Collaborative Business Process Support in IHE XDS through ebXML Business Processes

Asuman Dogac Veli Bicer Alper Okcan

Software Research and Development Center
Middle East Technical University (METU)
06531 Ankara Turkiye
email: asuman@srdc.metu.edu.tr

Abstract

Currently, clinical information is stored in all kinds of proprietary formats through a multitude of medical information systems available on the market. This results in a severe interoperability problem in sharing electronic healthcare records.

To address this problem, an industry initiative, called "Integrating Healthcare Enterprise (IHE)" has specified the "Cross Enterprise Document Sharing (XDS)" Profile to store healthcare documents in an ebXML registry/repository to facilitate their sharing.

Through a separate effort, IHE has also defined inter-departmental Workflow Profiles to identify the transactions required to integrate information flow among several information systems. Although the clinical documents stored in XDS registries are obtained as a result of executing these workflows, IHE has not yet specified collaborative healthcare processes for the XDS. Hence, there is no way to track the workflows in XDS and the clinical documents produced through the workflows are manually inserted into the registry/repository.

Given that IHE XDS is using the ebXML architecture, the most natural way to integrate IHE Workflow Profiles to IHE XDS is using ebXML Business Processes (ebBP). In this paper, we describe the implementation of an enhanced IHE architecture demonstrating how ebXML Business Processes, IHE Workflow Profiles and the IHE XDS architecture can all be integrated to provide collaborative business process support in the healthcare domain.

1. Introduction

In order to address the interoperability problem in the healthcare domain, an important industry initiative called "Integrating the Healthcare Enterprise (IHE)" [16] has defined "Integration Profiles" which specify the interactions between various healthcare applications to realize some real life scenarios. IHE has also specified the interfaces of healthcare applications and the messages exchanged by restricting the well known standards such as HL7 [14] and DICOM [7].

The IHE integration profiles consist of two categories: inter-departmental integration profiles which aim the interoperation of various departments within a healthcare institute and more recently, inter-institutional integration profiles known as "Cross Enterprise Document Sharing (XDS)" Profile [17].

IHE, through inter-departmental Integration Profiles, identifies the transactions required to integrate information flow among several information systems and organizes them into coarse grained workflows such as "Scheduled Workflow". Then for each transaction, a single standard such as HL7 messages is selected to implement the transaction.

The basic idea of IHE XDS, on the other hand, is to store healthcare documents in ebXML registry/repository [12] architecture to facilitate their sharing among departments within a healthcare institute or among different healthcare enterprises. IHE XDS also specifies metadata for the documents which are stored through ebXML registry constructs to facilitate their discovery.

One of the main problems in IHE XDS is that, although the documents stored in the ebXML registry are produced mostly by IHE inter-departmental Integration Profiles, these processes are not considered in the XDS specification yet. Therefore, it is not possible to track the flow of documents and also all the documents and their metadata need to be manually inserted into the registry.

In this paper, we describe the implementation of an enhanced IHE architecture, realized within the scope of the

Artemis project [5], demonstrating how ebXML Business Process Specification schema, IHE Integration Profiles and IHE XDS architecture can all be integrated to provide collaborative business process support for IHE XDS.

The highlights of the architecture are as follows:

- In the healthcare domain, there is a wide array of shared care delivery collaborative processes such as the placing and tracking of orders (e.g. drug prescriptions, radiology orders, etc.). Parts of these processes are executed by workflows running in different departments of a healthcare institute. These intra-departmental workflows are usually implemented either by proprietary applications or more recently by the composition of medical e-services [2] through the use of Business Process Execution Language (BPEL) [6].

  A BPEL process provides only the view point of a single participant. For collaborative, inter-departmental or inter-enterprise healthcare business processes, the choreography of multi-party interactions as well as the collaboration rules and constraints among all the interacting parties of the healthcare enterprise should be provided. To express such multi-party collaboration among departments in an enterprise or between enterprises, a choreography language like Web Services Choreography Definition Language (WSCDL) [22] or ebXML Business Processes (ebBP) [10] is needed.

  In the enhanced IHE XDS implementation described in this paper, we use the ebXML Business Processes (ebBP) to describe collaborative healthcare processes which is a natural fit for the XDS since the XDS architecture is based on the ebXML framework.

- The clinical documents stored in ebXML registries are, in fact, obtained by executing collaborative healthcare business processes. As an example, an ECG report is produced by executing the IHE Scheduled Workflow Profile for Radiology. In XDS, the metadata of such a document needs to be stored in the registry with a link to the document itself. However, since IHE workflows are not a part of the XDS system, there is no way to track the processes and also a user has to manually submit the documents and their associated metadata to the registry by preparing the necessary "SubmissionSets" using "SubmitObjectsRequest" command of the ebXML registry [13].

  To automate this process and the information flow, we have integrated the ebBP business processes based on IHE Workflow Profiles into the IHE XDS architecture. We introduce an IHE XDS actor to represent the ebXML registry in the choreography specified by the ebBP whose primary task is to call a Web service to store the resulting clinical documents automatically based on the "Provide and Register Document Set" transaction of the XDS.

- In order to lower the barrier for application integration, we have implemented the enhanced IHE XDS in terms of Web services.

  It should be noted that IHE defined only one of the Integration Profiles, namely, Retrieve Information for Display (RID) [16] as a Web service by providing its WSDL (Web Service Description Language) [23] description. Yet, there are several benefits to be gained from basing IHE XDS implementations on a Service Oriented Architecture [20]:

  - The interfaces conform to universal standards, namely, SOAP [21] and WSDL [23] providing platform independence.
  - Discovery of the Web services facilitated by using ebXML also as a Web service registry.
  - It becomes possible to define intra-departmental workflows through Web service composition mechanisms like BPEL [6]. It should be noted that how to implement IHE departmental workflows as BPEL processes has already been addressed in the literature [2, 3]. Therefore, for this part of the work in this paper, we focus on the relationship between the collaborative healthcare processes defined through ebBP and departmental workflows defined through BPEL.

The paper is organized as follows: In Section 2, an overview of the ebXML architecture is presented. Section 3 gives a brief background on IHE, Integration Profiles and the XDS. An overview of the system architecture is presented in Section 4. Section 5 describes how IHE XDS can be integrated with ebXML business processes which are defined using IHE Workflow Profiles. Section 6 gives the implementation status of the system. Finally, Section 7 concludes the paper and presents the future work.

2 ebXML Architecture

ebXML specification aims to facilitate electronic business as follows:

- In order for enterprises to conduct electronic business with each other, they must first discover each other and the products and services they offer. ebXML provides a registry/repository architecture specification where such information can be published and discovered. A repository is a location (or a set of distributed locations) where a document pointed at by the registry resides and can be retrieved by conventional means (e.g., http or ftp). The repository is capable of storing any type of electronic content, while the registry is capable of storing metadata that describes content. The content within the repository is referred to "repository items" while the metadata within the registry is referred to "registry objects".

- An enterprise needs to determine which business processes and documents are necessary to communicate with a potential partner. Registry metadata can be used for searching relevant documents and business processes. A Collaboration Protocol Profile (CPP) [10] provides the details of how an organization is able to conduct business electronically. It specifies such items as how to locate contact and other information about the organization, the types of network and file transport protocols it uses, network addresses, security implementations, and how it does business by providing a reference to a Business Process Specification.

  A Business Process Specification Schema (ebBP) [10] in ebXML provides the definition of an XML document that describes how an organization conducts its business. While the CPA/CPP deals with the technical aspects of how to conduct business electronically, the ebBP deals with the actual business process.

- After the enterprises discover each other, they need to determine how to exchange information. The Collaboration Protocol Agreement (CPA) [10] specifies the details of how two organizations have agreed to conduct electronic business. It is formed by combining the CPPs of the two organizations.

3 IHE Implementation Framework

Integrating the Healthcare Enterprise (IHE) is a non-for-profit initiative that was founded in 1998 in the USA by the Radiological Society of North America (RSNA) and the Healthcare Information and Management Systems Society (HIMSS). IHE specifies an implementation framework to facilitate the integration of healthcare information resources.

While IHE does not develop standards as such, it selects and recommends appropriate standards for specific use cases and also develops restrictions, that is, application profiles for these standards that allow for simplified system integration.

3.1 IHE Integration Profiles

IHE Integration Profiles are use cases describing selected real-world scenarios. Each IT system involved in the use cases is called an "Actor". The interactions between actors are defined through "Transactions". More specifically, transactions define how IT systems communicate by using existing standards such as HL7 or DICOM to accomplish a specific task. In this way, IHE Integration Profiles define a collection of real world functionality and group together the necessary Actors and Transactions to make it work.

Integration profiles can further be categorized as follows: *Content Profiles*, *Workflow Profiles* and *Infrastructure Profiles*. *Content Profiles* address the management of a particular type of content object; *Workflow Profiles* address the management of the workflow process by which content is created; and *Infrastructure Profiles* address general departmental issues.

Content Profiles are "workflow neutral". The profile address how the object is created, stored, queried and retrieved, but does not address the workflow management process. The *Workflow Profiles* address managing workflow processes, which typically involve providing worklists, and reporting/monitoring the progress and completion of workitems. Within this context, one or more content objects are generally created according to their content profile. Current *Workflow Profiles* include "Scheduled Workflow" (for acquisition), "Post-Processing Workflow" and "Reporting Workflow". The *Infrastructure Profiles* address general departmental issues like "Basic Security" or "Access to Radiology Information".

3.2 IHE XDS

In IHE XDS, the ebXML repository is used for storing the clinical documents in folders in a persistent manner and the related metadata stored at the ebXML registry is used to facilitate the discovery of the documents.

In the IHE XDS Profile, healthcare enterprises that agree to work together for clinical document sharing is called a "Clinical Affinity Domain". Such institutes agree on a common set of policies such as how the patients are identified, the access is controlled, and the common set of coding terms to represent the metadata of the documents. The metadata defined is used for searching the ebXML registry to locate the documents in the repository.

An XDS document, stored in the repository, is represented as an *ebXML ExtrinsicObject* [12] in the registry. ExtrinsicObjects in ebXML are used to provide metadata that describes submitted content whose type is not intrinsically known to the registry. In order to group the related documents, the XDS documents are organized into folders (e.g. a period of care, a problem, immunizations, etc.). In this way a document consumer can find all the entries placed in the same folder. XDS document folders are constructed in the ebXML registry by using *ebXML RegistryPackage*. ebXML RegistryPackage is used to group logically related RegistryObject instances together.

In order to support atomic submission of documents to

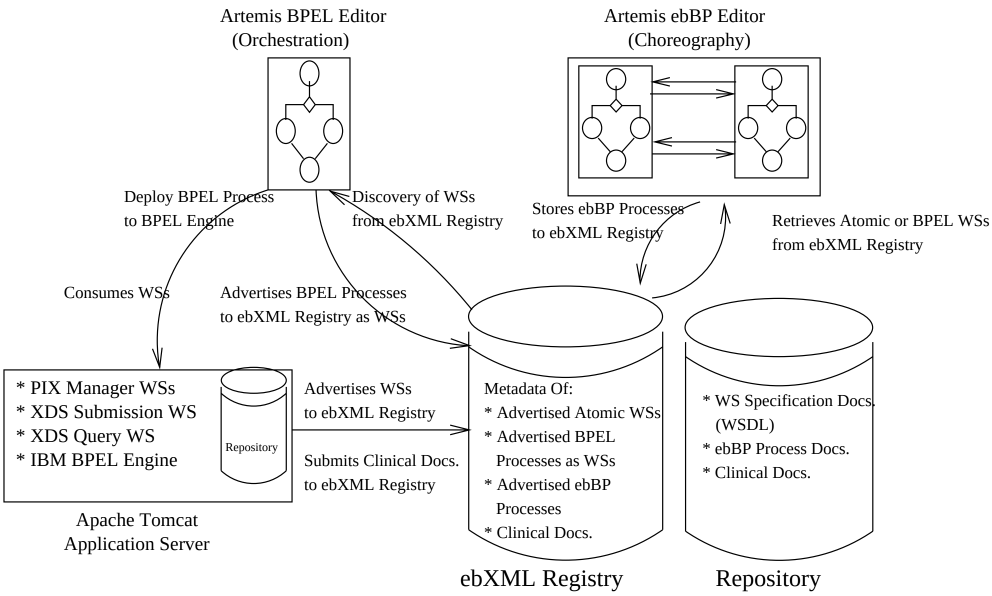

Figure 1. Overview of the System Architecture

the registry and to make a permanent record in the registry of objects, XDS documents are submitted as "XDS SubmissionSet"s by using the "SubmitObjectsRequest" command of the ebXML registry.

4 System Architecture Overview

As already mentioned, IHE Workflow Profiles describe transactions and choreography of these transactions as workflows. On the other hand, IHE XDS specifies how clinical documents can be shared in a clinical affinity domain. The missing link is associating the IHE XDS with IHE Workflow Profiles so that it becomes possible to automate and track the processes in XDS. Furthermore, the clinical documents produced by IHE workflows can thus be stored in the XDS registry without the need for human intervention.

The overall system architecture is presented in Figure 1. In this figure, all the documents created by the departmental workflows of the actors such as Admitting Office, Cardiography, Laboratory, and Radiology are shared through the central ebXML registry/repository. In addition, the actors also interact with each other for the notifications of activities such as patient registration, order placement and order result.

The main components of the system are as follows:

- *ebBP Editor*: In order to define collaborative healthcare processes in IHE XDS, we use ebBP [10] for the following reasons:
  - *Multi-party Choreography*: By using the choreography semantics of the ebBP, the ordering and sequencing of the activities between two or more independent parties can be achieved. Each party can then define its own logic and rules according to the specified choreography. Note that ebBP is for specifying collaboration among autonomous parties in contrast to BPEL which concentrates on centralized control (orchestration) of Web services for a single party.
  - *Reusability*: After an ebBP document is created, it can easily be used by any department to develop a Collaboration Protocol Profile (CPP) [11]. A CPP includes further information on how a department can realize the roles specified in the ebBP document. Therefore, it provides all the required details such as endpoint addresses, security protocols, and transport mechanism to achieve the interoperability. Once the CPPs of the departments are available; a collaboration agreement can be achieved by forming a Collaboration Protocol Agreement (CPA) [11].

Artemis ebBP Editor provides the basic functionalities in order to create process specifications. Due to the reusable nature of the ebBP specification, the editor allows a user to create main components (i.e. Package, Business Collaborations, Transactions, or Business Documents) independently from each other. It also allows a user to include further components as needed. The resulting process specification is serial-

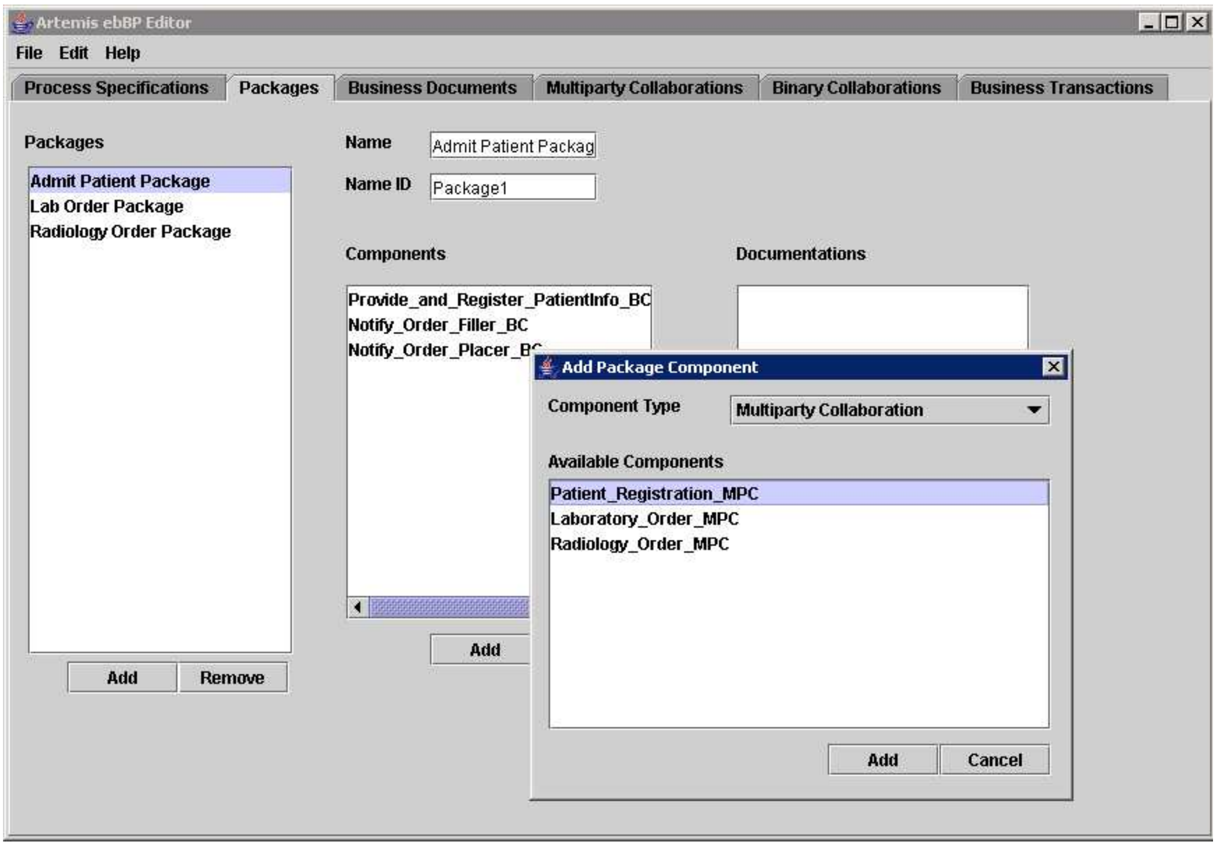

Figure 2. Artemis ebBP Editor User Interface

ized and advertised to the ebXML registry. As an example, in order to create a "Package" for the patient administration workflow, the user can select the required collaborations from the predefined building blocks as depicted in Figure 2.

- *ebXML Registry/Repository*: The metadata of the clinical documents, Web services supporting the infrastructure, BPEL processes exposed as Web services and ebXML business processes (ebBP) are stored in the ebXML registry. The clinical documents themselves, WSDL files of the Web services and the ebBP documents are stored in the repository.

- Web services supporting the basic infrastructure functionality are as follows:

  – Since there can be more than one enterprise or department in a "Clinical Affinity Domain" each with different patient identification schemes, a Web service with PIX manager functionality is implemented to facilitate mapping of patient identifiers in one domain into another.

  – Furthermore, to automate the communication with the ebXML registry, XDS Submission and XDS query facilities are also exposed as Web services.

- Once an ebBP process specification with XDS support is specified and a common agreement is achieved through a CPA, each intra-departmental workflow is implemented according to this agreement. To implement an internal workflow, we use Web Services and Web Service composition through BPEL [6] within a department. Note that this work has already been realized as described in [2, 3].

5 Integrating IHE Workflow Profiles with IHE XDS

In this section, we describe how IHE Workflow Profiles can be implemented through ebXML business processes and how they can be integrated to the IHE "Cross-Enterprise Document Sharing (XDS) Profile. In order to clarify the concepts introduced, we explain them through an example

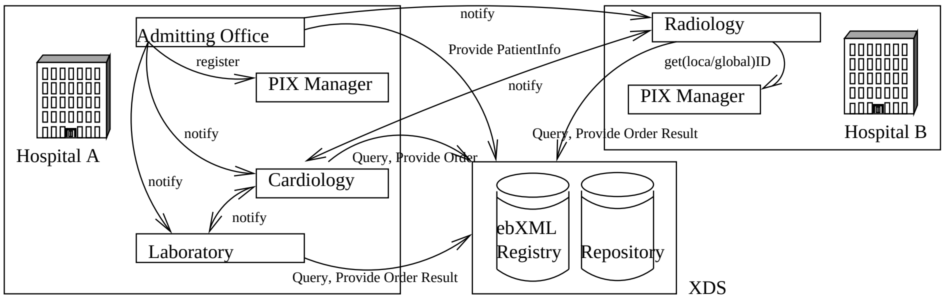

Figure 3. An Example Process Flow in the Healthcare Domain

scenario. We also show how the information in a CPA, which is obtained as a result of the ebXML Business process specification, is used to implement intra-departmental workflows through BPEL.

5.1 An Example Scenario

The scenario is about a cardiac patient who is hospitalized with severe chest pains and cold perspiration. The patient is admitted to the hospital after providing his demographics information and is referred to the Cardiology department. The patient information is also sent to the related departments in order to be used in further processes. The cardiologist places two orders, one for a "Creatine Kinase (CK)" test and another for the "CT color Doppler scan". The "Creatine Kinase (CK)" test order is sent to the Laboratory internal to the hospital, whereas the cardiac imaging order, "CT color Doppler scan", is sent to an external radiology unit. The cardiologist evaluates these test results for the final diagnosis.

The actors in this scenario are in the same "XDS Clinical Affinity Domain". The overall process in this scenario, as shown in Figure 3, is constructed through ebBP editor (Figure 2) from predefined processes based on the "IHE Scheduled Workflow Profile". The documents shared among the parties are automatically stored in the ebXML registry (i.e. XDS actor) by using a Web service based on the "Provide and Register" transaction. Furthermore, the parties which need a particular document can use the related Web services to query the registry and retrieve the document. In addition, the parties in the "Clinical Affinity Domain" which use different patient identifiers, for example, Cardiology and Radiology departments, use a Web service to access the PIX Manager to get the relevant patient ids.

5.2 Implementing IHE Workflow Profiles through ebXML Business Process (ebBP)

In this section, we present how the abstract IHE Workflow Profiles can be implemented through ebXML Business Processes (ebBP). Table 1 shows the correspondences between IHE Workflow Profile concepts and the ebBP concepts.

IHE Workflow Profiles	ebBP
Message	Business Document
Message Semantics	External Schema (pointed by Business Document)
Interaction	Business Transaction
Transaction	Business Collaboration
Sequence Diagram	Choreography

Table 1. Representing IHE Workflow Profiles through ebBP

As previously mentioned, an IHE workflow is made up of actors and transactions among them. Each IHE transaction can be considered as an exchange of one or more documents among two or more actors. Therefore, a document exchange, namely an interaction, can be represented as an ebBP "BusinessTransaction" (Table 1). A "BusinessTransaction" can define one-way or request-response document flows with the help of its "RequestingBusinessActivity" and "RespondingBusinessActivity" elements. Each element refers to one or more "BusinessDocument"s through its "DocumentEnvelope". However, for each requesting or responding activity of a "BusinessTransaction", there is only one primary document while others are specified as "Attachment"s.

As an example, consider "Patient Registration" transaction of the IHE Scheduled Workflow in the example scenario. A part of this transaction involves the submission of the "PatientInfo" document between Admitting Office and Cardiology department. This document flow is represented as a "BusinessTransaction" as shown in Figure 4.

In ebBP, the "BusinessTransaction"s, are organized into collaborations in order to specify the transactions among

```
<BusinessTransaction name="Patient Registration">
  <RequestingBusinessActivity name="">
    <DocumentEnvelope isPositiveResponse="true"
      businessDocument="PatientInfo">
    </DocumentEnvelope>
  </RequestingBusinessActivity>
  <RespondingBusinessActivity name=""/>
</BusinessTransaction>
```

Figure 4. Business Transaction

two or more actors. If a transaction happens between two actors, a "BinaryCollaboration" is used. A "BinaryCollaboration" can use a "BusinessTransaction" with the help of its "BusinessActivity" element. The "fromAuthorizedRole" of a "BusinessActivity" element points to one of the two roles performing the corresponding "BinaryCollaboration" while "toAuthorizedRole" refers to the other.

If an IHE transaction includes more than two actors, a "MultiPartyCollaboration" is used in ebBP. In fact, a "MultiPartyCollaboration" consists of a set of "BinaryCollaboration"s whose roles are performed by one or more "BusinessPartnerRole"s of the corresponding "MultiPartyCollaboration".

For the example scenario, the Multiparty collaboration among the Admitting Office(ADT), the XDS, Cardiology (Order Placer), and Radiology (Order Filler) for the "Patient Registration" IHE transaction is shown in Figure 5. In this figure each transaction between Admitting Office and any of the parties, Cardiology, Radiology, and XDS, is specified as a "BinaryCollaboration" in ebBP. Also note that the "BinaryCollaboration" can use the "BusinessTransaction"s in order to state the interactions within the collaboration. These "BinaryCollaboration"s are further used by the "MultiPartyCollaboration" in order to define "Patient Registration" transaction. By using this mechanism, any IHE Scheduled Workflow transaction can be easily represented through ebBP.

After the transactions are defined as collaborations in ebBP, the choreography of these transactions need to be defined. ebBP defines choreography through the Business States and the transitions between them. The "BusinessTransactionActivity", which is used in "BinaryCollaboration"s in Figure 5, is a concrete Business State. In addition to this, ebBP specification provides some supplementary kinds of Business States such as a "Start state" or a "Completion State" which can be "Success" or "Failure".

5.3 Integrating IHE XDS to ebBP process specification

Another problem that needs to be addressed is to be able to share the clinical documents produced as a result of healthcare processes through the IHE "Cross Enterprise Document Sharing (XDS)" profile.

To realize this, an XDS actor is introduced whose primary task is to store the clinical documents produced as a result of a workflow. Therefore, we slightly modify process flow for each actor that produces a clinical document. The actor stores the document at the ebXML registry by using the XDS Web service implementing the "Provide and Register Document Set" transaction of the XDS. In this transaction, the actor, which produces the clinical document, sends both the document and its metadata description based on XDS profile by using the XDS Web service. The further details of the XDS Submission Web service are discussed in Section 5.6.

For instance, in the example scenario to send the "PatientInfo" document from the Admitting Office by using the XDS Submission Web service, a "BusinessTransaction" shown in Figure 6 is used in the process specification. Note that the actual document, "PatientInfo", is attached to its metadata description document which is the primary document of the transaction.

```
<BusinessTransaction name="Patient Registration">
  <RequestingBusinessActivity name="">
    <DocumentEnvelope isPositiveResponse="true"
      businessDocument="PatientInfoMetadata">
      <Attachment name="PatientInfo" mimeType="XML"
       isConfidential="true" isAuthenticated="true"/>
    </DocumentEnvelope>
  </RequestingBusinessActivity>
  <RespondingBusinessActivity name=""/>
</BusinessTransaction>
```

Figure 6. Handling Documents in ebBP

The interaction defined in Figure 6 is then used by the "BinaryCollaboration" between the actor, which is the document source, and the XDS Actor representing the XDS Web service. Once such collaboration is defined, it can be used by the "MultiPartyCollaboration" as illustrated in Figure 5. Since the actual document is stored in the registry/repository of XDS, it is not sent to Cardiology and Radiology departments. Instead, a notification with the required information to query XDS registry is sent. Each department can then use the XDS Query Web service to obtain the actual document.

The documents stored in XDS registry/repository are associated with the patient IDs registered to the XDS Affinity Patient Identifier Domain. Note that each actor, which submits the clinical document by using the XDS Submission Web service, is responsible for the mapping of its local patient ID to the patient ID of the XDS Affinity Domain by using the PIX Manager Web service. In addition, PIX Manager Web service is also used by the actors which use the XDS Query Web service to query the documents (XDS document consumer) in order to map the patient ID of XDS Affinity Domain to the local patient ID. The PIX Manager, XDS Submission and Query Web services are discussed in

```xml
<!-- Patient Registration Transaction -->
<MultiPartyCollaboration name="Patient Registration">
  <!-- Admitting Office -->
  <BusinessPartnerRole name="ADT">
    <Performs initiatingRole=
      //binaryCollaboration[@name="Provide and Register Set"]/InitiatingRole[@name="initiator"]/>
    <Performs initiatingRole=
      //binaryCollaboration[@name="Notify Order Placer"]/InitiatingRole[@name="initiator"]/>
    <Performs initiatingRole=
      //binaryCollaboration[@name="Notify Order Filler"]/InitiatingRole[@name="initiator"]/>
    <Transition
      fromBusinessState= //binaryCollaboration[@name="Provide and Register Set"]/[@name="Register Patient Activity"]
      toBusinessState= //binaryCollaboration[@name="Notify Order Placer"]/[@name="Notify Order Placer Activity"]/>
    <Transition
      fromBusinessState= //binaryCollaboration[@name="Notify Order Placer"]/[@name="Notify Order Placer Activity"]
      toBusinessState= //binaryCollaboration[@name="Notify Order Filler"]/[@name="Notify Order Filler Activity"]/>
    ...
  </BusinessPartnerRole>
  <!-- XDS Actor -->
  <BusinessPartnerRole name="XDS">
    <Performs respondingRole=
      //binaryCollaboration[@name="Provide and Register Set"]/RespondingRole[@name="responder"]/>
  </BusinessPartnerRole>
  <!-- Radiology -->
  <BusinessPartnerRole name="OrderFiller">
    <Performs respondingRole=
      //binaryCollaboration[@name="Notify Order Filler"]/RespondingRole[@name="responder"]/>
  </BusinessPartnerRole>
  <!-- Cardiology -->
  <BusinessPartnerRole name="OrderPlacer">
    <Performs respondingRole=
      //binaryCollaboration[@name="Notify Order Placer"]/RespondingRole[@name="responder"]/>
  </BusinessPartnerRole>
</MultiPartyCollaboration>
<!-- Part of Patient Registration Transaction between ADT and XDS -->
<BinaryCollaboration name="Provide and Register Set">
  <InitiatingRole name="initiator""/>
  <RespondingRole name="responder"/>
  <BusinessTransactionActivity name="Register Patient Activity" businessTransaction="Patient Registration"
    fromAuthorizedRole="initiator" toAuthorizedRole="responder"/>
  <Start toBusinessState="Register Patient Activity"/>
</BinaryCollaboration>
<!-- Part of Patient Registration Transaction between ADT and Order Placer -->
<BinaryCollaboration name="Notify Order Placer">
  <InitiatingRole name="initiator""/>
  <RespondingRole name="responder"/>
  <BusinessTransactionActivity name="Notify Order Placer Activity" businessTransaction="Notify Registration Transaction"
    fromAuthorizedRole="initiator" toAuthorizedRole="responder"/>
</BinaryCollaboration>
<!-- Part of Patient Registration Transaction between ADT and Order Filler -->
<BinaryCollaboration name="Notify Order Filler">
  <InitiatingRole name="initiator"/>
  <RespondingRole name="responder"/>
  <BusinessTransactionActivity name="Notify Order Filler Activity" businessTransaction="Notify Registration Transaction"
    fromAuthorizedRole="initiator" toAuthorizedRole="responder"/>
</BinaryCollaboration>
```

Figure 5. Representing IHE Transactions as ebBP Collaborations

detail in Section 5.6

5.4 Collaboration Protocol Profiles (CPPs) and Collaboration Protocol Agreements (CPAs)

Once the overall process is specified in ebBP with all actors and documents, each actor involved in the process specification as a party defines its capabilities within a *Collaboration Protocol Profile (CPP)*. The CPPs are then used to construct the CPA for an agreement to execute healthcare processes in collaboration.

The capabilities of an actor are detailed in “PartyInfo” element of a CPP document. The “PartyInfo” element associates a party to a role in collaboration with its “CollaborationRole” element. There can be more than one “CollaborationRole” element within a “PartyInfo” in order to associate an actor with more than one role. Furthermore, these roles can be defined for different process specifications so that an actor can participate in one or more process specifications with its CPP. As an example of “CollaborationRole”, the Radiology department in the example scenario (Figure 3) is supposed to fulfill the “Order Filler” role of the process specification “IHE Scheduled Workflow”. Therefore, its CPP definition has a “CollaborationRole” as depicted in

Figure 7 which refers to the "OrderFiller" role of "MultiPartyCollaboration" shown in Figure 5.

```
<CollaborationRole id="CR1">
  <ProcessSpecification version="1.0" name="IHESWF"
    xlink:type="..."/>
  <Role name="OrderFiller" xlink:type="simple"
    xlink:href="..."/>
  <CertificateRef certId="Cer1"/>
  <ServiceBinding channelId="Channel1" packageId="N0402">
    ...
  </ServiceBinding>
</CollaborationRole>
```

Figure 7. Specifying the CollaborationRole of a party in ebBP

A CPP also gives the technical details on how an actor manages to fulfill a role in the scenario. For each "CollaborationRole", a "ServiceBinding" element, shown in Figure 7, is specified in order to define the channel and package of the message exchange with other actors in the collaboration. The "channelID" attribute of the "ServiceBinding" refers to a "DeliveryChannel" as shown in Figure 8. In addition to specifying some characteristics of the channel such as confidentiality, authentication and security, "DeliveryChannel" refers to a "Transport" element in which the communication capabilities of the actor to fulfill the roles are defined. As an example, the Radiology department can specify an endpoint in "Transport" element which sends and receives order messages through HTTP protocol.

```
<DeliveryChannel channelId="Channel1"
  transportId="Transport1"
  docExchangeId="...">
  <Characteristics syncReplyMode="none"
    secureTransport="true"
    confidentiality="true" authenticated="true"/>
</DeliveryChannel>
<Transport transportId="Transport1">
  <SendingProtocol version="1.1">HTTP</SendingProtocol>
  <ReceivingProtocol version="1.1">HTTP</ReceivingProtocol>
  <Endpoint uri="http://www.srdc.metu.edu.tr/radiologyOrder"
    type="allPurpose"/>
  <TransportSecurity>
    <Protocol version="3.0">SSL</Protocol>
    <CertificateRef certId="N03"/>
  </TransportSecurity>
</Transport>
```

Figure 8. Specifying the DeliveryChannel and the Transport in ebBP

Furthermore, each CPP also provides the technical details such as constraints on security, reliable messaging and packaging specification. In the example scenario (Section 5.1), the CPPs are used to form a Collaboration Protocol Agreement (CPA) among all the actors of the healthcare process. The process describing how to form a CPA from the CPPs is given in [11]. The agreement reached among the healthcare units by forming a CPA is then used by each actor to implement its own departmental workflow according to the specifications (i.e. endpoints, packaging), and constraints (i.e. security) in the CPA.

In Section 5.5, we show how the information in a CPA can be used by a department to implement its internal workflow.

5.5 Implementing Intra-Departmental Workflows

How to implement IHE departmental workflows as BPEL processes has already been addressed in the literature [2, 3]. In this section, we focus on the relationship between the collaborative healthcare processes defined through ebBP and departmental workflows defined through BPEL.

While defining its internal workflow, each actor should fulfill the roles assigned to it in the ebBP document and use the technical details specified in the CPA document. For instance, in the example scenario, the Admitting Office (ADT) actor needs to construct an orchestration for the "Patient Registration" which is activated when a new patient arrives. This orchestration processes the patient information retrieved from the admitting office, prepares and submits the document to the XDS registry as specified in Section 5.2 and 5.3. When the patient is referred to a cardiologist and the cardiologist places his orders, the Cardiology (Order Placer) and Radiology (Order Filler) are notified.

While providing the departmental orchestrations in BPEL, the entry point of each orchestration is published as a Web service by offering the service interface as a WSDL file. Then, the BPEL definition refers to the client which starts the orchestration by invoking this Web service as a partner.

In BPEL [6] (version 1.1), the actors (including the client) with which the orchestration interacts are specified as "partners". Each "partner" can have one or more "partnerLinks" in order to specify the relationship between the partner and the orchestration. Furthermore, each "partnerLink" has a "partnerLinkType" which refers to the service definition files (WSDL) in order to specify the port types of each interaction. As an example, consider some of the partner definitions in the "Admit Patient (ADT)" orchestration as shown in Figure 9. In this Figure, the "clientLink" refers to the service definition (WSDL) of the ADT orchestration entry point for its port type definition. Here, it should be noted that this port type is bound to a port whose endpoint URI corresponds to the predefined endpoint URI specified in the "Transport" element of the ADT "CollaborationRole" in the CPA.

The orchestration logic specified in BPEL for each actor should also reflect the choreography specified in the process specification. Based on the ebBP and CPA of the ex-

```
<partnerLinks>
  <partnerLink name="clientLink" partnerLinkType=
    "ADTWSDL:ADTLinkType"
    myRole="ADTProvider" partnerRole="ADTRequester">
  </partnerLink>
  <partnerLink name="XDSLink" partnerLinkType=
    "XDSWSDL:XDSLinkType"
    partnerRole="XDSSubmitProvider">
  </partnerLink>
 ...
</partnerLinks>
<partners>
  <partner name="client">
    <partnerLink name="clientLink"/>
  </partner>
  <partner name="XDS">
    <partnerLink name="XDSLink"/>
  </partner>
 ...
</partners>
```

Figure 9. BPEL partners and partnerLinks

ample scenario, the activities of the orchestration should perform the roles of each actor in collaboration. For example, the ADT orchestration needs to prepare and submit the "PatientInfo" document and its metadata to the XDS registry/repository before notifying Order Placer and Order Filler. A "Sequence" activity for ADT is depicted in Figure 10. As shown in this figure, the orchestration invokes PIX Manager Web service before the XDS submission, since XDS profile requires the submission of documents together with their patient IDs.

5.6 Infrastructure Support Web Services

In this section, a number of Web services implemented for infrastructure support are described:

- *The PIX Manager Web service* supports the cross-referencing of patient identifiers from multiple Patient Identifier Domains by using the interactions defined in PIX profile [16]. The PIX Manager Web service is implemented as follows:
  - The creation of patient identifiers that are aliases of one another across different domains is the responsibility of the PIX Manager Actor. We supply a "registerToPIX" operation to the PIX Manager Web service in order to implement this functionality. The ADT system generates its patient ID and invokes "registerToPIX" operation, which corresponds to the Patient Identity Feed transaction in IHE PIX, by also providing its own unique domain identifier. As a result, a patient ID in Cross-reference Domain is created and returned to the ADT. For each patient, the system holds a unique identifier in Cross-reference Domain and the local patient identifiers with their corresponding domains.

```
<sequence name="main">
 <!-- Receive input from client -->
 <receive name="receivePatientInfo" partnerLink="client"
   portType="ADTWSDL:ADTPortType" operation="registerPatient"
   variable="input" createInstance="yes"/>
 <assign name="PIXAssg">
  <copy>
   <from variable="input" part="payload"
     query="/PatientInfo/PatientID">
   </from>
   <to variable="PIXInput" part="payload"
     query="/pixns:PatientInfo/pixns:localPatientID"/>
   </copy>
 </assign>
 <invoke partnerLink="PIXLink" portType="PIXWSDL:PIXPortType"
   operation="registerLocalID" inputVariable="PIXInput"
   outputVariable="PIXResponse"/>
 <assign name="XDSAssg">
  <copy>
   <from variable="PIXResponse" part="payload"
     query="/pixns:PatientInfo/pixns:GlobalID">
   </from>
   <to variable="XDSInput" part="payload"
     query="/xdsns:PatientInfo/xdsns:PatientID"/>
  </copy>
  ...
 </assign>
 <invoke partnerLink="XDSLink" portType="XDSWSDL:XDSPortType"
   operation="submitDocument" inputVariable="XDSInput"/>
 ...
</sequence>
```

Figure 10. Sequence activity in ADT Orchestration

  - Another responsibility of the PIX Manager Actor is to provide a list of identifiers across different domains. Since the system stores each patient identifier with its corresponding domain identifier, we provide a "getGlobalID" operation to the PIX Web service which returns the unique identifier of the patient in Cross-reference Domain. Furthermore, we provide a "getLocalID" operation which supplies the identifier of a patient in a specific domain.
  - Each actor in a specific domain may also send requests if there is an update on a patient ID of in that domain. Therefore, we provide a "notify" operation to reflect such identifier updates on the PIX Manager.
  - The PIX Manager Web service should communicate with multiple domains which may use different architectures. Instead of using arbitrary message formats for the operations of PIX Manager Web service, we use Security Assertion Markup Language (SAML) [19] specification. The Name Identifier Mapping Protocol, defined in SAML, is used for mapping an identifier into a different identifier for the same principal in a different domain. In our system, each domain sends a "NameIDMappingRequest" to

the Identity Provider (PIX Manager Web service) and gets a "NameIDMappingResponse". Any domain can use this protocol by implementing an engine which translates the request to a corresponding SAML request and vice versa.

- *The XDS Submission and Query Web services* provide an interface for the ebXML registry/repository to be used by document source and document consumer. These Web services are implemented according to the IHE XDS profile [17] as follows:

  - XDS profile specifies "Provide and Register Document Set" transaction in order to submit a set of documents prepared by a Document Source to the registry/repository. The XDS Submission Web service provides an operation "submitDocument" which takes the document to be submitted and data about the document (metadata) as input. It then prepares a submission request according to the ebXML specifications.
  - Once the documents are submitted by the document source, the registry can be queried by the consumers in order to find the identifiers of the requested documents. XDS profile provides "Query Registry" transaction which searches the registry according to the metadata of the document. Currently, XDS Query Web service provides a limited set of operations each of which corresponds to one of the basic query types specified in XDS. As an example, one type of query is used to search the registry for a patient ID in a specific date range. The query returns a unique identifier of the requested document. This unique identifier is later used "Retrieve Document Request" transaction to retrieve the actual document.
  - Finally, the result of the "Query Registry" (unique identifier) is used to retrieve the document from the repository as specified in "Retrieve Document" transaction of the XDS profile. This is implemented as a "getDocument" operation in XDS Web service which takes the unique identifier of the document and returns the document as an attachment.

The XDS Submission, Query and PIX Manager Web services are also published to the ebXML registry to be discovered and invoked by the departmental workflows. Each service is represented as a "Service" class in ebXML registry. Technical specification(WSDL) of each "Service" is also specified through its "serviceBindings" attribute.

6 Implementation Status

This work is realized within the scope of the Artemis project [5, 8, 1] which aims to develop a semantic Web services based interoperability framework for the healthcare domain. In the Artemis project, the legacy systems are wrapped as Web services to access to the clinical information together with the semantic information that facilitates mappings between different formats. To complement the Artemis framework with an industry initiative, IHE implementation framework is currently being integrated into the system as Web services.

As a proof of concept implementation of the work described in this paper, OASIS ebXML Registry Reference Implementation [9] is used as the central ebXML registry/repository. As an application server to host Web services to be accessed through SOAP, Apache Tomcat 5.0 [4] is used. The IHE PIX Manager, XDS Submission and Query Web services are developed using Java Web Services Developer Pack 1.5 (JWSDP) [18].

The collaboration documents are constructed according to the ebXML Business Process Specifications [10]. The departmental workflows are constructed by the using the BPEL4WS version 1.1 [6]. These workflows are then deployed to the IBM Business Process Execution Language for Web Services Java (BPWS4J) [15].

7 Conclusions

IHE is an important initiative strongly supported by the industry: more than 110 companies have developed IHE compliant systems between 1999 and 2004 and participated in the cross-vendor testing events organized by IHE, including most of the market leaders. This means that standards recommended by IHE have a high probability of a quick uptake in the medical market.

Currently what is sorely missing in IHE XDS is business process support. Indeed, in the healthcare domain, there is a wide array of shared care delivery collaborative processes such as the placing and tracking of orders (e.g. drug prescriptions, radiology orders, etc.).

Although, through a separate effort, IHE has also defined inter-departmental Workflow Profiles, these have not been integrated to XDS Profile. Hence, there is no way to track the processes in XDS and the resulting clinical documents are manually inserted into the registry/repository.

In this paper, we describe the implementation of an enhanced IHE architecture integrating ebXML Business Processes, IHE Workflow Profiles and the IHE XDS architecture. Our choice of using ebXML Business Processes (ebBP) in IHE XDS is important for the following reasons:

- For collaborative, inter-departmental or inter-enterprise healthcare business processes, the choreography of multi-party interactions as well as the collaboration rules and constraints among all the interacting parties of the healthcare enterprise should be provided. To express such multi-party collaboration among departments in an enterprise or between enterprises, a choreography language like Web Services Choreography Definition Language (WSCDL) [22] or ebXML Business Processes (ebBP) [10] is needed; business process orchestration languages, like BPEL, falls short to satisfy this need.

- ebBP is the most natural way to integrate IHE Workflow Profiles to IHE XDS since IHE XDS is based on the ebXML architecture.

We show that by using the ebBP Binary and Multiparty collaborations, it is possible to completely specify the IHE workflow profiles. These collaborations include all the required information such as exchange of documents (interactions), actors, and other constraints or conditions in order to perform an enterprise-wide workflow management. The ebBP specification document is then used by each actor involved to have an agreement (CPA) and to create its own logic and rules to specify intra-departmental workflows in BPEL by composing Web services. We also provided a number of basic Web services to support the infrastructure.

Currently, the metadata elements we use in XDS Query Web services are quite limited. We only use metadata elements such as “patientId”, “creationTime”, “authorDepartment” or “authorInstitution” since only these can be automatically obtained from the system resources. As a future work we intend to extend the query capabilities of our implementation beyond the metadata defined by XDS by also considering the content of XDS documents defined through structured markup.

References

[1] Aden, T, Eichelberg, M., ARTEMIS Deliverable D5.1.1: Relevant Electronic Healthcare Record Standards and protocols for accessing medical information, http://www.srdc.metu.edu.-tr/webpage/projects/artemis/calendar.php.

[2] Anzböck, R., Dustdar, S. “Modeling Medical e-services”, 2nd International Conference on Business Process Management (BPM 2004), June 2004, Germany, Springer LNCS.

[3] Anzböck, R., Dustdar, S., “Modeling and Implementing Medical Web services”, Data and Knowledge Engineering, Elsevier, to appear.

[4] The Apache Jakarta Tomcat 5 Servlet/JSP Container, http:-//jakarta.apache.org/tomcat/tomcat-5.0-doc/index.html

[5] Artemis Project, http://www.srdc.metu.edu.tr/-webpage/projects /artemis

[6] BEA Systems, IBM, Microsoft, SAP AG, Siebel Systems, “Business Process Execution Language for Web Services 1.1”, http://www-128.ibm.com/developerworks/library/-specifi cation/ws-bpel/, 2003.

[7] DICOM, NEMA Standards Publication PS 3.x, Digital Imaging and Communications in Medicine (DICOM), National Electrical Manufacturers Association, 2004.

[8] Dogac, A., Laleci, G., Kirbas S., Kabak Y., Sinir S., Yildiz A. Gurcan, Y., “Artemis: Deploying Semantically Enriched Web Services in the Healthcare Domain”, Information Systems Journal (Elsevier), to appear.

[9] freebXML Implementation, http://ebxmlrr.sourceforge.net/-2.1/index.html

[10] ebXML Business Process Specifi cation Schema v1.01, May 2001, http://www.ebxml.org/specs/ebBPSS.pdf.

[11] ebXML, Collaboration-Protocol Profi le and Agreement Specifi cation v1.0, May 2001, http://www.ebxml.org/specs/-ebCCP.pdf.

[12] ebXML Registry Information Model v2.5, http://www.oasis-open.org/committees/regrep/documents/2.5/specs/ebRIM-2.5.pdf

[13] ebXML Registry Services Specifi cation v2.5, http://www.-oasis-open.org/committees/regrep/documents/2.5/specs/ebRS-2.5.pdf

[14] Health Level 7 (HL7), http://www.hl7.org

[15] IBM Business Process Execution Language for Web Services JavaTM Run Time (BPWS4J) , http:-//www.alphaworks.ibm.com/tech/bpws4j

[16] IHE IT Infrastructure Integration Profi les, http://www.rsna.-org/IHE/tf/ihe_tf_index.shtml

[17] IHE IT Infrastructure Technical Framework, Cross-Enterprise Clinical Documents Sharing (XDS), by the IHE ITI Technical Committee, Version 3.0, April 26, 2004.

[18] Java Web Services Developer Pack (JWSDP), http://java.-sun.com/webservices/jwsdp/index.jsp

[19] Security Assertion Markup Language (SAML) 2.0 Technical Overview, http://xml.coverpages.org/SAML--TechOverviewV20-Draft7874.pdf

[20] Yang, J., Bussler, C., Service Oriented Modelling. Special issue of the International Journal of Cooperative Information Systems (IJCIS). Volume 13, Number 4, December 2004.

[21] SOAP: Simple Object Access Protocol, http://www.w3.org/-TR/SOAP/.

[22] Web Services Choreography Description Language Version 1.0, http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427/

[23] WSDL: Web Service Description Language, http://www.-w3.org/TR/wsdl.