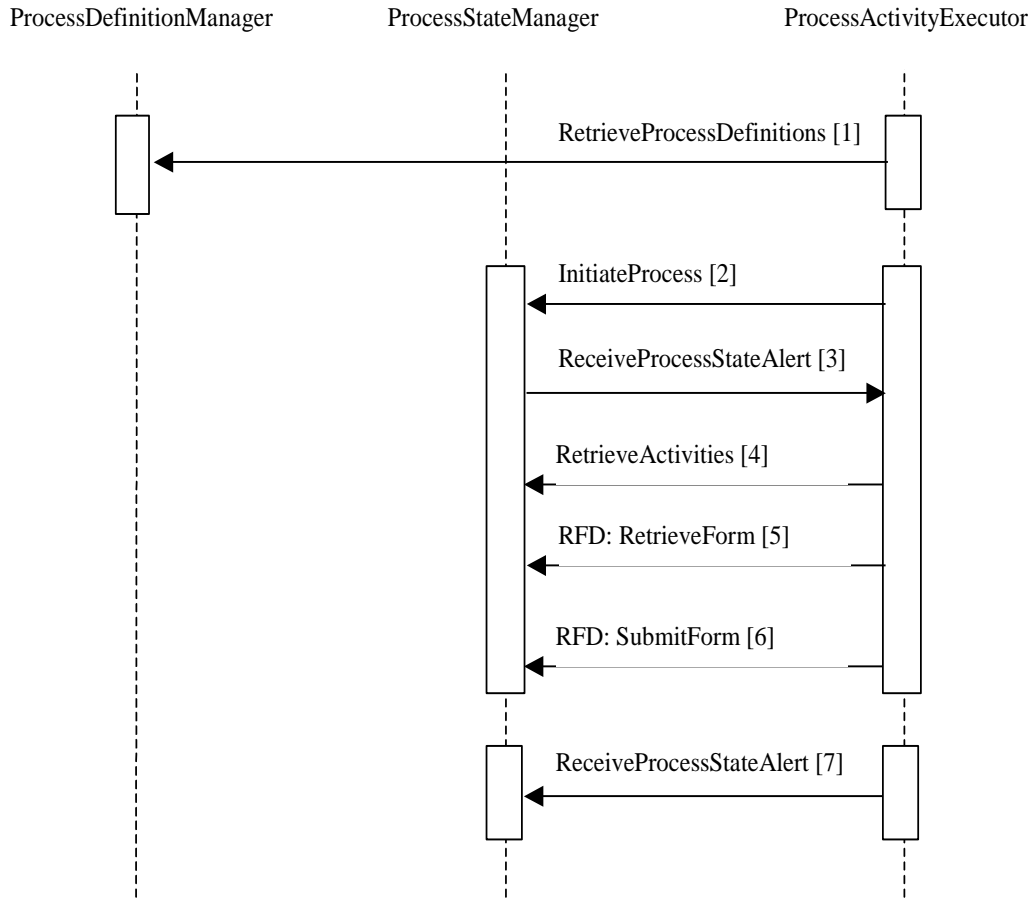


Figure X.9-3: RFD Process Flow in RPE Profile

Steps 1 thru 3 remain as for the Basic Process Flow.

- 4. The ProcessActivityExecutor retrieves the current activity(ies) it has to perform with the activity created and managed by the ProcessStateManager. The retrieved activity data contains an associated RFD form identifier.
- 5. The ProcessActivityExecutor retrieves the form from an RFD Form Manager (whose location may already be known and/or encapsulated within the retrieved activity data).

6. The ProcessActivityExecutor fills out and submits the form with data collected from its execution of the activity. This is equivalent to the role of UpdateActivity and as for the latter form submission may be used for activity data update and/or activity completion.

Step 7 is the same as 6 for the Basic Process Flow.

Appendix A: Actor Summary Definitions

Process Definition Manager – A system that manages access to a repository of process definitions allowing for search and retrieval. An example would be a research sponsor providing access to the clinical research protocols it has developed. Another would be a provider of evidence-based clinical practice guidelines.

Process State Manager – A system that manages the initiation and state of runtime process instances. The Process State Manager typically also supports the initiation and lifecycle management of task activities associated with a process while providing the ability for task performers to retrieve and update activities. An example would be a research sponsor conducting clinical trials in conjunction with its EHR participants. Another would be an EHR performing clinical care in accord with executable guideline-based processes.

Process Activity Executor – A system that performs activities as prescribed by a running process being managed by a Process State Manager. The Process Activity Executor retrieves current activity or task lists, works its list, updating the Process State Manager on activity state until completion. This cycle is repeated until all process activities have been worked and the process itself completes. An example would be an EHR performing activities as part of a clinical trial being managed by a research sponsor. Another would be an EHR performing guideline-based care process activities.

Appendix B: Transaction Summary Definitions

RetrieveProcessDefinitions – enables access to one or more process definitions specified by an identifier or other query criteria. This transaction is implemented by the ProcessDefinitionManager and used by both the ProcessStateManager – to deploy processes it wishes to manage – and the ProcessActivityExecutor – to examine processes it may be interested in becoming an activity participant.

Subscribe – allows either a ProcessStateManager or ProcessActivityExecutor to optionally establish a subscription to new or amended process definitions of matching interest, as they become available from a ProcessDefinitionManager.

ReceiveProcessDefinitions – enables either a ProcessStateManager or ProcessActivityExecutor to receive new or amended process definitions of matching interest, as defined by a previous Subscribe transaction, as they become available from a ProcessDefinitionManager.

InitiateProcess – enables a ProcessActivityExecutor to initiate a new process to be managed by a ProcessStateManager, e.g., an EHR entering a new patient candidate in a clinical trial being managed by a research sponsor.

RetrieveActivities – enables a ProcessActivityExecutor to retrieve the current set of activities it needs to execute as part of a process managed by a ProcessStateManager.

UpdateActivity – allows a ProcessActivityExecutor to provide an update on activity’s state or data to a ProcessStateManager for a process it is a participant in.

ReceiveProcessStateAlert – provides the ProcessStateManager and ProcessActivityExecutor the ability to notify each other of unscheduled events that effect the state of the process, e.g., an EHR patient withdrawing from a clinical trial or, a study being placed on hold.

InitiateActivity – allows the ProcessActivityExecutor to be in control of the lifecycle of activities it performs, e.g., an EHR may want to use its own task processor and interfaces to create and manage the activities it needs to perform as part of a process managed by the ProcessStateManager.

Glossary

Add the following terms to the Glossary:

Process Definition – A design definition of a process flow of activities involving one or more role-based activity performers. A process definition is implemented in XML and deployable to a runtime process engine.

Process – A specific instance of a process definition running in a process engine.

Activity Definition – A definition of a individual task activity which is deployable to a runtime task processor. Typically an activity is defined as part of a process definition but standalone activities may also be defined.

Activity – A specific instance of an activity created in, and available from, a task processor.

Volume 2 - Transactions

Add Section 3.Y

3. IHE Transactions

3.Y1 RetrieveProcessDefinitions

This section corresponds to transaction QRPH-Y1 of the IHE QRPH Transaction Framework QRPH-Y1 is used by the ProcessDefinitionManager and ProcessActivityExecutor actors.

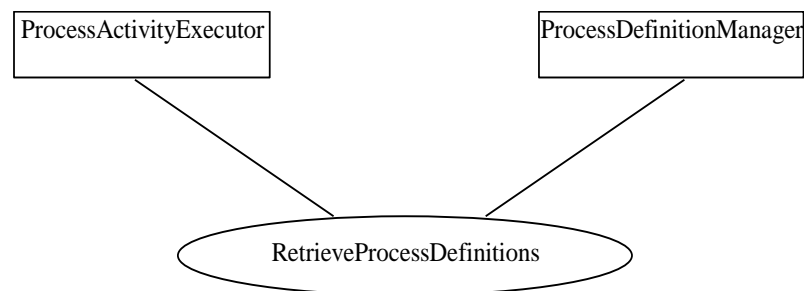
3.Y1.1 Scope

This transaction involves a ProcessActivityExecutor requesting one or more process definitions from a ProcessDefinitionManager. The ProcessActivityExecutor either has:

- A set of process definition identifiers obtained by means outside the scope of this profile or,
- Optionally uses either,
 - a set of SQL-like query clauses or,
 - a custom query interface supported by the given ProcessDefinitionManager.

The ProcessDefinitionManager returns a list of matching process definitions or else it returns an error response.

3.Y1.2 Use Case Roles



Actor: ProcessActivityExecutor

Role: A system that knows how to execute activities that are part of a process.

Actor: ProcessDefinitionManager

Role: A system that provides a set of process definitions upon request.

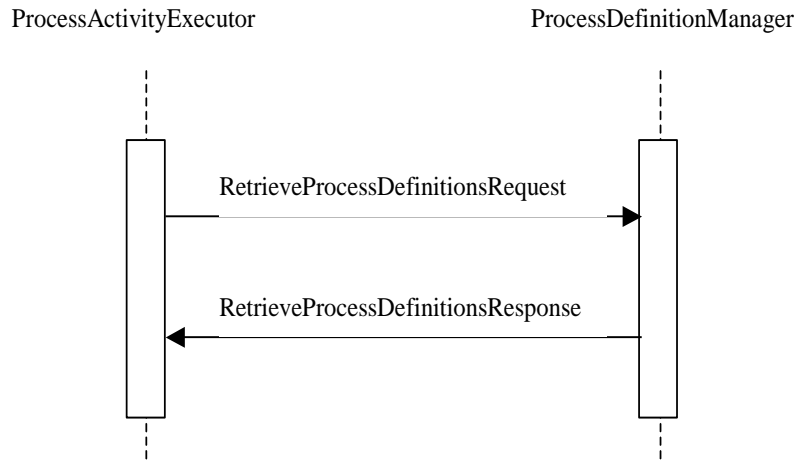
3.Y1.3 Referenced Standards

Implementers of this transaction shall comply with all requirements where applicable described in:

- ITI TF-2: Appendix V Web Services for IHE Transactions

- OASIS Web Services Business Process Execution Language (WS-BPEL) available at: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf>
- Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. <http://www.w3.org/TR/REC-xml>

3.Y1.4 Interaction Diagram



3.Y1.4.1 RetrieveProcessDefinitions Message

3.Y1.4.1.1 Trigger Events

The ProcessActivityExecutor, based upon human decision or application of a rule for automatic operation, wants to obtain one or more process definitions based on matching ids or other criteria from the ProcessDefinitionManager.

3.Y1.4.1.2 Message Semantics

The following parameters are specified for the body of this transaction.

Table 3.Y1.4.1.2–1: RetrieveProcessDefinitionsRequest Parameters

Parameter Name	REQ	Description	Value
processDefinitionIdentifier (list)	R	An identifier (or list of identifiers) for the process definition(s) to be retrieved.	The identifier is an XML element of any type.
or			
input	R	The context and/or input data for a custom retrieve operation. Optional specification of an expected pass-thru interface can be supplied.	A specialization of a complex XML element described in Table 3.Y1.4.1.2-2.

and			
maxOccurs	O	If specified, the number of tasks returned by either of the above 3 options MUST NOT exceed this limit.	An XML element of type int.
startIndex	O	The startIndex can be used to perform multiple identical queries and iterate over result sets where the maxTasks size exceeds the query limit.	An XML element of type int.

Section 3.Y1.6 describes the Web Services protocol requirements and the format of the message in full detail.

Table 3.Y1.4.1.2–2: InputType Parameter

Parameter Name	REQ	Description	Value
input	R	A flexible specification for the forwarding of context and input data to a custom operation implementation. Optional specification of an expected pass-thru interface can be supplied.	A complex XML element consisting of the following child elements and attributes.
contextData	O	Context data associated with the request input such as process identification or patient information.	A complex XML element consisting of the following child elements and attributes.
organizationIdentifier	O	Identifier for involved organization.	Each child element is of any type. See Table 3.Y1.4.1.2-3 for the name redefinitions available per domain.
alternateOrganizationIdentifier	O	Alternate identifier for organization.	
patientIdentifier	O	Identifier for the patient.	
alternatePatientIdentifier	O	Alternate identifier for patient.	
patientData	O	Patient data, e.g., demographics.	
processDefinitionIdentifier	O	Identifier for the process definition.	
processIdentifier	O	Identifier for the process instance.	
activityIdentifier	O	Identifier for the activity instance.	
<any>		Arbitrary context data per custom operation specification.	
inputData	O	Arbitrary input data per custom operation specification.	An XML element of any type.
portType	O	The WSDL port type of the associated pass-thru operation.	An XML attribute of type qualified name.
operation	O	The WSDL name of the associated pass-thru operation.	An XML attribute of type non-colonized name.

Table 3.Y1.4.1.2–3: Element Name Substitutions

Standard Name	Substitute Name	Description
organizationIdentifier	studySiteIdentifier	Clinical research study site.
alternateOrganizationIdentifier	alternateStudySiteIdentifier	Alternate site identifier (e.g., pseudonymized).
patientIdentifier	candidateIdentifier	Clinical research candidate.
alternatePatientIdentifier	subjectIdentifier	Clinical research subject.

processDefinitionIdentifier	studyDesignIdentifier publicHealthProcessDefinitionIdentifier qualityProcessDefinitionIdentifier	Clinical study design identifier. PH case reporting process definition identifier. Quality process definition identifier.
processIdentifier	studyIdentifier publicHealthProcessIdentifier qualityProcessIdentifier	Clinical study instance identifier. PH case reporting process instance identifier. Quality measure process instance identifier.
processDefinition	studyDesign publicHealthProcessDefinition qualityProcessDefinition	Clinical study design definition. PH case reporting process definition. Quality measure process definition.

3.Y1.4.1.3 Expected Actions

Upon reception of the RetrieveProcessDefinitions Message, the ProcessDefinitionManager shall parse the request and return the requested process definitions in the RetrieveProcessDefinitionsResponse message.

The successful response shall be one or more well-formed XML documents of the type requested, and matching the requested criteria.

Otherwise SOAP faults shall be generated accordingly:

- If there is missing information, such as no process definition identifier list, query or input parameter or the request is otherwise malformed. The fault should include:
 - faultcode: Client
 - faultstring: Invalid Request
 - and may provide further information in the details element.
- If no matching process definition is available The fault should include:
 - faultcode: Client
 - faultstring: No Matching Process Definition
-

3.Y1.4.2 RetrieveProcessDefinitionsResponse Message

3.Y1.4.2.1 Trigger Events

The delivery of a set of process definitions is triggered by a ProcessDefinitionManager actor responding to a RetrieveProcessDefinitions message.

3.Y1.4.2.2 Message Semantics

A matching list of one or more process definitions is returned. The format of each process definition is a well-formed XML document of the type requested.

3.Y1.4.2.3 Expected Actions

The ProcessActivityExecutor shall consume the set of process definitions based on the business rules of the system. If a SOAP fault is received then this fault should be handled based on the business rules of the system.

3.Y1.5 Security Considerations

See section X.6.

3.Y1.6 Protocol Requirements

The RetrieveProcessDefinitions request and response shall be transmitted using Synchronous Web Services Exchange, according to the requirements specified in ITI TF-2: Appendix V Web Services for IHE Transactions.

Table 3.Y1.6-1: WSDL Namespace Definitions

ihe	urn:ihe:qrph:rpe:2009
soap12	http://schemas.xmlsoap.org/wsdl/soap12/
wsaw	http://www.w3.org/2005/08/addressing
xsd	http://www.w3.org/2001/XMLSchema

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

- The following types shall be imported (xds:import) in the /definitions/types section:
namespace="urn:ihe:qrph:rpe:2009", schema="RPE.xsd"
- The /definitions/message/part/@element attribute of the RetrieveProcessDefinitions request message shall be defined as: "ihe:RetrieveProcessDefinitionsRequest"
- The /definitions/message/part/@element attribute of the RetrieveProcessDefinitions response message shall be defined as: "ihe:RetrieveProcessDefinitionsResponse"
- The /definitions/portType/operation/input/@wsaw:Action attribute for the RetrieveProcessDefinitions request message shall be defined as "urn:ihe:qrph:2009:RetrieveProcessDefinitions"
- The /definitions/portType/operation/output/@wsaw:Action attribute for the RetrieveProcessDefinitions response message shall be defined as: "urn:ihe:qrph:2009:RetrieveProcessDefinitionsResponse"
- The /definitions/binding/operation/soap12:operation/@soapAction attribute shall be defined as "urn:ihe:qrph:2009:RetrieveProcessDefinitions"

These are the requirements that affect the wire format of the SOAP message. The other WSDL properties are only used within the WSDL definition and do not affect interoperability. Full sample request and response messages are in the section 3.Y1.6.1 Sample SOAP Messages.

3.Y1.6.1 Sample SOAP Messages

The samples in the following two sections show a typical request and corresponding response as contained in a SOAP Body. *Note the SOAP Header should be populated according to the IHE Appendix V: Web Services for IHE Transactions and in accord with section 3.Y1.5 Security Considerations.*

A full WSDL for the ProcessDefinitionManager actor is found on the IHE FTP site at: ftp://ftp.ihe.net/Quality/2010_2011_YR_4/QRPH%20Technical%20Committee/Maintenance%20work/RPE/

3.Y1.6.1.1 Sample RetrieveProcessDefinitions SOAP Request

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

HL7-V3 Example (single):

```
<soap:Body>
  <rpe:RetrieveProcessDefinitionsRequest xmlns:rpe="urn:ihe:qrph:rpe:2009" type="HL7-V3">
    <rpe:studyDesignIdentifier>
      <id xmlns="urn:hl7-org:v3" root="1.2.5.2.3.4" extension="1006"/>
    </rpe:studyDesignIdentifier>
  </rpe:RetrieveProcessDefinitionsRequest>
</soap:Body>
```

CDISC-ODM Example (multiple):

```
<soap:Body>
  <rpe:RetrieveProcessDefinitionsRequest xmlns:rpe="urn:ihe:qrph:rpe:2009" type="CDISC-ODM">
    <rpe:studyDesignIdentifier>studyDefinitionOID1</rpe:studyDesignIdentifier>
    <rpe:studyDesignIdentifier>studyDefinitionOID2</rpe:studyDesignIdentifier>
  </rpe:RetrieveProcessDefinitionsRequest>
</soap:Body>
```

3.Y1.6.1.2 Sample RetrieveProcessDefinitions SOAP Response

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

HL7-V3 Example (single):

```
<soap:Body>
  <rpe:RetrieveProcessDefinitionsResponse xmlns:rpe="urn:ihe:qrph:rpe:2009">
    <rpe:studyDesign>
      <bpel:process xmlns:tns="http://www.company.com/rpe" xmlns:rpe="urn:ihe:qrph:rpe:2009"
        xmlns:hl7="urn:hl7-org:v3" xmlns:bpel="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
        xmlns:b4p="http://docs.oasis-open.org/ns/bpel4people/bpel4people/200803"
        xmlns:htd="http://docs.oasis-open.org/ns/bpel4people/ws-humantask/200803"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:bpelx="http://schemas.oracle.com/bpel/extension">
        ...
      <bpel:sequence>
        <!-- Start of clinical trial process for a patient -->
        <!-- ProcessStateManager.InitiateProcess -->
```

```

        <bpel:receive createInstance="yes" name="Enter_Patient_Request"
partnerLink="ProcessActivityExecutor" variable="InitiateProcessRequest"/>
        <bpel:scope name="EP1.PEROS">
            <!-- Screening, assignment and enrollment activities would be specified in
here-->
            ...
            <!-- Treatment -->
            <bpel:scope name="AR1.ENCSEQ">
                <bpel:sequence>
                    <bpel:extensionActivity>
                        <b4p:peopleActivity inputVariable="PatientDataInput"
name="Physician_Baseline" outputVariable="PatientDataOutput">
                            <htd:task name="Physician_Baseline_Task">
                                <htd:peopleAssignments>
                                    <htd:potentialOwners>
                                        <htd:from logicalPeopleGroup="ehrId"/>
                                    </htd:potentialOwners>
                                </htd:peopleAssignments>
                                <htd:renderings>
                                    <htd:rendering xmlns:rpe="urn:ihe:qrph:rpe:2009"
type="rpe:FormRenderingType">
                                        <rpe:formRendering>

<rpe:formIdentifier>"FM.PHYSICIANBASELINE"</rpe:formIdentifier>

<rpe:formManagerURL>"http://www.company.com/services/FormManager_Port_Soap12"</rpe:formM
anagerURL>

                                        </rpe:formRendering>
                                    </htd:rendering>
                                </htd:renderings>
                            </htd:task>
                        </b4p:peopleActivity>
                    </bpel:extensionActivity>
                    ...
                </bpel:sequence>
            </bpel:scope>
        </bpel:scope>
    </bpel:sequence>
</rpe:studyDesign>
</rpe:RetrieveProcessDefinitionsResponse>
</soap:Body>

```

3.Y2 Subscribe

This section corresponds to transaction QRPH-Y2 of the IHE QRPH Transaction Framework QRPH-Y2 is used by the ProcessDefinitionManager and ProcessActivityExecutor actors.

3.Y2.1 Scope

This transaction implements one half of a “publish/subscribe” model allowing a ProcessActivityExecutor to receive process definitions of specified interest and type, asynchronously, i.e., in “push” fashion, from a ProcessDefinitionManager.

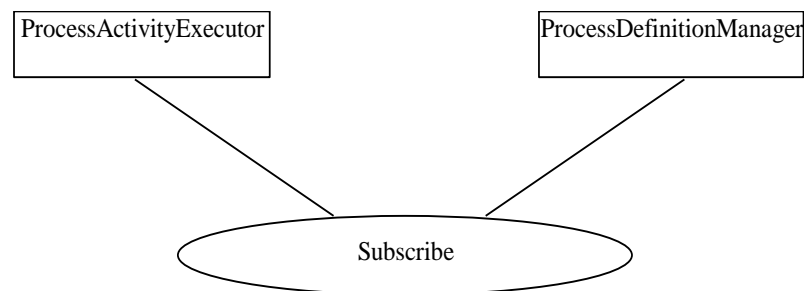
The Subscribe transaction can be either:

- A wrapper of a custom operation of the ProcessDefinitionManager that registers (or unsubscribes or renews) a subscription on behalf of the ProcessActivityExecutor for receipt of process definitions per specific “topic” of interest when available, or,

- A wrapper of the Subscribe operation from the Web Services Base Notification 1.3 standard, with the ProcessDefinitionManager playing the role of both NotificationProducer and SubscriptionManager (in DSUB this is equivalent to the combined Publisher/Notification Broker role in section 4.4.1.4 of the supplement).

When document messages are available, the ProcessDefinitionManager pushes a list of process definitions using the ReceiveProcessDefinitions transaction described later, or else it returns an error response.

3.Y2.2 Use Case Roles



Actor: ProcessActivityExecutor

Role: A system that knows how to execute activities that are part of a process.

Actor: ProcessDefinitionManager

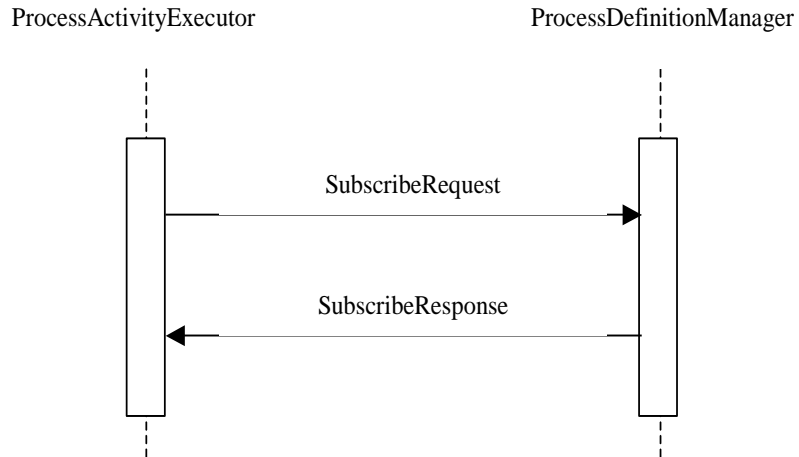
Role: A system that provides a set of process definitions upon request.

3.Y2.3 Referenced Standards

Implementers of this transaction shall comply with all requirements where applicable described in:

- ITI TF Supplement: Document Metadata Subscriptions (DSUB)
- ITI TF-2: Appendix V Web Services for IHE Transactions
- OASIS Web Services Base Notification 1.3 available at: http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf
- Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. <http://www.w3.org/TR/REC-xml>

3.Y2.4 Interaction Diagram



3.Y2.4.1 Subscribe Message

3.Y2.4.1.1 Trigger Events

The ProcessActivityExecutor, based upon human decision or application of a rule for automatic operation, wants to receive one or more process definitions on a subscription basis from the ProcessDefinitionManager.

3.Y2.4.1.2 Message Semantics

The following parameters are specified for the body of this transaction.

Table 3.Y2.4.1.2–1: SubscribeRequest Parameters

Parameter Name	REQ	Description	Value
input	R	Either: a WS-BaseNotification NotificationProducer or SubscriptionManager request to a pass-thru interface or, the context and/or input data for a custom subscribe operation.	A specialization of a complex XML element described in Table 3.Y1.4.1.2-2.

Section 3.Y2.6 describes the Web Services protocol requirements and the format of the message in full detail.

3.Y2.4.1.3 Expected Actions

Upon reception of the Subscribe message, the ProcessDefinitionManager shall parse the request and register the ProcessActivityExecutor as a subscriber for the given topic of interest.

The successful response shall be generated accordingly:

- If a wrapper of a custom subscribe operation, in accordance with the ProcessDefinitionManager custom specification.
- If a wrapper of the WS-BaseNotification Subscribe operation, a wrapped WS-BaseNotification SubscribeResponse.

Otherwise SOAP faults shall be generated accordingly:

- If a wrapper of a custom subscribe operation:
- If subscription is not supported. The fault should include:
 - faultcode: Client
 - faultstring: Resource Unknown
- If process type is not supported. The fault should include:
 - faultcode: Client
 - faultstring: Type Not Supported
- If topic expression is not supported. The fault should include:
 - faultcode: Client
 - faultstring: Topic Not Supported
- If a wrapper of the WS-BaseNotification Subscribe operation, faults shall be generated in accordance with the specification.

3.Y2.4.2 SubscribeResponse Message

3.Y2.4.2.1 Trigger Events

The message is triggered on successful subscription of a ProcessActivityExecutor in response to SubscribeRequest.

3.Y2.4.2.2 Message Semantics

A response is returned which confirms subscription to the topic of interest and may contain a subscription endpoint reference and termination time for the subscription itself.

If a wrapper of the WS-BaseNotification Subscribe operation, a wrapper of the WS-BaseNotification SubscribeResponse is returned.

3.Y2.4.2.3 Expected Actions

The ProcessActivityExecutor shall consume the set of process definitions based on the business rules of the system. If a SOAP fault is received then this fault should be handled based on the business rules of the system.

3.Y2.5 Security Considerations

See section X.6.

3.Y2.6 Protocol Requirements

The Subscribe request and response shall be transmitted using Synchronous Web Services Exchange, according to the requirements specified in ITI TF-2: Appendix V Web Services for IHE Transactions.

Table 3.Y2.6-1: WSDL Namespace Definitions

ihe	urn:ihe:qrph:rpe:2009
soap12	http://schemas.xmlsoap.org/wsdl/soap12/
wsaw	http://www.w3.org/2005/08/addressing
xsd	http://www.w3.org/2001/XMLSchema

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

- The following types shall be imported (xds:import) in the /definitions/types section:
namespace="urn:ihe:qrph:rpe:2009", schema="RPE.xsd"
- The /definitions/message/part/@element attribute of the Subscribe request message shall be defined as: "ihe:SubscribeRequest"
- The /definitions/message/part/@element attribute of the Subscribe response message shall be defined as: "ihe:SubscribeResponse"
- The /definitions/portType/operation/input/@wsaw:Action attribute for the Subscribe request message shall be defined as "urn:ihe:qrph:2009:Subscribe"
- The /definitions/portType/operation/output/@wsaw:Action attribute for the Subscribe response message shall be defined as: "urn:ihe:qrph:2009:SubscribeResponse"
- The /definitions/binding/operation/soap12:operation/@soapAction attribute shall be defined as "urn:ihe:qrph:2009:Subscribe"

These are the requirements that affect the wire format of the SOAP message. The other WSDL properties are only used within the WSDL definition and do not affect interoperability. Full sample request and response messages are in the section 3.Y2.6.1 Sample SOAP Messages.

3.Y2.6.1 Sample SOAP Messages

The samples in the following two sections show a typical request and corresponding response as contained in a SOAP Body. *Note the SOAP Header should be populated according to the IHE Appendix V: Web Services for IHE Transactions and in accord with section 3.Y2.5 Security Considerations.*

A full WSDL for the ProcessDefinitionManager actor is found on the IHE FTP site at: ftp://ftp.ihe.net/Quality/2010_2011_YR_4/QRPH%20Technical%20Committee/Maintenance%20work/RPE/

3.Y2.6.1.1 Sample Subscribe SOAP Request

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

Subscription Request Example:

```
<soap:Body>
<rpe:SubscribeRequest xmlns:rpe="urn:ihe:qrph:rpe:2009" type="WS-BPEL">
  <rpe:input xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
xmlns:wsa="http://www.w3.org/2005/08/addressing" portType="wsnt:NotificationProducer"
operation="Subscribe">
    <wsnt:Subscribe>
      <wsnt:ConsumerReference>
        <wsa:Address>
          https://www.company.com/services/ProcessActivityExecutor_Port_Soap12</wsa:Address>
        </wsnt:ConsumerReference>
        <wsnt:Filter>
          <wsnt:TopicExpression Dialect="http://docs.oasis-open.org/wsn/t-
1/TopicExpression/Simple">processDefinitionTopic</wsnt:TopicExpression>
        </wsnt:Filter>
        <wsnt:InitialTerminationTime>2010-05-31T00:00:00.000000Z</wsnt:InitialTerminationTime>
      </wsnt:Subscribe>
    </rpe:input>
  </rpe:SubscribeRequest>
</soap:Body>
```

3.Y2.6.1.2 Sample Subscribe SOAP Response

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

```
<soap:Body>
<rpe:SubscribeResponse xmlns:rpe="urn:ihe:qrph:rpe:2009">
  <rpe:output xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <wsnt:SubscribeResponse>
      <wsnt:SubscriptionReference>
        <wsa:Address>
          https://www.company.com/services/ProcessDefinitionManager_Port_Soap12</wsa:Address>
        </wsnt:SubscriptionReference>
      </wsnt:SubscribeResponse>
    </rpe:output>
  </rpe:SubscribeResponse>
</soap:Body>
```

3.Y3 ReceiveProcessDefinitions

This section corresponds to transaction QRPH-Y3 of the IHE QRPH Transaction Framework QRPH-Y3 is used by the ProcessDefinitionManager and ProcessActivityExecutor actors.

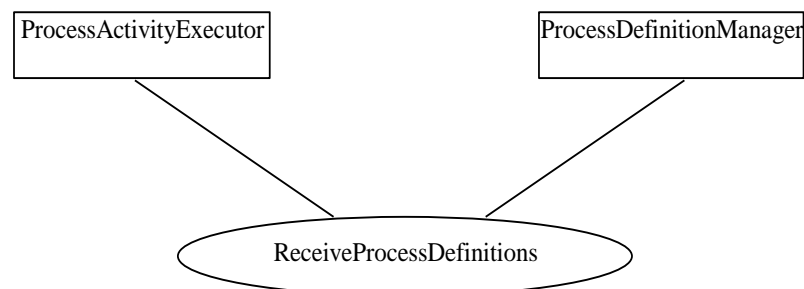
3.Y3.1 Scope

This transaction implements one half of a “publish/subscribe” model allowing a ProcessActivityExecutor to receive process definitions of specified interest and type, asynchronously, i.e., in “push” fashion, from a ProcessDefinitionManager.

The ReceiveProcessDefinitions transaction can be either:

- A wrapper of a custom operation of the ProcessActivityExecutor that receives process definitions as a result of a prior Subscribe request or,
- A wrapper of the Notify operation from the Web Services Base Notification 1.3 standard, with the ProcessActivityExecutor playing the role of NotificationConsumer (in DSUB this is equivalent to the combined Subscriber/Notification Recipient role in section 4.4.1.4 of the supplement).

3.Y3.2 Use Case Roles



Actor: ProcessActivityExecutor

Role: A system that knows how to execute activities that are part of a process.

Actor: ProcessDefinitionManager

Role: A system that provides a set of process definitions upon request.

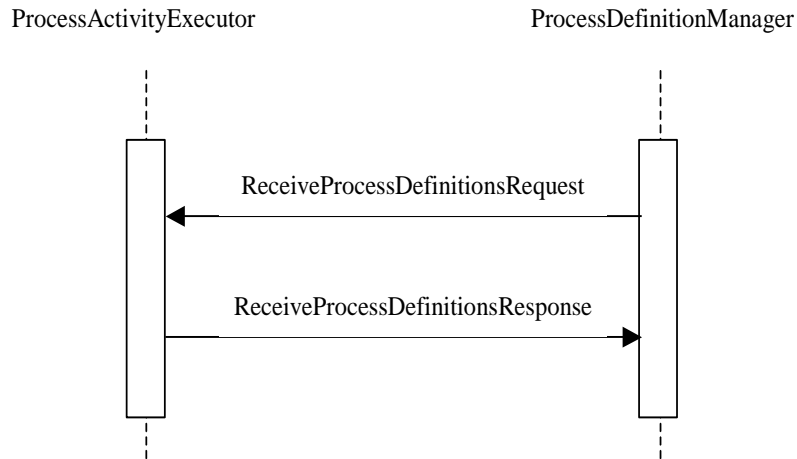
3.Y3.3 Referenced Standards

Implementers of this transaction shall comply with all requirements where applicable described in:

- ITI TF Supplement: Document Metadata Subscriptions (DSUB)
- ITI TF-2: Appendix V Web Services for IHE Transactions

- OASIS Web Services Base Notification 1.3 available at: http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf
- Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. <http://www.w3.org/TR/REC-xml>

3.Y3.4 Interaction Diagram



3.Y3.4.1 ReceiveProcessDefinitions Message

3.Y3.4.1.1 Trigger Events

The ProcessDefinitionManager, implementing a “publish/subscribe” model in which it is the publisher, forwards to all topic subscribers any process definitions published to those topics immediately as they are published.

3.Y3.4.1.2 Message Semantics

The following parameters are specified for the body of this transaction.

Table 3.Y3.4.1.2–1: ReceiveProcessDefinitionsRequest Parameters

Parameter Name	REQ	Description	Value
input	R	Either: a WS-BaseNotification NotificationConsumer Notify message to a pass-thru interface or, the context and/or input data for a custom ReceiveProcessDefinitions	A specialization of a complex XML element described in Table 3.Y1.4.1.2-2.

		operation.	
--	--	------------	--

Section 3.Y3.6 describes the Web Services protocol requirements and the format of the message in full detail.

3.Y3.4.1.3 Expected Actions

The ProcessActivityExecutor parses the ReceiveProcessDefinitions request and if well-formed: and of acceptable type(s) acknowledges receipt with a response code of “PROCESS_DEFINITION_RECEIVED”.

Otherwise SOAP faults shall be generated accordingly:

- If process definition(s) received are of an unrequested type, a SOAP fault shall be generated and shall include:
 - faultcode: Client
 - faultstring: Unexpected Process Type
- If process definition(s) are not well-formed, a SOAP fault shall be generated and shall include:
 - faultcode: Client
 - faultstring: Malformed Process Definition
 - and may provide further information in the details element.

3.Y3.4.2 ReceiveProcessDefinitionsResponse Message

3.Y3.4.2.1 Trigger Events

The message is triggered by a ProcessActivityExecutor receiving process definitions from a ProcessDefinitionManager in response to a previous Subscribe request.

3.Y3.4.2.2 Message Semantics

A responseCode XML element of type string is returned which confirms receipt of an acceptable set of process definitions with a value of “PROCESS_DEFINITION_RECEIVED”.

3.Y3.4.2.3 Expected Actions

The ProcessDefinitionManager shall handle the response code based on the business rules of the system. If a SOAP fault is received then this fault should also be handled based on the business rules of the system.

3.Y3.5 Security Considerations

See section X.6.

3.Y3.6 Protocol Requirements

The ReceiveProcessDefinitions request and response shall be transmitted using Synchronous Web Services Exchange, according to the requirements specified in ITI TF-2: Appendix V Web Services for IHE Transactions.

Table 3.Y3.6-1: WSDL Namespace Definitions

ihe	urn:ihe:qrph:rpe:2009
soap12	http://schemas.xmlsoap.org/wsdl/soap12/
wsaw	http://www.w3.org/2005/08/addressing
xsd	http://www.w3.org/2001/XMLSchema

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

- The following types shall be imported (xds:import) in the /definitions/types section:
namespace="urn:ihe:qrph:rpe:2009", schema="RPE.xsd"
- The /definitions/message/part/@element attribute of the ReceiveProcessDefinitions request message shall be defined as: "ihe:ReceiveProcessDefinitionsRequest"
- The /definitions/message/part/@element attribute of the Subscribe response message shall be defined as: "ihe:ReceiveProcessDefinitionsResponse"
- The /definitions/portType/operation/input/@wsaw:Action attribute for the Subscribe request message shall be defined as "urn:ihe:qrph:2009:ReceiveProcessDefinitions"
- The /definitions/portType/operation/output/@wsaw:Action attribute for the Subscribe response message shall be defined as:
"urn:ihe:qrph:2009:ReceiveProcessDefinitionsResponse"
- The /definitions/binding/operation/soap12:operation/@soapAction attribute shall be defined as "urn:ihe:qrph:2009:ReceiveProcessDefinitions"

These are the requirements that affect the wire format of the SOAP message. The other WSDL properties are only used within the WSDL definition and do not affect interoperability. Full sample request and response messages are in the section 3.Y3.6.1 Sample SOAP Messages.

3.Y3.6.1 Sample SOAP Messages

The samples in the following two sections show a typical request and corresponding response as contained in a SOAP Body. *Note the SOAP Header should be populated according to the IHE Appendix V: Web Services for IHE Transactions and in accord with section 3.Y3.5 Security Considerations.*

A full WSDL for the ProcessDefinitionManager actor is found on the IHE FTP site at: ftp://ftp.ihe.net/Quality/2010_2011_YR_4/QRPH%20Technical%20Committee/Maintenance%20work/RPE/

3.Y3.6.1.1 Sample ReceiveProcessDefinitions SOAP Request

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

```
<soap:Body>
  <rpe:ReceiveProcessDefinitionsRequest xmlns:rpe="urn:ihe:qrph:rpe:2009" type="CDISC-ODM">
    <rpe:input xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
      xmlns:wsa="http://www.w3.org/2005/08/addressing" portType="wsnt:NotificationConsumer"
      operation="Notify">
      <wsnt:Notify>
        <wsnt:NotificationMessage>
          <wsnt:SubscriptionReference>
            <wsa:Address>https://www.company.com/services/ProcessDefinitionManager_Port_Soap12</wsa:Address>
          </wsnt:SubscriptionReference>
          <wsnt:Topic Dialect="http://docs.oasis-open.org/wsn/t-
            1/TopicExpression/Simple">processDefinitionTopic</wsnt:Topic>
          <wsnt:ProducerReference>
            <wsa:Address>https://www.company.com/services/ProcessDefinitionManager_Port_Soap12</wsa:Address>
          </wsnt:ProducerReference>
          <wsnt:Message>
            <bpel:process xmlns:tns="http://www.company.com/rpe" xmlns:rpe="urn:ihe:qrph:rpe:2009"
              xmlns:hl7="urn:hl7-org:v3" xmlns:bpel="http://docs.oasis-
              open.org/wsbpel/2.0/process/executable"
              xmlns:b4p="http://docs.oasis-open.org/ns/bpel4people/bpel4people/200803"
              xmlns:htd="http://docs.oasis-open.org/ns/bpel4people/ws-humantask/200803"
              xmlns:xsd="http://www.w3.org/2001/XMLSchema"
              xmlns:bpelx="http://schemas.oracle.com/bpel/extension">
              ...
              <bpel:sequence>
                <!-- Start of clinical trial process for a patient -->
                <!-- ProcessStateManager.InitiateProcess -->
                <bpel:receive createInstance="yes" name="Enter_Patient_Request"
                  partnerLink="ProcessActivityExecutor" variable="InitiateProcessRequest"/>
                <bpel:scope name="EPl.PEROS">
                  <!-- Screening, assignment and enrollment activities would be specified in
                  here-->
                  ...
                  <!-- Treatment -->
                  <bpel:scope name="AR1.ENCSEQ">
                    <bpel:sequence>
                      <bpel:extensionActivity>
                        <b4p:peopleActivity inputVariable="PatientDataInput"
                          name="Physician_Baseline" outputVariable="PatientDataOutput">
                          <htd:task name="Physician_Baseline_Task">
                            <htd:peopleAssignments>
                              <htd:potentialOwners>
                                <htd:from logicalPeopleGroup="ehrId"/>
                              </htd:potentialOwners>
                            </htd:peopleAssignments>
                            <htd:renderings>
                              <htd:rendering xmlns:rpe="urn:ihe:qrph:rpe:2009"
                                type="rpe:FormRenderingType">
                                <rpe:formRendering>
                                  <rpe:formIdentifier>"FM.PHYSICIANBASELINE"</rpe:formIdentifier>
                                </htd:rendering>
                              </htd:renderings>
                            </htd:task>
                          </b4p:peopleActivity>
                        </bpel:extensionActivity>
                      </bpel:sequence>
                    </bpel:scope>
                  </!-- Treatment -->
                </bpel:scope>
              </bpel:scope name="EPl.PEROS">
            </bpel:process>
          </wsnt:Message>
        </wsnt:ProducerReference>
      </wsnt:Topic>
    </rpe:input>
  </rpe:ReceiveProcessDefinitionsRequest>
</soap:Body>
```

```
<rpe:formManagerURL>"http://www.company.com/services/FormManager_Port_Soap12"</rpe:formManagerURL>
    </rpe:formRendering>
  </htd:rendering>
</htd:renderings>
</htd:task>
</b4p:peopleActivity>
</bpel:extensionActivity>
...
</bpel:sequence>
</bpel:scope>
</bpel:scope>
</bpel:sequence>
</bpel:process>
</wsnt:Message>
<wsnt:NotificationMessage>
</wsnt:Notify>
</rpe:input>
</rpe:ReceiveProcessDefinitionsRequest>
</soap:Body>
```

3.Y3.6.1.2 Sample ReceiveProcessDefinitions SOAP Response

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

```
<soap:Body>
  <rpe:ReceiveProcessDefinitionsResponse xmlns:rpe="urn:ihe:qrph:rpe:2009">
    <rpe:responseCode>PROCESS_DEFINITION_RECEIVED</rpe:responseCode>
  </rpe:ReceiveProcessDefinitionsResponse>
</soap:Body>
```

3.Y4 InitiateProcess

This section corresponds to transaction QRPH-Y4 of the IHE QRPH Transaction Framework. QRPH-Y4 is used by the ProcessStateManager and ProcessActivityExecutor actors.

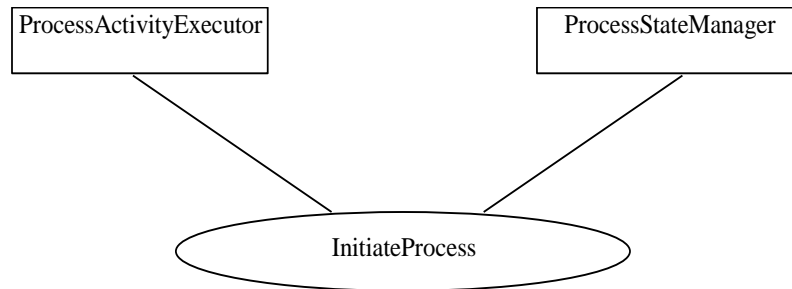
3.Y4.1 Scope

This transaction involves a ProcessActivityExecutor requesting initiation of a process with a ProcessStateManager.

In its request the ProcessActivityExecutor can supply:

- an organizational identifier
- a patient identifier
- other optional patient data such as demographics
- an identifier for the process definition of interest
- any other input data required by a custom or pass-thru operation implementation

3.Y4.2 Use Case Roles



Actor: ProcessActivityExecutor

Role: A system that knows how to execute activities that are part of a process.

Actor: ProcessStateManager

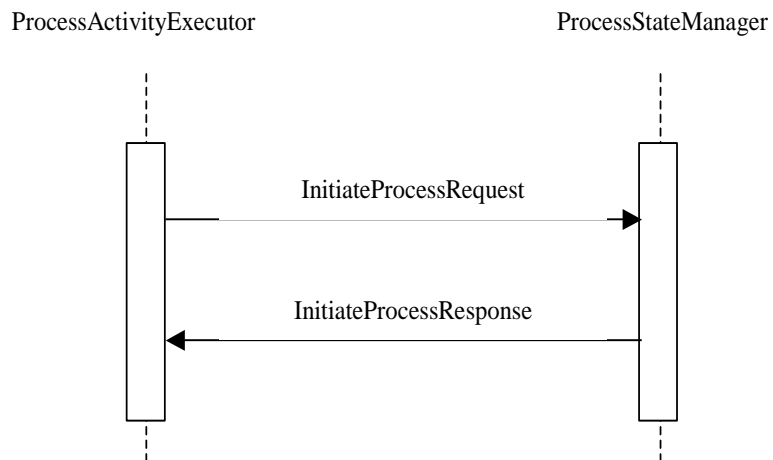
Role: A system that manages the runtime state of a process.

3.Y4.3 Referenced Standards

Implementers of this transaction shall comply with all requirements where applicable described in:

- ITI TF-2: Appendix V Web Services for IHE Transactions
- Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. <http://www.w3.org/TR/REC-xml>

3.Y4.4 Interaction Diagram



3.Y4.4.1 InitiateProcess Message

3.Y4.4.1.1 Trigger Events

The ProcessActivityExecutor, based upon human decision or application of a rule for automatic operation, wants to initiate a process with a ProcessStateManager.

3.Y4.4.1.2 Message Semantics

The following parameters are specified for the body of this transaction.

Table 3.Y4.4.1.2–1: InitiateProcessRequest Parameters

Parameter Name	REQ	Description	Value
input	R	Used for supplying: an organizational identifier a patient identifier other optional patient data (e.g., demographics) an identifier for the process definition of interest any other input data required by a custom or pass-thru operation implementation Supply of organization and process definition identifiers is mandatory either in the context or input data.	A specialization of a complex XML element described in Table 3.Y1.4.1.2-2.

Section 3.Y4.6 describes the Web Services protocol requirements and the format of the message in full detail.

3.Y4.4.1.3 Expected Actions

The ProcessStateManager parses the InitiateProcess request and if well-formed, initiates an instance of the given process and acknowledges the request with a response code of “PROCESS_INITIATED”.

Otherwise SOAP faults shall be generated accordingly:

- If request is not well-formed, a SOAP fault shall be generated and shall include:
 - faultcode: Client
 - faultstring: Invalid Request
 - and may provide further information in the details element.
- If the process failed to initiate due to a ProcessStateManager system error, a SOAP fault shall be generated and shall include:
 - faultcode: Server

- faultstring: System Error
- and may provide further information in the details element.

3.Y4.4.2 InitiateProcessResponse Message

3.Y4.4.2.1 Trigger Events

The message is triggered by a ProcessStateManager receiving an InitiateProcess request from a ProcessActivityExecutor and after successful initiation of the process.

3.Y4.4.2.2 Message Semantics

A responseCode XML element of type string is returned which confirms initiation of the process with a value of “PROCESS_INITIATED”.

3.Y4.4.2.3 Expected Actions

The ProcessStateManager shall generate the response code based on the business rules of the system. If a SOAP fault is received then this fault should also be handled based on the business rules of the system.

3.Y4.5 Security Considerations

See section X.6.

3.Y4.6 Protocol Requirements

The InitiateProcess request and response shall be transmitted using Synchronous Web Services Exchange, according to the requirements specified in ITI TF-2: Appendix V Web Services for IHE Transactions.

Table 3.Y4.6-1: WSDL Namespace Definitions

ihe	urn:ihe:qrph:rpe:2009
soap12	http://schemas.xmlsoap.org/wsdl/soap12/
wsaw	http://www.w3.org/2005/08/addressing
xsd	http://www.w3.org/2001/XMLSchema

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

- The following types shall be imported (xds:import) in the /definitions/types section:
namespace=”urn:ihe:qrph:rpe:2009”, schema=”RPE.xsd”
- The /definitions/message/part/@element attribute of the InitiateProcess request message shall be defined as: “ihe:InitiateProcessRequest”

- The /definitions/message/part/@element attribute of the InitiateProcess response message shall be defined as: “ihe:InitiateProcessResponse”
- The /definitions/portType/operation/input/@wsaw:Action attribute for the InitiateProcess request message shall be defined as “urn:ihe:qrph:2009:InitiateProcess”
- The /definitions/portType/operation/output/@wsaw:Action attribute for the InitiateProcess response message shall be defined as: “urn:ihe:qrph:2009:InitiateProcessResponse”
- The /definitions/binding/operation/soap12:operation/@soapAction attribute shall be defined as “urn:ihe:qrph:2009:InitiateProcess”

These are the requirements that affect the wire format of the SOAP message. The other WSDL properties are only used within the WSDL definition and do not affect interoperability. Full sample request and response messages are in the section 3.Y4.6.1 Sample SOAP Messages.

3.Y4.6.1 Sample SOAP Messages

The samples in the following two sections show a typical request and corresponding response as contained in a SOAP Body. *Note the SOAP Header should be populated according to the IHE Appendix V: Web Services for IHE Transactions and in accord with section 3.Y4.5 Security Considerations.*

A full WSDL for the ProcessStateManager actor is found on the IHE FTP site at:

ftp://ftp.ihe.net/Quality/2010_2011_YR_4/QRPH%20Technical%20Committee/Maintenance%20work/RPE/

3.Y4.6.1.1 Sample InitiateProcess SOAP Request

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

HL7-V3 Example:

```
<soap:Body>
  <rpe:InitiateProcessRequest xmlns:rpe="urn:ihe:qrph:rpe:2009">
    <rpe:input>
      <rpe:contextData>
        <rpe:studySiteIdentifier>
          <hl7:id xmlns:hl7="urn:hl7-org:v3" root="1.2.3.44.2.3.44" extension="ehrId"/>
        </rpe:studySiteIdentifier>
        <rpe:candidateIdentifier>
          <hl7:id xmlns:hl7="urn:hl7-org:v3" root="1.2.3.44.2.3.44" extension="01000124"/>
        </rpe:candidateIdentifier>
        <rpe:patientDemographics>
          <hl7:name xmlns:hl7="urn:hl7-org:v3">
            <family>WELLINGTON</family>
            <given>Masters</given>
          </hl7:name>
          <hl7:address xmlns:hl7="urn:hl7-org:v3">
            <country>USA</country>
            <state>WA</state>
            <city>SEATTLE</city>
          </hl7:address>
        </rpe:patientDemographics>
      </rpe:contextData>
    </rpe:input>
  </rpe:InitiateProcessRequest>
</soap:Body>
```

```
        <postalCode>98101</postalCode>
        <streetAddressLine>212 BRIDGE STREET</streetAddressLine>
    </hl7:address>
</rpe:patientDemographics>
<rpe:studyDesignIdentifier>
    <hl7:id xmlns:hl7="urn:hl7-org:v3" root="1.2.5.2.3.4" extension="1006"/>
</rpe:studyDesignIdentifier>
</rpe:contextData>
</rpe:input>
</rpe:InitiateProcessRequest>
</soap:Body>
```

3.Y4.6.1.2 Sample InitiateProcess SOAP Response

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

```
<soap:Body>
  <rpe:InitiateProcessResponse xmlns:rpe="urn:ihe:qrph:rpe:2009">
    <rpe:responseCode>PROCESS_INITIATED</rpe:responseCode>
  </rpe:InitiateProcessResponse>
</soap:Body>
```

3.Y5 RetrieveActivities

This section corresponds to transaction QRPH-Y5 of the IHE QRPH Transaction Framework. QRPH-Y5 is used by the ProcessStateManager and ProcessActivityExecutor actors.

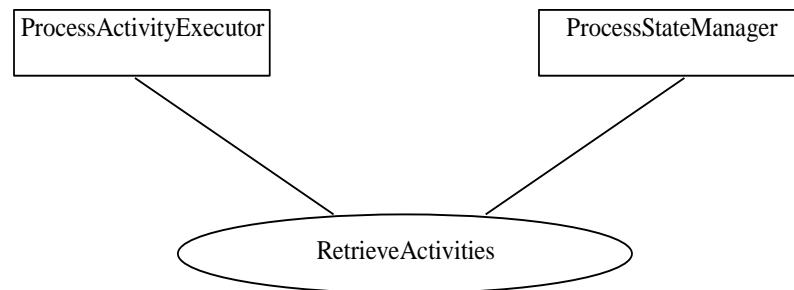
3.Y5.1 Scope

RetrieveActivities enables a ProcessActivityExecutor to retrieve the current set of activities it needs to execute as part of processes it is a participant in and managed by the given ProcessStateManager.

In its request the ProcessActivityExecutor can supply the following context:

- an organizational identifier
- a patient identifier
- an identifier for a process instance
- any other input data required by a custom or pass-thru operation implementation

3.Y5.2 Use Case Roles



Actor: ProcessActivityExecutor

Role: A system that knows how to execute activities that are part of a process.

Actor: ProcessStateManager

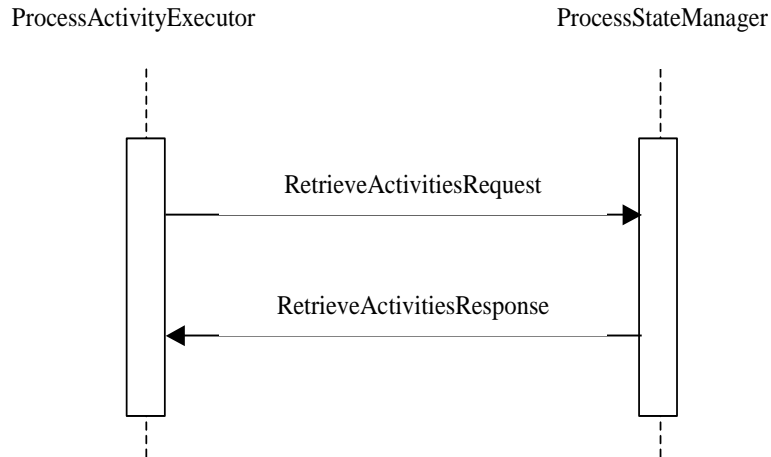
Role: A system that manages the runtime state of a process.

3.Y5.3 Referenced Standards

Implementers of this transaction shall comply with all requirements where applicable described in:

- ITI TF-2: Appendix V Web Services for IHE Transactions
- OASIS Web Services Business Process Execution Language (WS-BPEL) available at: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf>
- OASIS WS-BPEL Extension for People (BPEL4People) Specification Version 1.1 available at: <http://docs.oasis-open.org/bpel4people/bpel4people-1.1.pdf>
- OASIS Web Services – Human Task (WS-HumanTask) Specification Version 1.1 available at: <http://docs.oasis-open.org/bpel4people/ws-humantask-1.1.pdf>
- Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. <http://www.w3.org/TR/REC-xml>

3.Y5.4 Interaction Diagram



3.Y5.4.1 RetrieveActivities Message

3.Y5.4.1.1 Trigger Events

The ProcessActivityExecutor, based upon human decision or application of a rule for automatic operation, wants to retrieve the current set of process activities it needs to perform from a ProcessStateManager.

3.Y5.4.1.2 Message Semantics

The following parameters are specified for the body of this transaction.

Table 3.Y5.4.1.2–1: RetrieveActivitiesRequest Parameters

Parameter Name	REQ	Description	Value
input	R	Used for supplying: an organizational identifier a patient identifier an identifier for a process instance any other input data required by a custom or pass-thru operation implementation Supply of an organization identifier is mandatory either in the context or input data sections.	A specialization of a complex XML element described in Table 3.Y1.4.1.2-2.

Section 3.Y5.6 describes the Web Services protocol requirements and the format of the message in full detail.

3.Y5.4.1.3 Expected Actions

The ProcessStateManager parses the RetrieveActivities request and if well-formed, returns the current set of activities for the given organization and optionally constrained by a supplied context such as a process or patient identifier.

If no activities are outstanding an empty response is returned.

Otherwise SOAP faults shall be generated accordingly:

- If the request is not well-formed, a SOAP fault shall be generated and shall include:
 - faultcode: Client
 - faultstring: Invalid Request
 - and may provide further information in the details element.
- If there is no matching process identifier for the organization, a SOAP fault shall be generated and shall include:
 - faultcode: Client
 - faultstring: No Matching Process
- If the supplied process has previously completed or otherwise terminated, a SOAP fault shall be generated and shall include:
 - faultcode: Client
 - faultstring: Process Terminated

3.Y5.4.2 RetrieveActivitiesResponse Message

3.Y5.4.2.1 Trigger Events

The message is triggered by a ProcessStateManager receiving an RetrieveActivities request from a ProcessActivityExecutor.

3.Y5.4.2.2 Message Semantics

A list of zero or more process activities is returned. The format of each activity is a well-formed XML document of the type supported by the process of which they are part.

3.Y5.4.2.3 Expected Actions

The ProcessStateManager shall generate the list of activities based on the business rules of the system. If a SOAP fault is received then this fault should also be handled based on the business rules of the system.

3.Y5.5 Security Considerations

See section X.6.

3.Y5.6 Protocol Requirements

The RetrieveActivities request and response shall be transmitted using Synchronous Web Services Exchange, according to the requirements specified in ITI TF-2: Appendix V Web Services for IHE Transactions.

Table 3.Y5.6-1: WSDL Namespace Definitions

ihe	urn:ihe:qrph:rpe:2009
soap12	http://schemas.xmlsoap.org/wsdl/soap12/
wsaw	http://www.w3.org/2005/08/addressing
xsd	http://www.w3.org/2001/XMLSchema

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

- The following types shall be imported (xds:import) in the /definitions/types section:
namespace="urn:ihe:qrph:rpe:2009", schema="RPE.xsd"
- The /definitions/message/part/@element attribute of the RetrieveActivities request message shall be defined as: "ihe:RetrieveActivitiesRequest"
- The /definitions/message/part/@element attribute of the RetrieveActivities response message shall be defined as: "ihe:RetrieveActivitiesResponse"
- The /definitions/portType/operation/input/@wsaw:Action attribute for the RetrieveActivities request message shall be defined as "urn:ihe:qrph:2009:RetrieveActivities"
- The /definitions/portType/operation/output/@wsaw:Action attribute for the RetrieveActivities response message shall be defined as:
"urn:ihe:qrph:2009:RetrieveActivitiesResponse"
- The /definitions/binding/operation/soap12:operation/@soapAction attribute shall be defined as "urn:ihe:qrph:2009:RetrieveActivities"

These are the requirements that affect the wire format of the SOAP message. The other WSDL properties are only used within the WSDL definition and do not affect interoperability. Full sample request and response messages are in the section 3.Y5.6.1 Sample SOAP Messages.

3.Y5.6.1 Sample SOAP Messages

The samples in the following two sections show a typical request and corresponding response as contained in a SOAP Body. *Note the SOAP Header should be populated according to the IHE Appendix V: Web Services for IHE Transactions and in accord with section 3.Y5.5 Security Considerations.*

A full WSDL for the ProcessStateManager actor is found on the IHE FTP site at:
ftp://ftp.ihe.net/Quality/2010_2011_YR_4/QRPH%20Technical%20Committee/Maintenance%20work/RPE/

3.Y5.6.1.1 Sample RetrieveActivities SOAP Request

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

For task list display:

```
<soap:Body>
  <rpe:RetrieveActivitiesRequest xmlns:rpe="urn:ihe:qrph:rpe:2009">
    <rpe:input xmlns:hta="http://docs.oasis-open.org/ns/WS-BPEL4people/ws-humantask/api/200803"
      portType="hta:taskOperations" operation="getMyTaskDetails">
      <hta:getMyTaskDetails>
        <hta:taskType>ALL</hta:taskType>
        <hta:genericHumanRole>
          <hta:organizationalEntity>
            <hta:group>ehrId</hta:group>
          </hta:organizationalEntity>
        </hta:genericHumanRole>
        <hta:workQueue>studyId</hta:workQueue>
      </hta:getMyTaskDetails>
    </rpe:input>
  </rpe:RetrieveActivitiesRequest>
</soap:Body>
```

For task content display (using RFD rendering):

```
<soap:Body>
  <rpe:RetrieveActivitiesRequest xmlns:rpe="urn:ihe:qrph:rpe:2009">
    <rpe:input xmlns:hta="http://docs.oasis-open.org/ns/WS-BPEL4people/ws-humantask/api/200803"
      portType="hta:taskOperations" operation="getRendering">
      <hta:getRendering>
        <hta:identifier>taskId</hta:identifier>
        <hta:renderingType>"rpe:FormRenderingType"</hta:renderingType>
      </hta:getRendering>
    </rpe:input>
  </rpe:RetrieveActivitiesRequest>
</soap:Body>
```

3.Y5.6.1.2 Sample RetrieveActivities SOAP Response

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

For task list display:

```
<soap:Body>
  <rpe:RetrieveActivitiesResponse xmlns:rpe="urn:ihe:qrph:rpe:2009">
    <rpe:output>
      <hta:getMyTaskDetailsResponse xmlns:hta="http://docs.oasis-open.org/ns/WS-BPEL4people/ws-
        humantask/api/200803">
        <htt:taskDetails xmlns:htt="http://docs.oasis-open.org/ns/WS-BPEL4people/ws-
          humantask/types/200803">
          <htt:id>taskId</htt:id>
          <htt:taskType>TASK</htt:taskType>
          <htt:name> Physician_Baseline_Task</htt:name>
          <htt:status>READY</htt:status>
          ...
        </htt:taskDetails>
        ...
      </hta:getMyTaskDetailsResponse>
    </rpe:output>
  </rpe:RetrieveActivitiesResponse>
```

```
</soap:Body>
```

For individual task content display (using RFD rendering):

```
<soap:Body>
  <rpe:RetrieveActivitiesResponse xmlns:rpe="urn:ihe:qrph:rpe:2009">
    <rpe:output>
      <hta:getRenderingResponse xmlns:hta="http://docs.oasis-open.org/ns/WS-BPEL4people/ws-
        humantask/api/200803">
        <hta:rendering xmlns:rpe="urn:ihe:qrph:rpe:2009" type="rpe:FormRenderingType">
          <rpe:formRendering>
            <rpe:formIdentifier>"FM.PHYSICIANBASELINE"</rpe:formIdentifier>

            <rpe:formManagerURL>"http://www.company.com/services/FormManager_Port_Soap12"</rpe:formMana
              gerURL>
              </rpe:formRendering>
            </hta:rendering>
          </hta:getRenderingResponse>
        </rpe:output>
      </rpe:RetrieveActivitiesResponse>
    </soap:Body>
```

3.Y6 UpdateActivity

This section corresponds to transaction QRPH-Y6 of the IHE QRPH Transaction Framework. QRPH-Y6 is used by the ProcessStateManager and ProcessActivityExecutor actors.

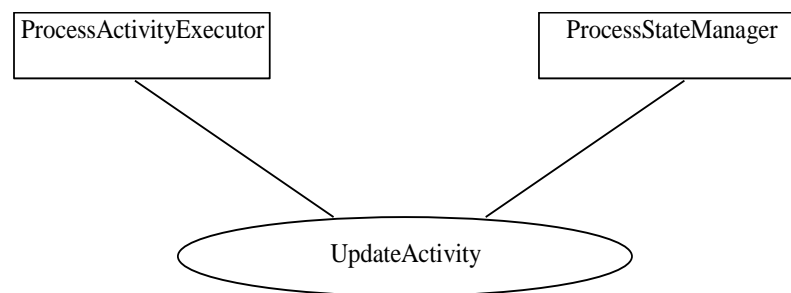
3.Y6.1 Scope

UpdateActivity allows a ProcessActivityExecutor to provide an update on activity's state or data to a ProcessStateManager for a process it is a participant in.

In UpdateActivity request the ProcessActivityExecutor can supply the following context:

- an organizational identifier
- a patient identifier
- the process instance identifier
- the activity or task instance identifier
- any other input data required by a custom or pass-thru operation implementation

3.Y6.2 Use Case Roles



Actor: ProcessActivityExecutor

Role: A system that knows how to execute activities that are part of a process.

Actor: ProcessStateManager

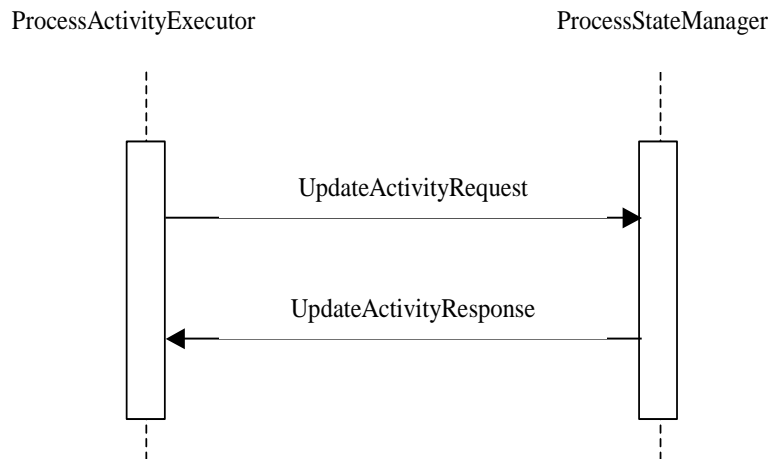
Role: A system that manages the runtime state of a process.

3.Y6.3 Referenced Standards

Implementers of this transaction shall comply with all requirements where applicable described in:

- ITI TF-2: Appendix V Web Services for IHE Transactions
- OASIS Web Services Business Process Execution Language (WS-BPEL) available at: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf>
- OASIS WS-BPEL Extension for People (BPEL4People) Specification Version 1.1 available at: <http://docs.oasis-open.org/bpel4people/bpel4people-1.1.pdf>
- OASIS Web Services – Human Task (WS-HumanTask) Specification Version 1.1 available at: <http://docs.oasis-open.org/bpel4people/ws-humantask-1.1.pdf>
- Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. <http://www.w3.org/TR/REC-xml>

3.Y6.4 Interaction Diagram



3.Y6.4.1 UpdateActivity Message

3.Y6.4.1.1 Trigger Events

The ProcessActivityExecutor, based upon human decision or application of a rule for automatic operation, wants to update the ProcessStateManager with the current state or data associated with an activity it is performing.

3.Y6.4.1.2 Message Semantics

The following parameters are specified for the body of this transaction.

Table 3.Y6.4.1.2–1: UpdateActivityRequest Parameters

Parameter Name	REQ	Description	Value
input	R	Used for supplying: an organizational identifier a patient identifier the process instance identifier the activity or task instance identifier any other input data required by a custom or pass-thru operation implementation Supply of an activity identifier is mandatory either in the context or input data sections.	A specialization of a complex XML element described in Table 3.Y1.4.1.2-2.

Section 3.Y6.6 describes the Web Services protocol requirements and the format of the message in full detail.

3.Y6.4.1.3 Expected Actions

The ProcessStateManager parses the UpdateActivity request and if well-formed, updates the given activity for the state and/or data supplied.

If the activity is completed as a result of this transaction the ProcessStateManager acknowledges the request with a response code of “ACTIVITY_COMPLETED”.

If the activity is updated but not completed as a result of this transaction the ProcessStateManager acknowledges the request with a response code of “ACTIVITY_UPDATED”.

Otherwise SOAP faults shall be generated accordingly:

- If the request is not well-formed, a SOAP fault shall be generated and shall include:
- faultcode: Client
- faultstring: Invalid Request
- and may provide further information in the details element.

- If there is no matching activity identifier, a SOAP fault shall be generated and shall include:
- faultcode: Client
- faultstring: No Matching Activity
- If the process failed to update the activity due to a ProcessStateManager system error, a SOAP fault shall be generated and shall include:
- faultcode: Server
- faultstring: System Error
- and may provide further information in the details element.

3.Y6.4.1 UpdateActivityResponse Message

3.Y6.4.2.1 Trigger Events

The message is triggered by a ProcessStateManager receiving an UpdateActivity request from a ProcessActivityExecutor.

3.Y6.4.2.2 Message Semantics

A responseCode XML element of type string is returned which confirms successful update of the activity with either a value of “ACTIVITY_COMPLETED” or “ACTIVITY_UPDATED”.

3.Y6.4.2.3 Expected Actions

The ProcessStateManager shall update the activity based on the business rules of the system. If a SOAP fault is received then this fault should also be handled based on the business rules of the system.

3.Y6.5 Security Considerations

See section X.6.

3.Y6.6 Protocol Requirements

The UpdateActivity request and response shall be transmitted using Synchronous Web Services Exchange, according to the requirements specified in ITI TF-2: Appendix V Web Services for IHE Transactions.

Table 3.Y6.6-1: WSDL Namespace Definitions

ihe	urn:ihe:qrph:rpe:2009
soap12	http://schemas.xmlsoap.org/wsdl/soap12/
wsaw	http://www.w3.org/2005/08/addressing
xsd	http://www.w3.org/2001/XMLSchema

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

- The following types shall be imported (xds:import) in the /definitions/types section:
namespace="urn:ihe:qrph:rpe:2009", schema="RPE.xsd"
- The /definitions/message/part/@element attribute of the UpdateActivity request message shall be defined as: "ihe:UpdateActivityRequest"
- The /definitions/message/part/@element attribute of the UpdateActivity response message shall be defined as: "ihe:UpdateActivityResponse"
- The /definitions/portType/operation/input/@wsaw:Action attribute for the UpdateActivity request message shall be defined as "urn:ihe:qrph:2009:UpdateActivity"
- The /definitions/portType/operation/output/@wsaw:Action attribute for the UpdateActivity response message shall be defined as: "urn:ihe:qrph:2009:UpdateActivityResponse"
- The /definitions/binding/operation/soap12:operation/@soapAction attribute shall be defined as "urn:ihe:qrph:2009:UpdateActivity"

These are the requirements that affect the wire format of the SOAP message. The other WSDL properties are only used within the WSDL definition and do not affect interoperability. Full sample request and response messages are in the section 3.Y6.6.1 Sample SOAP Messages.

3.Y6.6.1 Sample SOAP Messages

The samples in the following two sections show a typical request and corresponding response as contained in a SOAP Body. *Note the SOAP Header should be populated according to the IHE Appendix V: Web Services for IHE Transactions and in accord with section 3.Y6.5 Security Considerations.*

A full WSDL for the ProcessStateManager actor is found on the IHE FTP site at:

ftp://ftp.ihe.net/Quality/2010_2011_YR_4/QRPH%20Technical%20Committee/Maintenance%20work/RPE/

3.Y6.6.1.1 Sample UpdateActivity SOAP Request

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

HL7-V3 Example:

```
<soap:Body>
  <rpe:UpdateActivityRequest xmlns:rpe="urn:ihe:qrph:rpe:2009">
    <rpe:input xmlns:hta="http://docs.oasis-open.org/ns/WS-BPEL4people/ws-humantask/api/200803"
      portType="hta:taskOperations" operation="setOutput">
      <hta:setOutput>
        <hta:identifier>taskId</hta:identifier>
        <hta:taskData>
          <hl7:encounter xmlns:hl7="urn:hl7-org:v3" classCode="ENC" moodCode="INT">
            <hl7:id root="1.2.5.2.3.4" extension="ENC.PHYSICIANBASELINE"/>
          </hl7:encounter>
        </hta:taskData>
      </rpe:input>
    </rpe:UpdateActivityRequest>
  </soap:Body>
```

```
<hl7:code code="GENRL"/>
<hl7:text>PHYSICIAN BASELINE</hl7:text>
<!-- Substance admin, vitals, labs, etc. could be included here -->
...
</hl7:encounter>
</hta:taskData>
</hta:setOutput>
</rpe:input>
</rpe:UpdateActivityRequest>
</soap:Body>
```

3.Y6.6.1.2 Sample UpdateActivity SOAP Response

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

```
<soap:Body>
<rpe:RetrieveActivitiesResponse xmlns:rpe="urn:ihe:qrph:rpe:2009">
<rpe:responseCode>ACTIVITY_UPDATED</rpe:responseCode>
</rpe:RetrieveActivitiesResponse>
</soap:Body>
```

3.Y7 ReceiveProcessStateAlert

This section corresponds to transaction QRPH-Y7 of the IHE QRPH Transaction Framework. QRPH-Y7 is used by the ProcessStateManager and ProcessActivityExecutor actors.

3.Y7.1 Scope

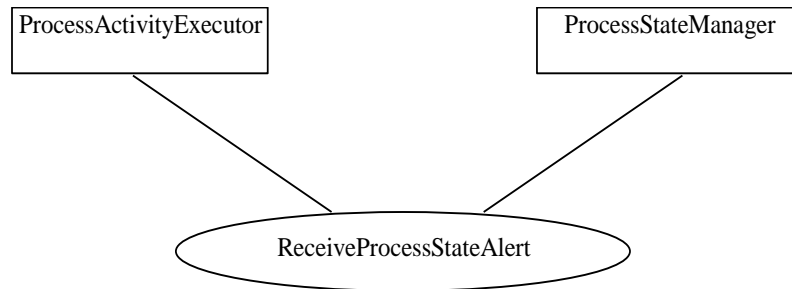
This transaction involves either:

- a ProcessActivityExecutor alerting a ProcessStateManager of a change that affects the state of a process it initiated, in which case the alert can contain:
- an organizational identifier
- a patient identifier
- the process identifier
- the process state change

OR:

- a ProcessStateManager alerting a ProcessActivityExecutor of a change associated with a process it initiated with the ProcessStateManager, in which case the alert contains:
- an organizational identifier
- a patient identifier
- a process definition identifier
- the process identifier
- the process state change

3.Y7.2 Use Case Roles



Actor: ProcessActivityExecutor

Role: A system that knows how to execute activities that are part of a process.

Actor: ProcessStateManager

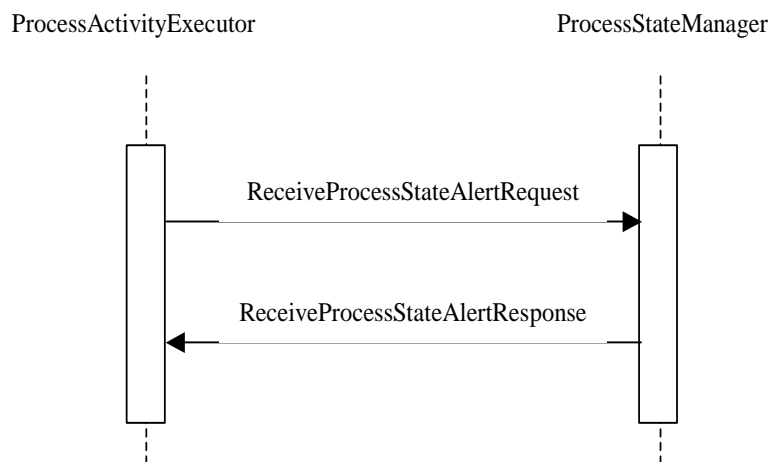
Role: A system that manages the runtime state of a process.

3.Y7.3 Referenced Standards

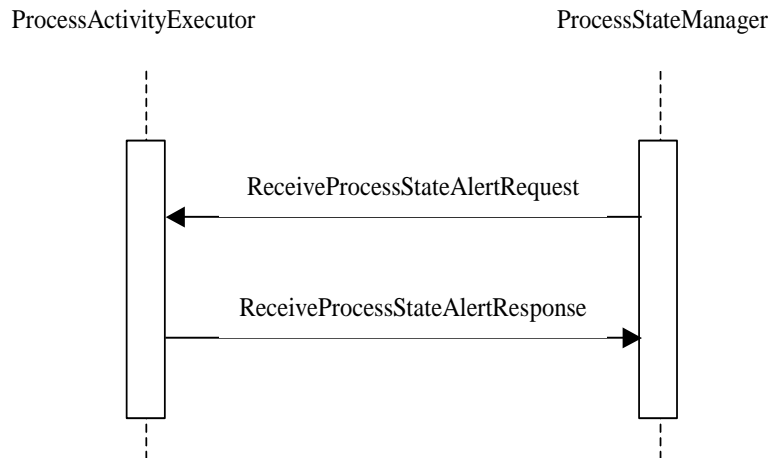
Implementers of this transaction shall comply with all requirements where applicable described in:

- ITI TF-2: Appendix V Web Services for IHE Transactions
- Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. <http://www.w3.org/TR/REC-xml>

3.Y7.4 Interaction Diagram



OR:



3.Y7.4.1 ReceiveProcessStateAlert Message

3.Y7.4.1.1 Trigger Events

Either:

- the ProcessActivityExecutor, based upon human decision or application of a rule for automatic operation, wants to alert the ProcessStateManager of a change that affects a process it initiated or,
- the ProcessStateManager, based upon human decision or application of a rule for automatic operation, wants to alert the ProcessActivityExecutor of a change associated with a process it initiated with the ProcessStateManager.

3.Y7.4.1.2 Message Semantics

The following parameters are specified for the body of this transaction.

Table 3.Y7.4.1.2–1: ReceiveProcessStateAlertRequest Parameters

Parameter Name	REQ	Description	Value
contextData	R	Used for supplying: an organizational identifier a patient identifier a process definition identifier a process identifier Supply of a process identifier is mandatory.	A specialization of a complex XML element described in Table 3.Y1.4.1.2-2.
processState	R	The process state change being alerted.	An XML attribute of type enumerated string with (current) possible values: PROCESS_ACTIVE – the process has been initiated and is active, e.g., patient

Parameter Name	REQ	Description	Value
			has been enrolled in a clinical study PROCESS_HOLD – the process has been suspended and is currently inactive, e.g., a hold has been placed on a clinical study PROCESS_EXITED – the process has been terminated prematurely, e.g., a clinical study has been discontinued PROCESS_COMPLETED – the process has completed as expected, e.g., a clinical study has concluded

Section 3.Y7.6 describes the Web Services protocol requirements and the format of the message in full detail.

3.Y7.4.1.3 Expected Actions

The ProcessActivityExecutor or ProcessStateManager parses the ReceiveProcessStateAlert request and if well-formed acknowledges the request with a response code of “PROCESS_STATE_ALERT_RECEIVED”.

Otherwise SOAP faults shall be generated accordingly:

- If request is not well-formed, a SOAP fault shall be generated and shall include:
- faultcode: Client
- faultstring: Invalid Request
- and may provide further information in the details element.

3.Y7.4.2 ReceiveProcessStateAlertResponse Message

3.Y7.4.2.1 Trigger Events

The message is triggered by a ProcessActivityExecutor or ProcessStateManager receiving a well-formed ReceiveProcessStateAlert request.

3.Y7.4.2.2 Message Semantics

A responseCode XML element of type string is returned to confirm receipt of the alert with a value of “PROCESS_STATE_ALERT_RECEIVED”.

3.Y7.4.2.3 Expected Actions

The ProcessStateManager shall generate the response code based on the business rules of the system. If a SOAP fault is received then this fault should also be handled based on the business rules of the system.

3.Y7.5 Security Considerations

See section X.6.

3.Y7.6 Protocol Requirements

The ReceiveProcessStateAlert request and response shall be transmitted using Synchronous Web Services Exchange, according to the requirements specified in ITI TF-2: Appendix V Web Services for IHE Transactions.

Table 3.Y7.6-1: WSDL Namespace Definitions

ihe	urn:ihe:qrph:rpe:2009
soap12	http://schemas.xmlsoap.org/wsdl/soap12/
wsaw	http://www.w3.org/2005/08/addressing
xsd	http://www.w3.org/2001/XMLSchema

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

- The following types shall be imported (xds:import) in the /definitions/types section:
- namespace="urn:ihe:qrph:rpe:2009", schema="RPE.xsd"
- The /definitions/message/part/@element attribute of the ReceiveProcessStateAlert request message shall be defined as: "ihe:ReceiveProcessStateAlertRequest"
- The /definitions/message/part/@element attribute of the ReceiveProcessStateAlert response message shall be defined as: "ihe:ReceiveProcessStateAlertResponse"
- The /definitions/portType/operation/input/@wsaw:Action attribute for the ReceiveProcessStateAlert request message shall be defined as "urn:ihe:qrph:2009:ReceiveProcessStateAlert"
- The /definitions/portType/operation/output/@wsaw:Action attribute for the ReceiveProcessStateAlert response message shall be defined as: "urn:ihe:qrph:2009:ReceiveProcessStateAlertResponse"
- The /definitions/binding/operation/soap12:operation/@soapAction attribute shall be defined as "urn:ihe:qrph:2009:ReceiveProcessStateAlert"

These are the requirements that affect the wire format of the SOAP message. The other WSDL properties are only used within the WSDL definition and do not affect interoperability. Full sample request and response messages are in the section 3.Y7.6.1 Sample SOAP Messages.

3.Y7.6.1 Sample SOAP Messages

The samples in the following two sections show a typical request and corresponding response as contained in a SOAP Body. *Note the SOAP Header should be populated according to the IHE*

Appendix V: Web Services for IHE Transactions and in accord with section 3.Y7.5 Security Considerations.

A full WSDL for the ProcessStateManager actor is found on the IHE FTP site at:
ftp://ftp.ihe.net/Quality/2010_2011_YR_4/QRPH%20Technical%20Committee/Maintenance%20work/RPE/

3.Y7.6.1.1 Sample ReceiveProcessStateAlert SOAP Request

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

ProcessStateManager to ProcessActivityExecutor (notification of patient enrolment):

```
<soap:Body>
  <rpe:ReceiveProcessStateAlertRequest xmlns:rpe="urn:ihe:qrph:rpe:2009">
    <rpe:contextData>
      <rpe:studySiteIdentifier>
        <hl7:id xmlns:hl7="urn:hl7-org:v3" root="1.2.3.44.2.3.44" extension="ehrId"/>
      </rpe:studySiteIdentifier>
      <rpe:candidateIdentifier>
        <hl7:id xmlns:hl7="urn:hl7-org:v3" root="1.2.3.44.2.3.44" extension="01000124"/>
      </rpe:candidateIdentifier>
      <rpe:subjectIdentifier>
        <hl7:id xmlns:hl7="urn:hl7-org:v3" root="1.2.5.2.3.4" extension="1006-0001-01000124"/>
      </rpe:subjectIdentifier>
      <rpe:studyDesignIdentifier>
        <hl7:id xmlns:hl7="urn:hl7-org:v3" root="1.2.5.2.3.4" extension="1006"/>
      </rpe:studyDesignIdentifier>
      <rpe:studyIdentifier>
        <hl7:id xmlns:hl7="urn:hl7-org:v3" root="1.2.5.2.3.4" extension="1006-0001"/>
      </rpe:studyIdentifier>
    </rpe:contextData>
    <rpe:processState>PROCESS_ACTIVE</rpe:processState>
  </rpe:ReceiveProcessStateAlertRequest>
</soap:Body>
```

ProcessActivityExecutor to ProcessStateManager (notification of patient withdrawal):

```
<soap:Body>
  <rpe:ReceiveProcessStateAlertRequest xmlns:rpe="urn:ihe:qrph:rpe:2009">
    <rpe:contextData>
      <rpe:studySiteIdentifier>ehrId</rpe:studySiteIdentifier>
      <rpe:subjectIdentifier>subjectId</rpe:subjectIdentifier>
      <rpe:studyIdentifier>studyId</rpe:studyIdentifier>
    </rpe:contextData>
    <rpe:processState>PROCESS_EXITED</rpe:processState>
  </rpe:ReceiveProcessStateAlertRequest>
</soap:Body>
```

3.Y7.6.1.2 Sample ReceiveProcessStateAlert SOAP Response

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

```
<soap:Body>
  <rpe:ReceiveProcessStateAlertResponse xmlns:rpe="urn:ihe:qrph:rpe:2009">
    <rpe:responseCode>PROCESS_STATE_ALERT_RECEIVED</rpe:responseCode>
  </rpe:ReceiveProcessStateAlertResponse>
</soap:Body>
```

```
</rpe:ReceiveProcessStateAlertResponse>  
</soap:Body>
```

3.Y8 InitiateActivity

This section corresponds to transaction QRPH-Y8 of the IHE QRPH Transaction Framework. QRPH-Y8 is used by the ProcessStateManager and ProcessActivityExecutor actors.

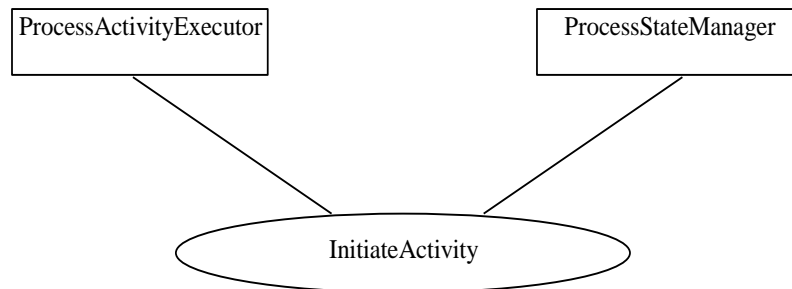
3.Y8.1 Scope

This transaction involves a ProcessStateManager requesting the creation and execution of an activity instance by a ProcessActivityExecutor.

The ProcessStateManager sends in the request to the ProcessActivityExecutor an activity definition that can contain:

- context information such as a process and patient identifiers
- presentation information such as a form identifier and form manager location
- input data specific to the task to be performed

3.Y8.2 Use Case Roles



Actor: ProcessActivityExecutor

Role: A system that knows how to execute activities that are part of a process.

Actor: ProcessStateManager

Role: A system that manages the runtime state of a process.

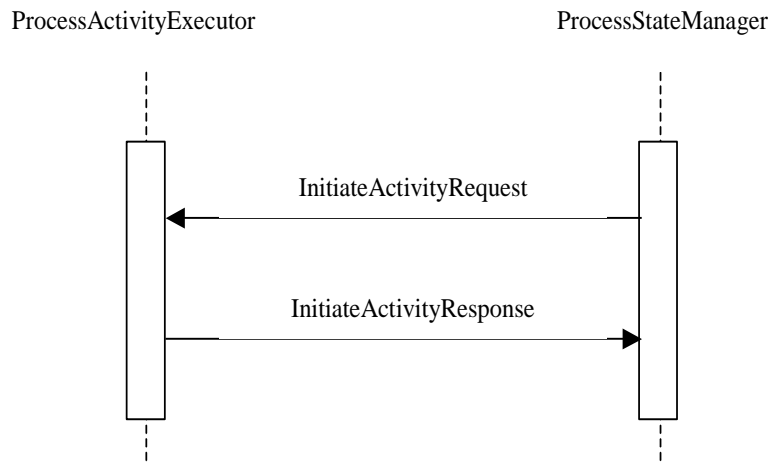
3.Y8.3 Referenced Standards

Implementers of this transaction shall comply with all requirements where applicable described in:

- ITI TF-2: Appendix V Web Services for IHE Transactions
- OASIS Web Services Business Process Execution Language (WS-BPEL) available at: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf>

- OASIS WS-BPEL Extension for People (BPEL4People) Specification Version 1.1 available at: <http://docs.oasis-open.org/bpel4people/bpel4people-1.1.pdf>
- OASIS Web Services – Human Task (WS-HumanTask) Specification Version 1.1 available at: <http://docs.oasis-open.org/bpel4people/ws-humantask-1.1.pdf>
- Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. <http://www.w3.org/TR/REC-xml>

3.Y8.4 Interaction Diagram



3.Y8.4.1 InitiateActivity Message

3.Y8.4.1.1 Trigger Events

The ProcessStateManager based upon human decision or the automated flow of process activities, needs to initiate activity creation and execution by a ProcessActivityExecutor.

3.Y8.4.1.2 Message Semantics

The following parameters are specified for the body of this transaction.

Table 3.Y8.4.1.2–1: InitiateActivityRequest Parameters

Parameter Name	REQ	Description	Value
activityDefinition	R	A flexible definition specification with sections for context, presentation and input data.	A complex XML element consisting of the following child elements.
context	O	Activity context which could include identifying properties like: an organizational identifier	An XML element of any type (e.g., rpe:contextData element).

		a patient identifier a process identifier	
presentation	O	Presentation information which could include an associated form id, form manager location or (prefetched) form URL.	An XML element of any type (e.g., rpe:formRendering element).
inputData	O	Specific task input data associated with the activity.	An XML element of any type.

Section 3.Y8.6 describes the Web Services protocol requirements and the format of the message in full detail.

3.Y8.4.1.3 Expected Actions

The ProcessActivityExecutor parses the InitiateActivity request and if well-formed, creates an instance of the activity in accord with the supplied definition and acknowledges the request with a response code of “ACTIVITY_INITIATED”.

Otherwise SOAP faults shall be generated accordingly:

- If request is not well-formed, a SOAP fault shall be generated and shall include:
 - faultcode: Client
 - faultstring: Invalid Request
 - and may provide further information in the details element.
- If the activity type is not supported, a SOAP fault shall be generated and shall include:
 - faultcode: Client
 - faultstring: Activity Type Not Supported
- If activity creation failed due to a ProcessActivityExecutor system error, a SOAP fault shall be generated and shall include:
 - faultcode: Server
 - faultstring: System Error
 - and may provide further information in the details element.

3.Y8.4.2 InitiateActivityResponse Message

3.Y8.4.2.1 Trigger Events

The message is triggered by a ProcessActivityExecutor receiving an InitiateActivity request from a ProcessStateManager and successfully creating an activity.

3.Y8.4.2.2 Message Semantics

A responseCode XML element of type string is returned which confirms initiation of the activity with a value of “ACTIVITY_INITIATED”.

3.Y8.4.2.3 Expected Actions

The ProcessActivityExecutor shall generate the response code based on the business rules of the system. If a SOAP fault is received then this fault should also be handled based on the business rules of the system.

3.Y8.5 Security Considerations

See section X.6.

3.Y8.6 Protocol Requirements

The InitiateActivity request and response shall be transmitted using Synchronous Web Services Exchange, according to the requirements specified in ITI TF-2: Appendix V Web Services for IHE Transactions.

Table 3.Y8.6-1: WSDL Namespace Definitions

ihe	urn:ihe:qrph:rpe:2009
soap12	http://schemas.xmlsoap.org/wsdl/soap12/
wsaw	http://www.w3.org/2005/08/addressing
xsd	http://www.w3.org/2001/XMLSchema

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

- The following types shall be imported (xds:import) in the /definitions/types section:
namespace="urn:ihe:qrph:rpe:2009", schema="RPE.xsd"
- The /definitions/message/part/@element attribute of the InitiateActivity request message shall be defined as: "ihe:InitiateActivityRequest"
- The /definitions/message/part/@element attribute of the InitiateActivity response message shall be defined as: "ihe:InitiateActivityResponse"
- The /definitions/portType/operation/input/@wsaw:Action attribute for the InitiateActivity request message shall be defined as "urn:ihe:qrph:2009:InitiateActivity"
- The /definitions/portType/operation/output/@wsaw:Action attribute for the InitiateActivity response message shall be defined as: "urn:ihe:qrph:2009:InitiateActivityResponse"
- The /definitions/binding/operation/soap12:operation/@soapAction attribute shall be defined as "urn:ihe:qrph:2009:InitiateActivity"

These are the requirements that affect the wire format of the SOAP message. The other WSDL properties are only used within the WSDL definition and do not affect interoperability. Full sample request and response messages are in the section 3.Y8.6.1 Sample SOAP Messages.

3.Y8.6.1 Sample SOAP Messages

The samples in the following two sections show a typical request and corresponding response as contained in a SOAP Body. *Note the SOAP Header should be populated according to the IHE Appendix V: Web Services for IHE Transactions and in accord with section 3.Y8.5 Security Considerations.*

A full WSDL for the ProcessStateManager actor is found on the IHE FTP site at:

ftp://ftp.ihe.net/Quality/2010_2011_YR_4/QRPH%20Technical%20Committee/Maintenance%20work/RPE/

3.Y8.6.1.1 Sample InitiateActivity SOAP Request

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

```
<soap:Body>
  <rpe:InitiateActivityRequest xmlns:rpe="urn:ihe:qrph:rpe:2009">
    <rpe:activityDefinition>
      <rpe:context>
        <htc:humanTaskRequestContext xmlns:htc="http://docs.oasis-open.org/ws-BPEL4people/ws-humantask/api/200803">
          <htc:priority>1</htc:priority>
          <htc:peopleAssignments>
            <htc:potentialOwners>
              <htc:from>
                <htc:literal>
                  <htc:organizationalEntity>
                    <htc:group>ehrId</htc:group>
                  </htc:organizationalEntity>
                </htc:literal>
              </htc:from>
            </htc:potentialOwners>
            <htc:taskInitiator>
              <htc:from>
                <htc:literal>
                  <htc:organizationalEntity>
                    <htc:group>sponsorId</htc:group>
                  </htc:organizationalEntity>
                </htc:literal>
              </htc:from>
            </htc:taskInitiator>
          </htc:peopleAssignments>
          <htc:isSkipable>no</htc:isSkipable>
          <htc:expirationTime>2010-12-31T12:00:00</htc:expirationTime>
          <htc:activationDeferralTime>2010-10-31T12:00:00</htc:activationDeferralTime>
        </rpe:contextData>
        <rpe:candidateIdentifier>patientId</rpe:candidateIdentifier>
        <rpe:subjectIdentifier>subjectId</rpe:subjectIdentifier>
        <rpe:studyIdentifier>studyId</rpe:studyIdentifier>
      </rpe:contextData>
    </htc:humanTaskRequestContext>
  </rpe:context>
  <rpe:presentation>
    <rpe:formRendering>
      <rpe:formIdentifier>"CDASH_FM_VS"</rpe:formIdentifier>
    </rpe:formRendering>
  </rpe:formManagerURL>"http://www.company.com/services/FormManager_Port_Soap12"</rpe:formManagerURL>
</soap:Body>
```

```
        </rpe:formRendering>
    </rpe:presentation>
    <rpe:inputData>
        <sdm:ActivityDef xmlns:sdm="http://www.cdisc.org/ns/studydesign/v1.0"
            Name="collection of vital signs 1" OID="ACTDEF_VITALSIGNS_01" Type="Data Collection">
            <Description>
                <TranslatedText xml:lang="en">Collection of Vital Signs and temperature: Pre-
                Ambulatory-ECG</TranslatedText>
            </Description>
            <FormRef FormOID="CDASH_FM_VS" Mandatory="Yes"/>
        </sdm:ActivityDef>
    </rpe:inputData>
</rpe:activityDefinition>
</rpe:InitiateActivityRequest>
</soap:Body>
```

3.Y8.6.1.2 Sample InitiateActivity SOAP Response

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

```
<soap:Body>
  <rpe:InitiateActivityResponse xmlns:rpe="urn:ihe:qrph:rpe:2009">
    <rpe:responseCode>ACTIVITY_INITIATED</rpe:responseCode>
  </rpe:InitiateActivityResponse>
</soap:Body>
```

Appendix A – RPE Motivations

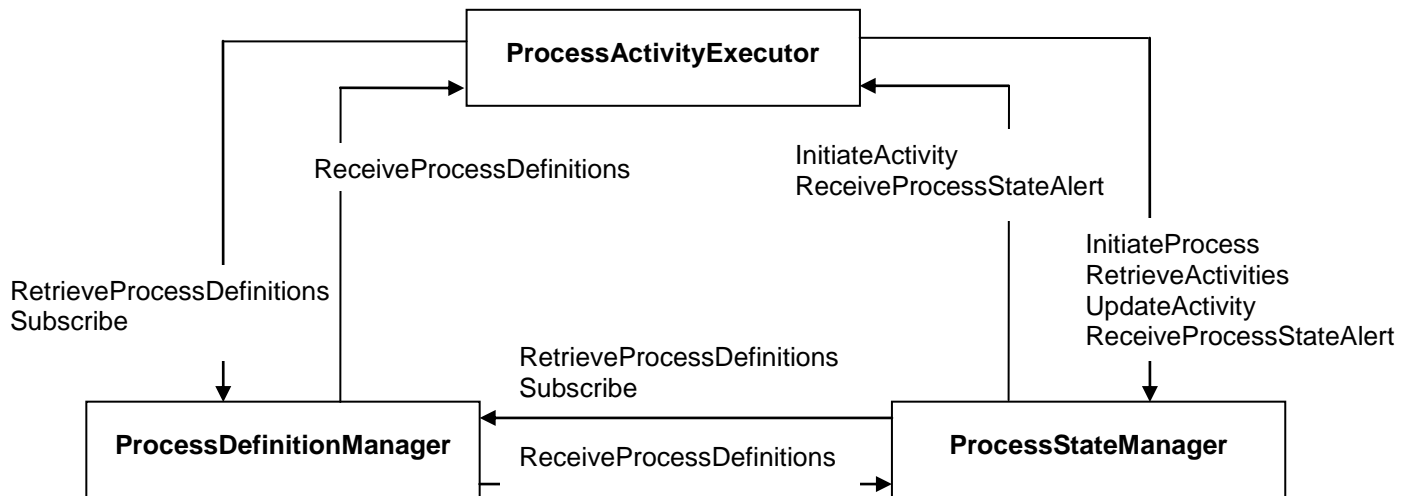
This appendix is the content of a technical note circulated within the IHE QRPH Technical Committee in April, 2010.

The key motivations behind the design of the new Retrieve *Process* for Execution transaction set are:

1. Provide support for *collaborative* process management
2. Be consistent with well-accepted process management architecture principles
3. Accommodate different process/task engine implementation interfaces
4. Support arbitrary process definition formats

1. Provide support for collaborative process management, i.e., bridging different IT system boundaries

Well-accepted IT standards do exist for business process management (BPM) architecture. However, these are primarily focused on processes managed within an organizational boundary. RPE defines a set of transactions for collaboration across IT system boundaries involving three main actors: the manager of process definitions, the manager of runtime processes and the performer of process activities:



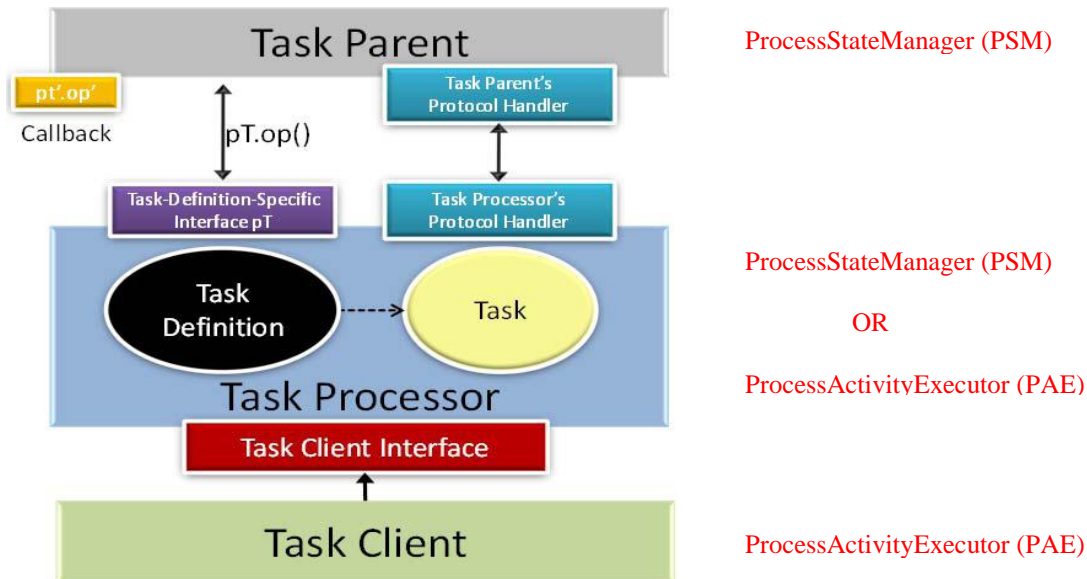
2. Be consistent with well-accepted process management architecture principles

BPM system design has a long history stretching back decades and RPE attempts to be consistent with the derived principles and latest standards in this area.

One design principle is that the management of processes and their constituent human task activities is best done by separate processor components, i.e., a process engine and a task processor. The fundamental reasons for this are:

- Runtime management of processes and their state has different requirements to the lifecycle management of component human activities, e.g., the definition of a specific human task may be reusable across different processes.
- An activity performer assigned one process’s tasks can also be concurrently assigned to perform tasks of another process, i.e., people can multi-task and serve different processes just as program services do.

As a result, task management infrastructure and task list client interfaces, have very different requirements to process management ones. Below is a picture of the relationship between the process engine (“Task Parent”), task processor and client with a correspondence to the RPE actors:



RPE acknowledges in the above that the PAE, i.e., typically an EHR, may want or need to play the role of task processor, having more internal control over the creation and presentation of task list activities. The table below presents the key RPE transactions and actors involved in the context of the above architecture:

RPE Transaction	Process Engine (PSM)	Task Processor (PSM or PAE)	Task Client (PAE)	WS-HumanTask Examples
InitiateProcess	X		X	n/a
RetrieveActivities		X	X	getMyTaskAbstracts, getMyTaskDetails...
UpdateActivity	X	X	X	setOutput, suspend, complete...,

RPE Transaction	Process Engine (PSM)	Task Processor (PSM or PAE)	Task Client (PAE)	WS-HumanTask Examples
				or, a user-defined, task-specific operation
InitiateActivity	X	X		a user-defined, task-specific operation
ReceiveProcessStateAlert	X	X	X	skippedOperation, exitOperation...

3. Accommodate different process/task engine implementation interfaces

The RPE transaction interfaces and message content are deliberately designed to be flexible, for while well accepted IT process management standards exist, their adoption is scant at this stage within HIT. Mandating full implementation support for instance, of the relatively new BPEL4People/WS-HumanTask 1.1 standards, while perhaps a future goal, would be premature at this stage.

For this purpose, the RPE schema type, InputType is defined with a flexible context argument and arbitrary input data element together with optional interface attributes. The latter – along the lines of BPEL4People and WS-HumanTask – allows for hand-off to a standard or custom task interface to handle the request. (This design also aligns with the wrapper service approach in IHE TF white paper, “A SOA View of IHE Profiles”.)

An example of InputType’s use in UpdateActivityRequest, here passing thru to the WS-HumanTask API:

```
<rpe:UpdateActivityRequest>
  <rpe:input xmlns:hta="http://docs.oasis-open.org/ns/WS-BPEL4people/ws-
humantask/api/200803"
    portType="hta:taskOperations" operation="setOutput">
    <hta:setOutput>
      <hta:identifier>taskId</hta:identifier>
      <hta:taskData>
        <h17:encounter xmlns:h17="urn:h17-org:v3" classCode="ENC"
moodCode="INT">
          <h17:id root="1.2.5.2.3.4"
extension="ENC.PHYSICIANBASELINE"/>
          <h17:code code="GENRL"/>
          <h17:text>PHYSICIAN BASELINE</h17:text>
          ...
        </h17:encounter>
      </hta:taskData>
    </hta:setOutput>
  </rpe:input>
</rpe:UpdateActivityRequest>
```

4. Support arbitrary process definition formats

RPE was originally developed in the context of multiple process definition standards for clinical study design. As the CDISC web site states (see <http://www.cdisc.org/study-trial-design>):

Two xml implementations of study design are currently under development, an extension [TDM] to ODM and an HL7 message. The first builds on the representation of study data collection definitions already in ODM, adding information on activities, their timing and workflow, as well as inclusion and exclusion criteria, study summary and high-level experimental design information. The HL7 Study Design message builds on existing HL7 messages and methodology for describing workflow, while adding representation of experimental design.

Neither of these approaches uses the dominant IT standard for process specification OASIS's WS-BPEL and related tasking standards, BPEL4People/WS-HumanTask. And neither design-time modeling approach is built for *runtime* deployment and execution of a process. Additionally, for other HIT domains, there are no uniform XML standards for representing process definitions and/or only beginning steps in this direction (e.g., trying again to twist HL7 CDA to this purpose).

Likewise, it has been previously noted that EHR's are still in the process of moving to standards-based process engine/task processor implementations.

For these reasons, while an argument could be made to mandate use of BPEL for process definition representation, this too was deemed premature and so the message format for process definition transactions allows any content.

Appendix B – Backwards Compatibility

In the first version of RPE known as Retrieve *Protocol* for Execution, aka RPE 1.0, only two transactions were required, implemented and tested at Connectathon 2010 and HIMSS10. The table below shows a mapping between transactions from RPE 1.0 and those in Retrieve Process for Execution, aka RPE 2.0.

RPE 1.0 Transaction	RPE 2.0 Transaction
RetrieveProtocolDefinition	RetrieveProcessDefinitions
EnterPatientRequest	InitiateProcess
EnrollPatientRequest	ReceiveProcessStateAlert
PatientScreeningVisitsScheduled	UpdateActivity
RecordPatientScreeningVisit	UpdateActivity
PatientStudyVisitsScheduled	UpdateActivity
RecordPatientStudyVisit	UpdateActivity
AmendProtocolDef	ReceiveProcessDefinitions
AlertProtocolState	ReceiveProcessStateAlert

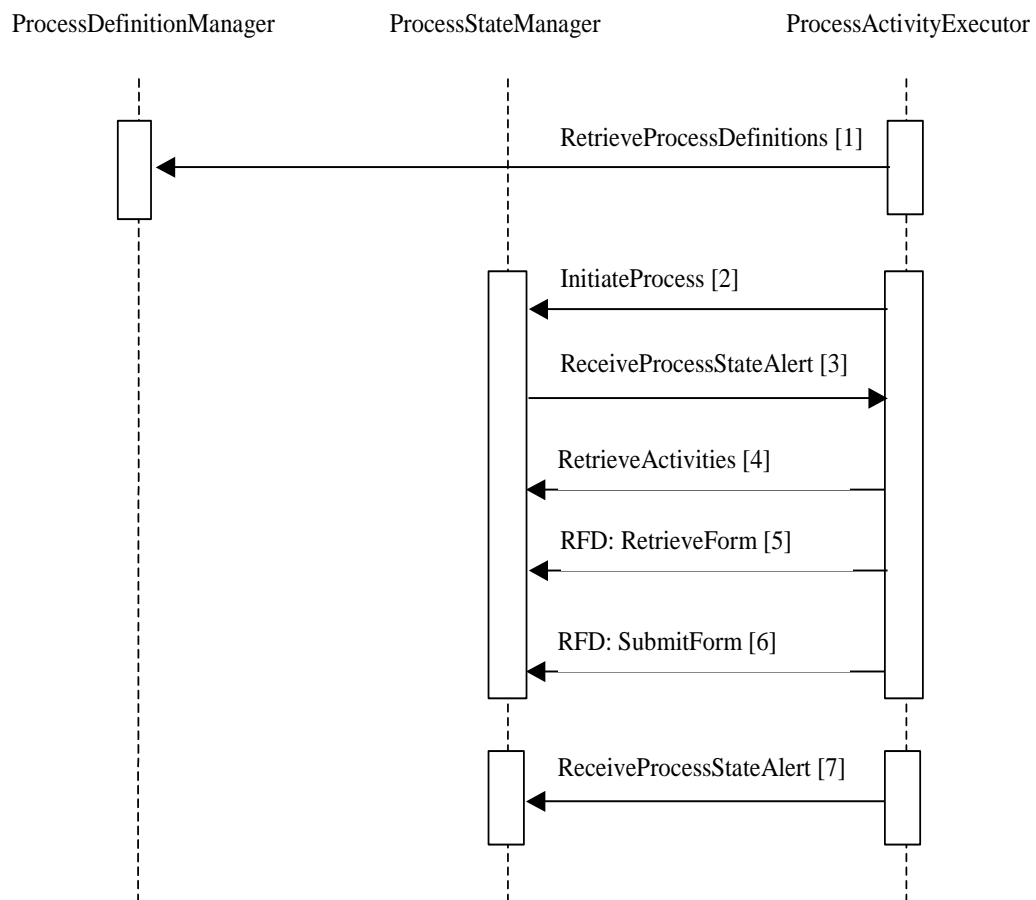
For more detail, please see the respective Sample SOAP Messages sections for RetrieveProcessDefinitions, InitiateProcess and ReceiveProcessStateAlert for how the functionality of RPE 1.0 required transactions, RetrieveProtocolDefinition and EnrollPatientRequest, is implemented. (*Note the examples in these Sample SOAP Messages sections use the test data from Connectathon 2010.*)

Appendix C – Informative Example with BPEL/WS-HumanTask

BPEL and WS-HumanTask have been adopted as the standard process definition and activity definition languages respectively for use in related RPE transactions. This does not necessarily compel use of BPEL and WS-HumanTask processors within a ProcessStateManager or ProcessActivityExecutor environment but it does:

- Assume a transformation exists from any variant process definition format being used (e.g., HL7 Study Design or CDISC ODM) and that such transformation is performed before/after use of the RPE transaction in question.
- Encourage native use of BPEL and WS-HumanTask behind RPE transaction implementation by the ProcessStateManager and ProcessActivityExecutor.

As an example of BPEL/WS-HumanTask use, we first revisit the RFD variant of the basic process flow for RPE and show how a study definition expressed in HL7 Study Design is transformed into an equivalent BPEL process definition conforming to this sequence of transactions:



RFD Process Flow in RPE Profile

1. The ProcessActivityExecutor retrieves process definitions of potential interest from the ProcessDefinitionManager.
2. The ProcessActivityExecutor request's process initiation by the ProcessStateManager forwarding a given process definition identifier as well as other required data, e.g., a patient identifier, demographics or eligibility criteria.
3. The ProcessStateManager notifies the ProcessActivityExecutor that the process is proceeding or otherwise.
4. The ProcessActivityExecutor retrieves the current activity(ies) it has to perform with the activity created and managed by the ProcessStateManager. The retrieved activity data contains an associated RFD form identifier.
5. The ProcessActivityExecutor retrieves the form from an RFD Form Manager (whose location may already be known and/or encapsulated within the retrieved activity data).
6. The ProcessActivityExecutor fills out and submits the form with data collected from its execution of the activity. This is equivalent to the role of UpdateActivity and as for the latter form submission may be used for activity data update and/or activity completion.
7. The ProcessActivityExecutor can always notify the ProcessStateManager of unscheduled events that may affect the process state, e.g., a patient withdrawal from a clinical trial.

The table below (an XSL transformation of the underlying HL7 Study Design XML) describes a simple clinical trial example (based on a RPE scenario developed for HIMSS10).

Clinical Trial Title

RPE HL7 Study Design Example 1

Description

This is the full description of the study

Eligibility Criteria

Male	
Age	>=30 and <=60years
BP Diastolic	>90 mmHg
BP Systolic	>140 mmHg
HDLc SerPl-mCnc	<40 mg/dL
Hgb A1c Fr Bld	>6.9 %

Encounters

ENC.PHYSICIANBASELINE	Physician: Baseline	1 day(s)
ENC.PHYSICIAN6MONTHFUP	Physician: 6 Month Follow-up	6 month(s)
ENC.PHYSICIAN12MONTHFUP	Physician: 12 Month Follow-up	12 month(s)

Below is the underlying HL7 Study Design XML representation consisting of a single epoch (“EP1.PEROS”) and a single arm (“AR1.ENCSEQ”) comprised of a sequence of six month-spaced encounters (“ENC.PHYSICIANBASELINE”, “ENC. PHYSICIAN6MONTHFUP”, “ENC.PHYSICIAN12MONTHFUP”):

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="xsl/HL7PlannedStudy.xsl"?>
<plannedStudy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:hl7-org:v3" ITSVersion="XML_1.0"
  xsi:schemaLocation="urn:hl7-org:v3 ../xsd/HL7/StudyDesign.xsd" classCode="CLNTRL"
  moodCode="DEF">
  <id root="1.2.5.2.3.4" extension="RPEHL7StudyDesignEx1"/>
  <title>RPE HL7 Study Design Example 1</title>
  <text>This is the full description of the study</text>
  <precondition typeCode="PRCN" contextControlCode="AN">
    <eligibilityCriterion classCode="OBS" moodCode="CRT">
      <id root="1.2.5.2.3.4" extension="GENDER"/>
      <code code="M" codeSystem="2.16.840.1.113883.1.11.1" codeSystemName="Administrative
Gender Code"
        displayName="Male">
          <originalText>Male</originalText>
        </code>
        <statusCode code="active"/>
      </eligibilityCriterion>
    </precondition>
    <precondition typeCode="PRCN" contextControlCode="AN">
      <eligibilityCriterion classCode="OBS" moodCode="CRT">
        <id root="1.2.5.2.3.4" extension="AGE"/>
        <code code="424144002" codeSystem="2.16.840.1.113883.6.96" displayName="Age">
          <originalText>Age</originalText>
        </code>
        <statusCode code="active"/>
        <value xsi:type="IVL_PQ">
          <low value="30" unit="a" inclusive="true"/>
          <high value="60" unit="a" inclusive="true"/>
        </value>
      </eligibilityCriterion>
    </precondition>
    <precondition typeCode="PRCN" contextControlCode="AN">
      <eligibilityCriterion classCode="OBS" moodCode="CRT">
        <id root="1.2.5.2.3.4" extension="DBP"/>
        <code code="8462-4" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC"
displayName="BP Diastolic">
          <originalText>Diastolic Blood Pressure</originalText>
        </code>
        <statusCode code="active"/>
        <value xsi:type="IVL_PQ">
          <low value="90" unit="mmHg" inclusive="false"/>
          <high nullFlavor="PINF"/>
        </value>
      </eligibilityCriterion>
    </precondition>
    <precondition typeCode="PRCN" contextControlCode="AN">
      <eligibilityCriterion classCode="OBS" moodCode="CRT">
        <id root="1.2.5.2.3.4" extension="SBP"/>
        <code code="8480-6" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC"
displayName="BP Systolic">
          <originalText>Systolic Blood Pressure</originalText>
        </code>
        <statusCode code="active"/>
        <value xsi:type="IVL_PQ">
          <low value="140" unit="mmHg" inclusive="false"/>
          <high nullFlavor="PINF"/>
        </value>
      </eligibilityCriterion>
    </precondition>
    <precondition typeCode="PRCN" contextControlCode="AN">
      <eligibilityCriterion classCode="OBS" moodCode="CRT">
        <id root="1.2.5.2.3.4" extension="HDL"/>
        <code code="2085-9" codeSystemName="LOINC" displayName="HDLc SerPl-mCnc">
          <originalText>HDL</originalText>
        </code>

```

```

        <statusCode code="active"/>
        <value xsi:type="IVL_PQ">
            <low nullFlavor="PINF"/>
            <high value="40" unit="mg/dL" inclusive="false"/>
        </value>
    </eligibilityCriterion>
</precondition>
<precondition typeCode="PRCN" contextControlCode="AN">
    <eligibilityCriterion classCode="OBS" moodCode="CRT">
        <id root="1.2.5.2.3.4" extension="HGBA1C"/>
        <code code="4548-4" codeSystemName="LOINC" displayName="Hgb Alc Fr Bld">
            <originalText>HgbAlc</originalText>
        </code>
        <text>HgbAlc</text>
        <statusCode code="active"/>
        <value xsi:type="IVL_PQ">
            <low value="6.9" unit="%" inclusive="false"/>
            <high nullFlavor="PINF"/>
        </value>
    </eligibilityCriterion>
</precondition>
<component1 typeCode="COMP">
    <sequenceNumber value="1"/>
    <epoch classCode="ACT" moodCode="DEF">
        <id root="1.2.5.2.3.4" extension="EP1.PEROS"/>
        <title>Subject PE and ROS</title>
    </epoch>
</component1>
<component2 typeCode="COMP">
    <arm classCode="ACT" moodCode="DEF">
        <id root="1.2.5.2.3.4" extension="AR1.ENCSEQ"/>
    </arm>
</component2>
<component4 typeCode="COMP">
    <timePointEventDefinition classCode="CTTEVENT" moodCode="DEF">
        <id root="SD" extension="SD01"/>
        <code code="STUDY"/>
        <component1 typeCode="COMP">
            <sequenceNumber value="1"/>
            <timePointEventDefinition classCode="CTTEVENT" moodCode="DEF">
                <id root="SD" extension="SD01-T1"/>
                <code code="CELL"/>
                <precondition typeCode="PRCN">
                    <timePointEventCriterion classCode="ACT" moodCode="CRT">
                        <code code="BL.PHYSICIANSTUDYDISC" displayName="Study
Discontinued"/>
                        <value xsi:type="BL">>false</value>
                    </timePointEventCriterion>
                </precondition>
            <component2 typeCode="COMP">
                <sequenceNumber value="0"/>
                <pauseQuantity value="1" unit="d"/>
                <encounter classCode="ENC" moodCode="INT">
                    <id root="1.2.5.2.3.4" extension="ENC.PHYSICIANBASELINE"/>
                    <code code="visit"/>
                    <text>Physician Baseline</text>
                </encounter>
            </component2>
            <subjectOf typeCode="SUBJ">
                <timePointEventCharacteristic classCode="OBS" moodCode="EVN">
                    <id root="1.2.5.2.3.4" extension="FM.FORMID"/>
                    <value xsi:type="ST">FM.PHYSICIANBASELINE</value>
                </timePointEventCharacteristic>
            </subjectOf>
        </timePointEventDefinition>
    </component1>
</component1 typeCode="COMP">

```

```

        <sequenceNumber value="2"/>
        <timePointEventDefinition classCode="CTTEVENT" moodCode="DEF">
          <id root="SD" extension="SD01-T2"/>
          <code code="CELL"/>
          <precondition typeCode="PRCN">
            <timePointEventCriterion classCode="ACT" moodCode="CRT">
              <code code="BL.PHYSICIANSTUDYDISC" displayName="Study
Discontinued"/>
              <value xsi:type="BL">>false</value>
            </timePointEventCriterion>
          </precondition>
          <component2 typeCode="COMP">
            <sequenceNumber value="0"/>
            <pauseQuantity value="6" unit="m"/>
            <encounter classCode="ENC" moodCode="INT">
              <id root="1.2.5.2.3.4" extension="ENC.PHYSICIAN6MONTHFUP"/>
              <code code="visit"/>
              <text>Physician 6 Month Follow-up</text>
            </encounter>
          </component2>
          <subjectOf typeCode="SUBJ">
            <timePointEventCharacteristic classCode="OBS" moodCode="EVN">
              <id root="1.2.5.2.3.4" extension="FM.FORMID"/>
              <value xsi:type="ST">FM.PHYSICIAN6MONTHFUP</value>
            </timePointEventCharacteristic>
          </subjectOf>
        </timePointEventDefinition>
      </component1>
      <component1 typeCode="COMP">
        <sequenceNumber value="3"/>
        <timePointEventDefinition classCode="CTTEVENT" moodCode="DEF">
          <id root="SD" extension="SD01-T3"/>
          <code code="CELL"/>
          <precondition typeCode="PRCN">
            <timePointEventCriterion classCode="ACT" moodCode="CRT">
              <code code="BL.PHYSICIANSTUDYDISC" displayName="Study
Discontinued"/>
              <value xsi:type="BL">>false</value>
            </timePointEventCriterion>
          </precondition>
          <component2 typeCode="COMP">
            <sequenceNumber value="0"/>
            <pauseQuantity value="12" unit="m"/>
            <encounter classCode="ENC" moodCode="INT">
              <id root="1.2.5.2.3.4" extension="ENC.PHYSICIAN12MONTHFUP"/>
              <code code="visit"/>
              <text>Physician 12 Month Follow-up</text>
            </encounter>
          </component2>
          <subjectOf typeCode="SUBJ">
            <timePointEventCharacteristic classCode="OBS" moodCode="EVN">
              <id root="1.2.5.2.3.4" extension="FM.FORMID"/>
              <value xsi:type="ST">FM.PHYSICIAN12MONTHFUP</value>
            </timePointEventCharacteristic>
          </subjectOf>
        </timePointEventDefinition>
      </component1>
      <componentOf1 typeCode="COMP">
        <epochStub classCode="ACT" moodCode="DEF">
          <id root="1.2.5.2.3.4" extension="PEROS"/>
        </epochStub>
      </componentOf1>
      <componentOf2 typeCode="COMP">
        <armStub classCode="ACT" moodCode="DEF">
          <id root="1.2.5.2.3.4" extension="ENCOUNTER"/>
        </armStub>
      </componentOf2>

```

```
        </timePointEventDefinition>
    </component4>
    <subjectOf typeCode="SUBJ">
        <studyCharacteristic classCode="OBS" moodCode="EVN">
            <code code="DESIGN">
                <originalText>Description of Trial Design</originalText>
            </code>
            <value xsi:type="ST">Metabolic Syndrome Study</value>
        </studyCharacteristic>
    </subjectOf>
    <subjectOf typeCode="SUBJ">
        <studyCharacteristic classCode="OBS" moodCode="EVN">
            <code code="LENGTH">
                <originalText>Trial Length</originalText>
            </code>
            <value xsi:type="PQ" unit="m" value="12"/>
        </studyCharacteristic>
    </subjectOf>
    <subjectOf typeCode="SUBJ">
        <studyCharacteristic classCode="OBS" moodCode="EVN">
            <code code="RFD.FORMMANAGER.URL">
                <originalText>RFD Form Manager URL</originalText>
            </code>
            <value
xsi:type="URL">http://www.company.com/services/FormManager_Port_Soap12</value>
        </studyCharacteristic>
    </subjectOf>
</plannedStudy>
```

The following is a transformation of the above representation into the equivalent BPEL/WS-HumanTask form:

```

<?xml version="1.0" encoding="UTF-8"?>
<bpel:process xmlns:tns="http://www.company.com/rpe" xmlns:rpe="urn:ihe:qrph:rpe:2009"
xmlns:h17="urn:h17-org:v3" xmlns:bpel="http://docs.oasis-
open.org/wsbpel/2.0/process/executable"
xmlns:b4p="http://docs.oasis-open.org/ns/bpel4people/bpel4people/200803"
xmlns:htd="http://docs.oasis-open.org/ns/bpel4people/ws-humantask/200803"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:bpelx="http://schemas.oracle.com/bpel/extension">
  <partnerLinks>
    <partnerLink name="ProcessActivityExecutor"
partnerLinkType="rpe:ProcessActivityExecutor_PortType_PL"
myRole="ProtocolStateManager_PortType_Role"/>
  </partnerLinks>
  <b4p:humanInteractions>
    <htd:logicalPeopleGroups>
      <htd:logicalPeopleGroup name="ehrId"/>
    </htd:logicalPeopleGroups>
  </b4p:humanInteractions>
  <bpel:extensions>
    <bpel:extension mustUnderstand="yes" namespace="http://docs.oasis-
open.org/ns/bpel4people/bpel4people/200803"/>
    <bpel:extension mustUnderstand="yes" namespace="http://docs.oasis-
open.org/ns/bpel4people/ws-humantask/200803"/>
  </bpel:extensions>
  <bpel:variables>
    <bpel:variable element="h17:eligibilityCriterion" name="GENDER"/>
    <bpel:variable element="h17:eligibilityCriterion" name="AGE"/>
    <bpel:variable element="h17:eligibilityCriterion" name="DBP"/>
    <bpel:variable element="h17:eligibilityCriterion" name="SBP"/>
    <bpel:variable element="h17:eligibilityCriterion" name="HDL"/>
    <bpel:variable element="h17:eligibilityCriterion" name="HGBA1C"/>
    <bpel:variable element="h17:encounter" name="ENC.PHYSICIANBASELINE"/>
    <bpel:variable element="h17:encounter" name="ENC.PHYSICIAN6MONTHFUP"/>
    <bpel:variable element="h17:encounter" name="ENC.PHYSICIAN12MONTHFUP"/>
    <bpel:variable element="xsd:boolean" name="BL.PHYSICIANSTUDYDISC"/>
    <bpel:variable element="rpe:patientData" name="PatientDataInput"/>
    <bpel:variable element="rpe:patientData" name="PatientDataOutput"/>
    <bpel:variable element="rpe:InitiateProcessRequest" name="InitiateProcessRequest"/>
    <bpel:variable element="rpe:ReceiveProcessStateAlertRequest"
name="ReceiveProcessStateAlertRequest"/>
    <bpel:variable element="rpe:ReceiveProcessStateAlertResponse"
name="ReceiveProcessStateAlertResponse"/>
  </bpel:variables>
  <bpel:sequence>
    <!-- Start of clinical trial process for a patient -->
    <!-- ProcessStateManager.InitiateProcess -->
    <bpel:receive createInstance="yes" name="Enter_Patient_Request"
partnerLink="ProcessActivityExecutor" variable="InitiateProcessRequest"/>
    <bpel:scope name="EP1.PEROS">
      <!-- Screening, assignment and enrollment activities would be specified in here-->
      <bpel:sequence>
        <bpel:if name="Preconditions">
          <bpel:condition>$PatientDataInput/gender = $GENDER/code/@code and
$PatientDataInput/age >= $AGE/value/low/@value and $PatientDataInput/age >=
$AGE/value/high/@value
                                $PatientDataInput/dbp >= $DBP/value/low/@value and
$PatientDataInput/sbp >= $SBP/value/low/@value and $PatientDataInput/hdl >=
$HDL/value/low/@value and
                                $PatientDataInput/hgbalc >=
$HGBA1C/value/low/@value</bpel:condition>
          <bpel:then>
            <bpel:assign>
              <bpel:copy>
                <bpel:from>'true'</bpel:from>
                <bpel:to variable="BL.ELIGIBLE"/>
              </bpel:copy>
            </bpel:assign>
          </bpel:then>
        </bpel:if>
      </bpel:sequence>
    </bpel:scope>
  </bpel:sequence>

```



```

        </bpel:then>
    </bpel:if>
    <bpel:if name="Eligible">
        <bpel:condition>$BL.ELIGIBLE = 'false'</bpel:condition>
        <!-- ProcessActivityExecutor.ReceiveProcessStateAlert -->
        <bpel:invoke name="Notify_InEligible" partnerLink="ProcessActivityExecutor"
operation="ProcessActivityExecutor_ReceiveProcessStateAlert"
inputVariable="ReceiveProcessStateAlertRequest"
        outputVariable="ReceiveProcessStateAlertResponse"/>
        <terminate name="Study_Discontinuation_or_End"/>
    </bpel:if>
    <!-- ProcessActivityExecutor.ReceiveProcessStateAlert -->
    <bpel:invoke name="Notify_Eligible" partnerLink="ProcessActivityExecutor"
operation="ProcessActivityExecutor_ReceiveProcessStateAlert"
inputVariable="ReceiveProcessStateAlertRequest"
        outputVariable="ReceiveProcessStateAlertResponse"/>
    </bpel:sequence>
    <!-- Treatment -->
    <bpel:scope name="AR1.ENCSEQ">
        <bpel:sequence>
            <bpel:extensionActivity>
                <b4p:peopleActivity inputVariable="PatientDataInput"
name="Physician_Baseline" outputVariable="PatientDataOutput">
                    <htd:task name="Physician_Baseline_Task">
                        <htd:peopleAssignments>
                            <htd:potentialOwners>
                                <htd:from logicalPeopleGroup="ehrId"/>
                            </htd:potentialOwners>
                        </htd:peopleAssignments>
                        <htd:renderings>
                            <htd:rendering xmlns:rpe="urn:ihe:qrph:rpe:2009"
type="rpe:FormRenderingType">
                                <rpe:formRendering>

<rpe:formIdentifier>"FM.PHYSICIANBASELINE"</rpe:formIdentifier>

<rpe:formManagerURL>"http://www.company.com/services/FormManager_Port_Soap12"</rpe:formManagerU
RL>

                                </rpe:formRendering>
                            </htd:rendering>
                        </htd:renderings>
                    </htd:task>
                </b4p:peopleActivity>
            </bpel:extensionActivity>
            <bpel:if name="Study_Discontinued">
                <bpel:condition>$BL.PHYSICIANSTUDYDISC = 'true'</bpel:condition>
                <terminate name="Study_Discontinued"/>
            </bpel:if>
            <bpel:extensionActivity>
                <b4p:peopleActivity inputVariable="PatientDataInput"
name="Physician_6_Month_Followup" outputVariable="PatientDataOutput">
                    <htd:task name="Physician_Baseline_Task">
                        <htd:peopleAssignments>
                            <htd:potentialOwners>
                                <htd:from logicalPeopleGroup="ehrId"/>
                            </htd:potentialOwners>
                        </htd:peopleAssignments>
                        <htd:renderings>
                            <htd:rendering xmlns:rpe="urn:ihe:qrph:rpe:2009"
type="rpe:FormRenderingType">
                                <rpe:formRendering>

<rpe:formIdentifier>"FM.PHYSICIAN6MONTHFUP"</rpe:formIdentifier>

<rpe:formManagerURL>"http://www.company.com/services/FormManager_Port_Soap12"</rpe:formManagerU
RL>

                                </rpe:formRendering>
                            </htd:rendering>
                        </htd:renderings>
                    </htd:task>
                </b4p:peopleActivity>
            </bpel:extensionActivity>
        </bpel:sequence>
    </bpel:scope>

```

```

        </htd:rendering>
    </htd:renderings>
</htd:task>
<b4p:scheduledActions>
    <b4p:deferActivation>
        <b4p:for>'PT180M'</b4p:for>
    </b4p:deferActivation>
</b4p:scheduledActions>
</b4p:peopleActivity>
</bpel:extensionActivity>
<bpel:if name="Study_Discontinued">
    <bpel:condition>$BL.PHYSICIANSTUDYDISC = 'true'</bpel:condition>
    <terminate name="Study_Discontinued"/>
</bpel:if>
<bpel:extensionActivity>
    <b4p:peopleActivity inputVariable="PatientDataInput"
name="Physician_12_Month_Followup" outputVariable="PatientDataOutput">
        <htd:task name="Physician_Baseline_Task">
            <htd:peopleAssignments>
                <htd:potentialOwners>
                    <htd:from logicalPeopleGroup="ehrId"/>
                </htd:potentialOwners>
            </htd:peopleAssignments>
            <htd:renderings>
                <htd:rendering xmlns:rpe="urn:ihe:qrph:rpe:2009"
type="rpe:FormRenderingType">
                    <rpe:formRendering>

<rpe:formIdentifier>"FM.PHYSICIAN12MONTHFUP"</rpe:formIdentifier>

<rpe:formManagerURL>"http://www.company.com/services/FormManager_Port_Soap12"</rpe:formManagerU
RL>

                </rpe:formRendering>
            </htd:rendering>
        </htd:renderings>
    </htd:task>
</b4p:scheduledActions>
    <b4p:deferActivation>
        <b4p:for>'PT180M'</b4p:for>
    </b4p:deferActivation>
</b4p:scheduledActions>
</b4p:peopleActivity>
</bpel:extensionActivity>
</bpel:sequence>
</bpel:scope>
</bpel:scope>
</bpel:sequence>
</bpel:process>

```