Modeling Medical e-services

Rainer Anzböck[1], Schahram Dustdar[2]

[1] D.A.T.A. Corporation,
Invalidenstrasse 5-7/10, 1030 Wien, Austria
ar@data.at
[2] Distributed Systems Group, Vienna University of Technology
Argentinierstrasse 8/184-1, 1040 Wien, Austria
dustdar@infosys.tuwien.ac.at

Abstract. On the one hand Web services are gaining increasing attention. A lot of standardization has improved their stability and range of application. Composition and coordination techniques for Web services enable an application integration effort beyond loosely coupled systems. On the other hand medical e-services are covered by the DICOM and HL7 communication protocols and profiled by the IHE (Integrating the Healthcare Enterprise) technical framework. Standardization is more extensive, most workflows are well defined and integration is tighter than in most other domains. Nevertheless standardization focused on conventional workflow systems. In an Internet-based medical environment with high security standards, communication is strongly restricted and conventional systems fail to deliver. This paper proposes a modeling process for medical Web services. The IHE *patient administration process flow* serves as a well defined example. Furthermore, the paper defines requirements of a Web service based middleware for the execution of medical e-services. The technique should enable building integrated medical applications for Internet-based workflow execution.

1 Introduction

With recent work in the field of workflows it is possible to define more flexible business models than in traditional workflows based on the Workflow reference model (WFMC) [1]. With the standardization of coordination, composition, transaction and security for Web services a new implementation method for Web service based scenarios is available. Especially the medical services domain is in a permanent evolution. Its workflows are complex and highly structured and a standardization of communication protocols has been covered by HL7 [2], DICOM [3] and the IHE framework [4]. Further standardization processes for health informatics are enforced by the European Union with the CEN/TC 251 work program [5].

One goal of our paper is to outline a modeling process for medical e-services. From the medical services domain initially introduced and the requirements defined subsequently, we conclude how to model such services based on the IHE *administrative process flow* sample in 5 steps. The modeling process should be refined in further research and result in a guideline or semi-automatic process for defining medical e-services' workflows using Web service based composition.

Another goal of our paper is to show how recent work on protocols of the Web service stack and standardization efforts in the medical services domain (the IHE framework) help to solve application integration. First, we provide an introduction to the medical services domain. Then we outline requirements of a Web service oriented approach and use a specific example, the *administrative process flow*. When going into detail, we further focus on two IHE transactions, *patient registration* and *modality worklist provided*, as they are representative for HL7 and DICOM communication.

A third goal is the discussion of requirements for modeling medical e-services. Related to the example introduced we discuss Web service concepts and standards like SOAP [6], WSDL [7], WS-Coordination [8], WS-Transaction [9], WS-Security [10] and many more. From there we focus on the composition of Web services using BPEL [11] and define requirements to model IHE transactions as medical e-services. Finally, we conclude the results and provide topics for future work.

To summarize, our paper (i) suggests a modeling process for the *IHE administrative process flow* example and outlines implications for a general modeling process to implement medical e-services, (ii) introduces the medical services domain and the *administrative process flow* and (iii) defines requirements of a modeling process based on current Web service stack standards.

The paper is structured as follows. Section 2 introduces medical information systems, communication protocols and the IHE technical framework. Section 3 provides requirements of a modeling process for services like the IHE *administrative process flow*. Section 4 outlines a modeling process for medical e-services. Section 5 concludes the results and outlines future work.

2 Medical e-services

In this chapter we briefly introduce medical information systems, communication protocols and the IHE framework.

2.1 Medical information systems

Three types of medical information systems, the HIS (Hospital Information System), the RIS (Radiology Information System) and the PACS (Picture Achieving and Communication system) are the backbone of current information systems in the hospital and medical e-services environment. They are comparable to ERP (Enterprise Resource Planning) or SCM (Supply Chain Management) in business organizations. The HIS is an enterprise-wide system used for administrative services like patient and visit management, operation planning, billing, etc.. The RIS is a management system for medical imaging facilities (radiologists) and covers patient registration, examination scheduling and control, report generation, speech recognition, etc.. As can be concluded, both systems have overlapping services to fulfill: one on an enterprise the other on a department level. The second main software system category in medical e-services is called PACS and is responsible for all image management services. It transfers patient data to examination facilities (modalities), announces finished procedures and stores, prints, burns CDs, archives or transfers the generated image data.

These software systems are often integrated as departmental services for a larger hospital environment or spread across several locations. Because of their special storage, network and process performance requirements RIS and PACS systems are very important departmental services. Company related information on these systems can be found in [12-18], more theoretical work in [19-21].

2.2 Medical communication standards

The most relevant protocol standards for these services are HL7 for the RIS and DICOM for the PACS. PACS and RIS both implement a workflow model and cover implementations of the standard. Both systems have to be tightly integrated to perform services efficiently. The DICOM standard covers Client/Server communications used to exchange patient and examination information. The standard covers objects like patients, visits, medical procedures, images, etc.. Additionally, notifications, data query and exchange services based on these objects are defined. The HL7 standard is used for data exchange between different healthcare providers and is more suited for non-radiological institutions. Some functionality overlaps with DICOM for example the scheduling process or the patient and result management. Other functionality such as the exchange of image data is not part of HL7. More detailed information on HL7 can be found in [2] and on DICOM in [3, 22, 23]. Besides these protocols additional standards like CEN 251 [5] exist. Ambitions to converge these standards by using a common framework have led to the definition of IHE [4].

2.3 A medical workflow framework

The IHE technical framework has been defined to extend the enterprise application integration to a level of scenario-based interaction. Over the years software products implemented the DICOM and HL7 standards by their own interpretation. This led to a situation of incompatibility and a lot of effort has to be put into application integration. The framework defines usage-scenarios with the goal that products conforming to the framework can be integrated seamlessly.

IHE defines transactions (workflow transactions) between applications by profiling DICOM and HL7 operations. Messages (domain activities) are selected and put into sequences to implement real-world scenarios. Additionally, flows (workflow services) are defined that correspond to a set of related transactions performed by different actors (administration application, image archive, etc.). Applications may perform the role of one or more such actors in one or more of these flows. To claim IHE conformity for a role in a workflow, a required set of flows and transactions has to be implemented.

IHE conformant applications can be integrated more tightly than applications in other domains. Nevertheless integration based on this framework is currently done using traditional workflow models in Intranet-based environments. An Internet-based infrastructure, as currently common in most environments, restricts interorganizational workflow [24] integration. In a real world scenario integrators have to deal with applications in a mixed Intranet and Internet environment. Workflow items like patient and image data are exchanged within and across organizational boundaries. Figure 1 shows an example of such an environment.

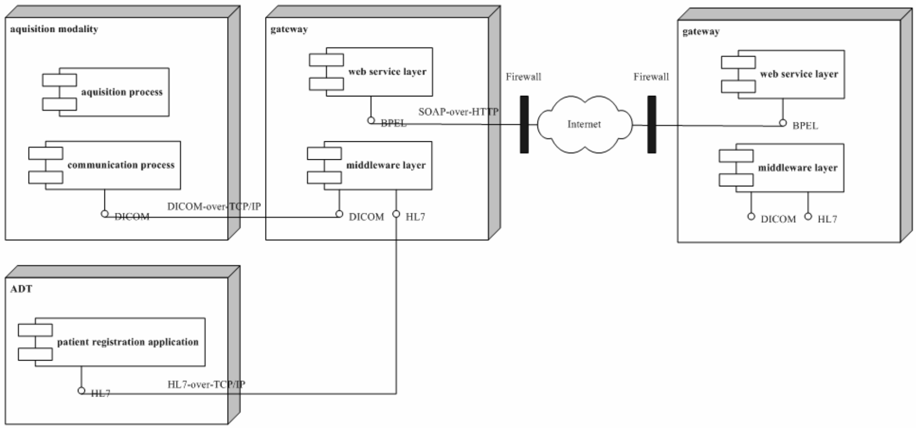

Fig. 1. Mixed Intranet/Internet environment for medical e-services

An Intranet-based environment consists of conventional HL7 and DICOM communication over a secure and reliable transport. Additionally, the IHE framework provides a solid foundation for defining medical workflows in this environment. Current solutions integrate applications based on conventional middleware. For example, gateways, acquisition modalities and patient registration applications are directly connected by their middleware layers. In contrast, we have to deal with interorganizational workflows, which are executed between nodes distributed over the Internet.

The gateways mentioned have two different responsibilities. On one hand, they implement IHE conformant Web service based workflow models for medical e-services. On the other hand, they enable internal nodes to participate in IHE conformant workflows, to attach their messages to XML workflow messages and to apply security and transaction support. In this paper we focus on the first functionality.

This scenario is beneficial for many reasons, like exchange of patient information which results in a reduced number of examinations, load balancing work between specialized physicians, etc. Through the standardization process related to the Web service stack [25] it is feasible to suggest a workflow implementation based on a separate layer that meets the requirements of an Internet environment on one hand, and supports standardization efforts of the industry, as outlined above, on the other.

Related to Web services, we have to consider the following aspects. First, we have to provide a transport mechanism, where SOAP-over-HTTP communication is a reasonable option. Next, we have to meet reliability and security requirements with additions like WS-Security [10], WS-ReliableMessaging [26] and others. To model workflows in a service-oriented computing (SOC) environment a composition language like BPEL is required. Furthermore, transactional behavior is beneficial for the quality of the business processes. For example, BPEL prefers the use of WS-Transaction [9], which we will focus on, when defining service modeling requirements. To summarize the aspects that have to be discussed when modeling medical e-services, we find

- a high degree of vertical standardization through DICOM, HL7 and IHE
- currently implemented systems based on conventional middleware
- lack of interorganizational workflow support as a common problem
- no current Web service based approach which tries to fill this gap

Therefore we suggest a Web service based workflow model that implements IHE conformant transactions to provide medical e-services functionality in a mixed Intranet/Internet environment.

2.4 Related work

Most information related to medical e-services can be found in the corresponding standardization documents for HL7, DICOM and IHE. Similarly, all current standards related to the Web service stack are available. More specifically, a discussion of an interorganizational workflow in the medical imaging domain can be found in [27]. A first approach of Web service definition and middleware design for the medical imaging domain can be found in [28]. The paper covers the separation of the workflow layer, using BPEL [11] and WSDL [7], and the domain layer, using DICOM and HL7. Additionally, it performs a mapping between BPEL activities and DICOM and HL7 messages.

3 Requirements for medical e-service modeling

In this section we cover requirements that have to be met when modeling medical e-services. We outline the relationship of HL7, DICOM and IHE concepts to Web service modeling (especially BPEL) constructs. Additionally, we discuss the impact and usefulness of current Web service stack protocols.

3.1 HL7 and DICOM encoding

When implementing medical e-services using Web service technology, we have to consider transferring HL7 and DICOM messages using XML and SOAP. One solution is a conversion of messages and binary data into XML. Another more advantageous approach is to simply attach original messages to SOAP messages and to only use identifiers and other attributes required for a proper workflow execution within the SOAP message. A third approach is to separate workflow and domain communication, with the disadvantage of an additional communication channel inappropriate for a firewall based Internet environment (see Figure 1). In this paper we focus on attaching HL7 and DICOM data, the second approach and on workflow modeling with attributes required for its execution.

When using attachments we have to consider techniques where HL7 and DICOM data has to be transferred together with the workflow messages. Because both standards define binary data types, an encapsulation and payload transfer should be supported. Several techniques are available like WS-Attachments [29] based on DIME [30] or SOAP Messages with Attachments (SwA) as described in [31]. More recently, the SOAP 1.2 [6] specification supports base64binary encoding [32] of data and is currently evolving as the standard mechanism for transferring binary data as it doesn't require additional protocol parsers. Furthermore, security as in WS-Security can be applied on binary data too. However, a modeling process has to provide techniques to transfer HL7 and DICOM messages over a Web service infrastructure.

3.2 Data and Service identification

First, a clear identification of messages and data items is required. A necessary similarity between the HL7 and DICOM protocols is that they contain message identifiers (message ID for HL7 and association ID for DICOM). Furthermore, the data exchanged is identified by system wide identifiers (patient ID, visit ID, image ID). DICOM objects and HL7 messages use different definitions and identifiers for data items. Related to our example in section 4, the *patient registration* transaction messages are identified by the PID-3 (Patient identifier list) and the PV1-19 (Visit number) HL7 segment attributes. The DICOM *modality worklist* service uses patient UIDs, examination IDs and others.

Fortunately, IHE chooses the more specific protocol for a given situation. It defines a mapping between identifiers used in HL7 and DICOM and describes usage conventions to provide interoperability of the standards. The standardization effort lets us easily select the message segment IDs (HL7) or object modules UIDs (DICOM) suggested by IHE in each modeled IHE transaction. For service identification the unique IHE transaction name (e.g. patient registration) can be used. This identification is required by Coordination and Registration protocols as described in the next sections. A modeling process should select identifiers from the standard documents and provide a mapping between an IHE transaction and its Web service.

3.3 Web service Coordination

When using Web services, the coordination of business partners is required for distributed activities. Currently, the main purposes of coordination protocols like WS-Coordination [8, 33] or other approaches [34-36] are reliable messaging, transactions and security. For medical e-services business partners are correlated by IHE transactions. Each of these transactions might be executed between two participants requiring transaction or security services. It has to be stated, that not every IHE actor might be a separate application. Therefore, participants are normally not 1:1 related to an actor. However, the IHE actor's name perfectly expresses the role in an IHE transaction.

To support coordination protocols unique identifiers are required. These identifiers are used by coordinators to define a coordination-context for the participants. As stated above, IDs for messages and transactions can be derived from the standards. Nevertheless, process instances that register coordination-contexts might use the same messages and transactions during their execution which makes these IDs improper. A unique ID generator must be used instead.

To coordinate service instances, information about used ports (service endpoint) can be extracted from the WSDL definition. Furthermore, specific roles, like master or slave in a 2PC transaction, might be required by the coordinator. However, BPEL uses a different transaction mechanism based on compensation, which better fits to the definition of an IHE transaction as outlined in the next section. For security purposes, service participants might define a security context. As for transactions, unique identifiers are required and have to be generated. For reliable messaging services, like WS-ReliableMessaging, there are additional message sequence numbers, which have to be generated by the middleware like context identifiers. Furthermore, medical e-services require delivery semantics of *ExactlyOnce* and *InOrder*, because the IHE framework only mentions messages delivered accordingly. The behavior for messages that are out of sequence is undefined. For example, the Collaxa BPEL Server [37] product contains support for reliable messaging in a delivery service module. Furthermore, it uses WS-Addressing [38] to handle the correlation of asynchronous messages.

A modeling process has to cover the appliance of transaction and security attributes to IHE transactions. Additionally, compensation activities have to be identified.

3.4 Web service transactions

Transaction protocols are used to increase the quality of a Web service based business process to the standards already provided by conventional middleware. Currently the most important standards are WS-Transaction and more recent but not yet widely used WS-TransactionManagement [39]. In general, there are different transaction models for direct, queued and compensation-based transaction processing [40]. For our infrastructure, we consider the use of BPEL and therefore a compensation-based approach. In Compensation-based Transaction Processing compensating actions are executed to "undo" the effects of actions that have been successfully completed [11]. More information on Web service transactions can be found in [41, 42].

DICOM and HL7 basically don't specify any transactional behavior. The application logic takes care that, for example, payments are not booked twice. With the introduction of an IHE based Web service middleware it is feasible to provide transaction services. As their name suggests, IHE transactions provide a granularity of activities useable for a transaction context. To implement a compensation-based model, compensation actions for IHE transactions have to be defined. Some transactions perform only read operations and therefore don't require any transactional semantics. A modeling process should provide a guideline to decide transactional behavior based on the operations executed in the IHE transaction.

As an example for compensation-based transaction processing, the *patient registration* transaction uses a HL7 ADT^A01 or A04 message to register a patient. In case of an error in the sending application, the registration process has to be undone with the A11 cancel message. If a patient is pre-registered (A05) the A38 cancel message is used. We provide a model of this example in section 4. As a second example, the *modality worklist provided* is read-only and therefore has no compensation activity. Models like the Direct Transaction Processing using the 2-phase commit (2PC) protocol and the Queued Transaction Processing used in queue-based middleware systems are currently inappropriate for the modeling of BPEL processes. For example, the Collaxa BPEL Server [37] contains support for WS-Transactions and executes compensation activities defined in the BPEL workflow model.

A modeling process has to integrate compensation activities into the composed Web service. Transactions should be part of the modeling phase and not applied afterwards.

3.5 Web service security

Several requirements for security have to be met when modeling medical e-services, because the data transferred is often highly confidential. For Internet-based infrastructures as outlined in Figure 1 existing standards in the medical industry [2, 3, 4, 5, 43] require strong encryption with a minimal key length of 128bit and authentication based on asymmetric keys. WS-Security supports username/password security, X.509 certificates, Kerberos authentication or SSL. It only defines the SOAP encoding of these standards. An established infrastructure for the authentication and encryption process has to be in place. If trust relationships as defined in WS-Trust [44] are used, an additional infrastructure for a Security Token Service is required. In WS-SecureConversation [45] Web service providers specify security requirements and requestors provide claims that can be matched prior to security establishment. The standard also states which parts of a SOAP message have to be signed and encrypted to avoid message tampering and ensure the privacy of the communication partners.

The gateways (Figure 1) used to transfer data via SOAP have to implement these security standards. An IHE transaction is performed between two actors; intermediaries are not mentioned in this context. For each transaction a security context has to be defined. For modeling purposes it is reasonable to use the same granularity of an IHE transaction as in WS-Transaction. In the case of using HL7 and DICOM as attachments in SOAP messages, WS-Security provides a specification of how this data has to be encrypted additionally. Because DICOM data can be very large (several 100MB) an application-level encryption using WS-Security might be infeasible. In such cases encryption can only be applied to the remaining part of the SOAP message. Another possibility is the use of transport layer security like TLS [46] or IP-sec [47], besides there are implementation difficulties in Internet-based scenarios.

On the other hand, the IHE standard itself defines transactions for a Kerberos service. The messages could be exchanged as supposed in 3.1. However, just few medical applications support this. Therefore, it might be necessary to provide an infrastructure based completely on Web service standards. A modeling process should at least identify security attributes for IHE transactions. How these transactions are secured in a specific scenario might be postponed to the implementation.

3.6 Web service registration and binding

The UDDI standard [48] specifies Web services for service registration, subscription and binding. UDDI stores information about companies, services in general and Web services in particular in a 1:n relationship. For our purpose, the registry can provide yellow pages and green pages services. The former can be used to search for a service that implements specific IHE transactions. The latter is required to bind to the service at run-time. There is a private and a public model to distribute UDDI registries. We consider a private model where a registry is maintained by one participant of an IHE transaction. UDDI supports a security model for the communication with and the manipulation within the registry. Because the gateway (see Figure 1) already requires a security infrastructure, securing the registration service is reasonable.

For yellow pages, the IHE framework can be mapped to the registry by creating entries for IHE applications (services) and IHE Web services. Furthermore, a classification scheme is supported and can be used in the IHE context by classifying applications for their support of IHE actor (classes), and IHE Web services for their support of IHE transaction (classes). There is not necessarily a 1:1 relationship between a Web service and an IHE transaction. For green pages, the binding process can be implemented at design-time or at run-time. For workflows based on the IHE standard run-time binding is required, if a decision for a specific IHE actor is made on a process instance base. This is the case, for example, if a report for an examination is created by a physician based on the patient's diagnosis. The dynamic binding depends on attributes like modality name and requesting physician (DICOM) or referring doctor and assigned patient location (HL7). All attributes, required for dynamic binding, have to be modeled in BPEL.

There is currently no mechanism for service registration in IHE. A modeling process should address service binding requests for a selection beyond different IHE actors or run-time decisions within a process as mentioned above.

3.7 Web service composition

For Web service composition we have to consider the structure and granularity of a Web service to be a suitable part of the executed workflow. The following table provides a mapping between IHE concepts and BPEL language constructs that will be discussed further.

IHE concept	BPEL construct
IHE actor	BPEL partner
IHE flow	BPEL process
IHE transaction	BPEL service link, 2 BPEL ports
HL7 message/DICOM service	1 BPEL invoke+receive activity, 1 SOAP message
HL7 message/DICOM service	1 BPEL compensation activity

Tab. 1. Relationship of IHE concepts and BPEL constructs

An IHE actor is modeled as a BPEL business partner. Applications might perform one or more roles and therefore participate in different BPEL processes. An IHE flow, like the *administrative process flow* is modeled as a BPEL process (see section 4). An IHE transaction is mapped on a BPEL service link, where only two business partners are communicating with each other over two BPEL ports. A single HL7 message or DICOM object is embedded in a SOAP message and transferred between the business partners using a BPEL invoke and receive activity. As stated above, BPEL uses WS-Transactions and a compensation mechanism. Compensation activities themselves are implemented as HL7 messages and DICOM objects.

3.7.1 BPEL variables

To specify a BPEL process, variables have to be defined, that are required for the workflow. For medical e-services they consist of the following four categories. First, we require environment attributes for the participating IHE actors and the implemented IHE transactions. This information is stored during composition in the BPEL server itself or for dynamic binding in a UDDI registry. For dynamic binding attributes suggested in section 4.6 (requesting physician, etc.) have to be stored additionally. The second category are attributes used to identify the message type (HL7 ADT^A01, etc.) and message content (patient UID, etc.). All message content identifying attributes are used to construct a BPEL correlation set. The third category consists of attributes used in state information and BPEL expressions. For example, the HL7 PatientClass is used to control the process flow of the patient registration transaction. The last category are the remaining attributes that reside only in the payload and are not part of the BPEL definition.

3.7.2 Basic activities

BPEL uses basic activities to execute the workflow between business partners. In e-services IHE transactions are executed by performing HL7 and DICOM operations. For each operation between two partners the initiating part executes an *invoke* activity on a defined BPEL port and the receiving partner performs a corresponding *receive* activity on another port. The ports are related in a BPEL service link associating the business partners. The modeling process in section 5 provides a corresponding example. The paper in [28] provides details of this relationship for a medical workflow. Another approach focusing on a supply chain example can be found in [49].

3.7.3 Expressions and structured activities

BPEL uses expressions for conditions and variable assignment using extensions of the X-Path [50] standard. Variables of the first three categories can be used in expressions. For example, the HL7 PatientClass can be used in a boolean expression. BPEL supports among other things *sequence*, *switch* and *while* activities to structure the process. A model of these activities can be partially derived from the sequence diagrams provided in the IHE framework. As shown in the example of Figure 4 an A01, A04 and A05 message can be sent depending on the HL7 PatientClass, therefore a switch construct is used within the process. The modeling process in section 4 provides a corresponding example. For other examples refer to [28, 49]. A detailed analysis of BPEL patterns can be found in [51].

3.7.4 Message correlation and correlation sets

The messages sent and received in an IHE transaction have to be correlated by an unique identifier, a BPEL correlation set. This set can be construed by appending all identifying HL7 and DICOM message attributes and depends on the structure of the underlying messages exchanged. Examples can be found in the Appendix. In general, the attributes are derived from the standardization documents for each transaction.

3.7.5 Scopes and compensation activities

A scope is a BPEL construct used for error or compensation handling. Compensation handlers can be defined on a scope level to perform compensation activities in case of application level errors. Compensation activities can also be used in error handlers for system level errors. As mentioned earlier, some of the IHE transactions activities require compensation and some do not. This information has to be derived from the respective standard documents. For the patient registration example the HL7 A01 message has to be compensated by an A11 message. The granularity of an IHE transaction is a candidate for defining scopes as its outcomes are defined clearly within the IHE framework. Further modeling examples should proof this assumption. Currently there is no evidence for the use of nested transactions.

3.8 Conclusions for a modeling process

For a mixed Intranet/Internet environment as introduced in Figure 1 we require a Web service infrastructure. However, IHE doesn't mention Web services. Nevertheless, IHE defines workflow transactions that can be mapped directly to a Web service composition language like BPEL. Furthermore, IHE defines compensation activities and a Kerberos infrastructure which narrows down modeling decisions related to security and transactions. As a first step to implement IHE transactions in a Web service infrastructure, we provide a modeling process for BPEL in the next section.

4 Outlining a modeling process

The modeling process is separated into four steps. First, we provide the four layer model to structure the content of the IHE framework. In the second step the process flow is defined and normalized. In the third step a similar approach is performed for the IHE transactions. Finally, based on the normalized descriptions BPEL and WSDL definitions are derived.

4.1 Definition of a 4 level Use-Case model

The first step for modeling medical e-services is the definition of 4-level UML [52, 53] Use-Case model. This model has been introduced in [20] and is shown in Figure 2.

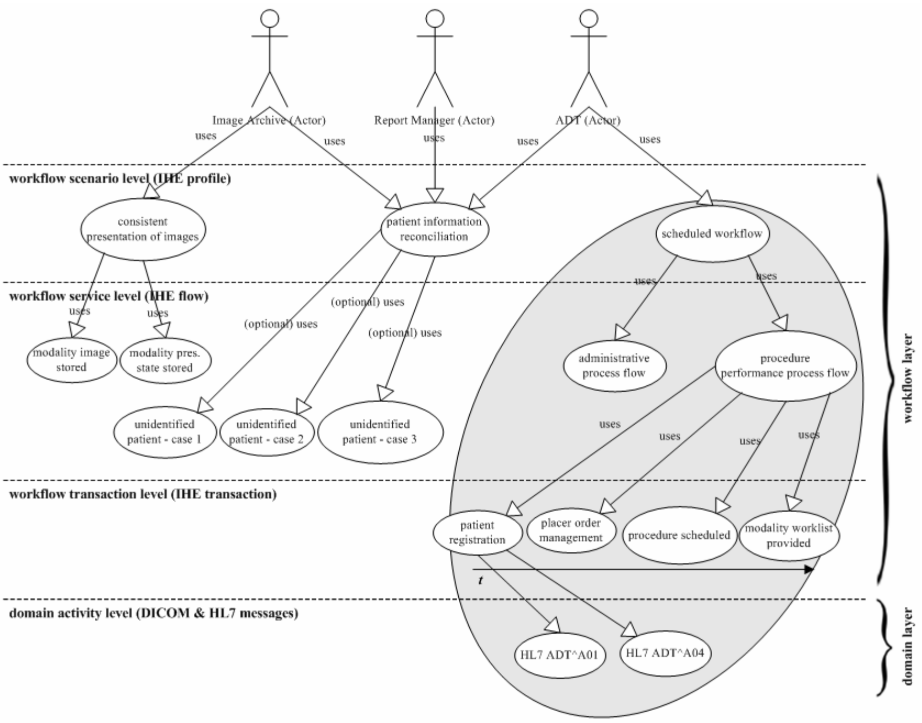

Fig. 2. Modeling process - 4-level Use-Case model

The layers used correspond to the definitions for profiles, flows, transactions and messages used in the IHE framework. On the top layer the IHE integration profiles are shown, a coarse grained overview of what an application performs. The *IHE Scheduled Workflow profile* we focus on is shown in the gray shaded area. These profiles are split into several flows. Each flow must be supported by an application that implements the profile (in our example the *administrative process flow*). IHE flows are defined as sequence diagrams in the IHE framework. Each IHE flow is further defined using several IHE transactions. These transactions are sequentially order and not all transactions of a flow have to be implemented by every participating actor. Finally, a transaction consists of one or several HL7 and DICOM messages that have to be sent or received. The upper three levels correspond to the workflow layer of the middleware, while the forth resides in the domain layer. While conventional workflow systems focus on the third and forth layer our approach takes the structure of the whole IHE specification into account. For readability different Use-Case models should be created to focus on the implemented IHE actors of a specific application. The IHE transactions that have to be modeled in the next step can be depicted from layer 3. For designing medical e-services we further focus on the *IHE administrative process flow*. The Use-Case model for medical workflows has been introduced in [20].

4.2 Selection, Definition and Normalization of process flow

In a second step we can proceed to focus on the *administrative process flow* and provide an activity diagram (Figure 3) that corresponds to the public workflow for the department system scheduler / order filler IHE actor and is derived from the corresponding sequence diagram defined in the IHE framework [4].

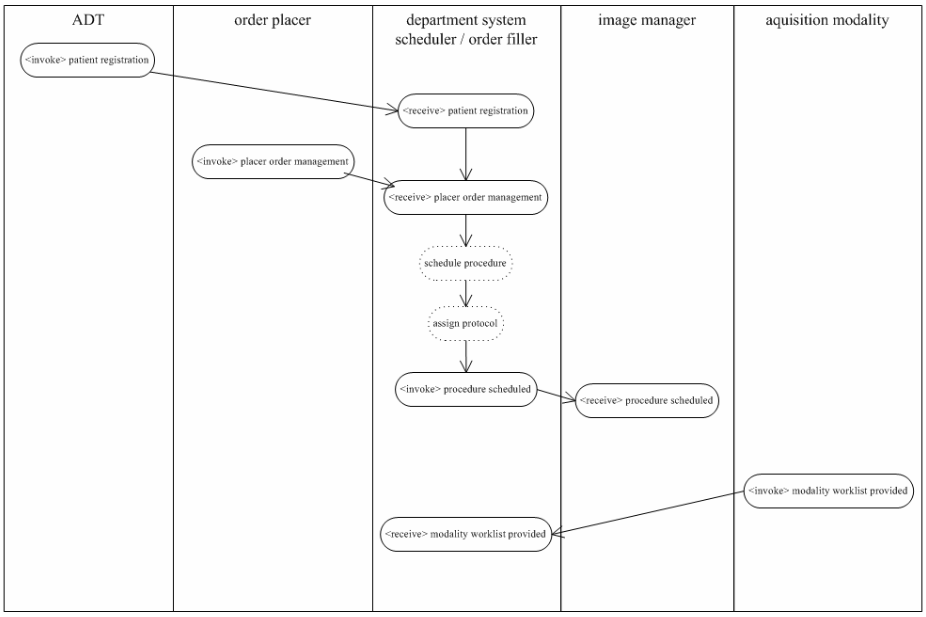

Fig. 3. *Administrative process flow* - public process of Department System Scheduler

UML activity diagrams are widely used as a representation language for workflows as discussed in [54]. The public process contains all activities (IHE transactions) performed by the IHE actor, internal operations are shown for readability. The diagram can be derived from the sequence diagram by performing several normalization operations.

First, an IHE actor, to define the public process for, is selected and actor independent and internal operations are deleted. Next, IHE transactions are translated into BPEL invoke and receive activities. Caution has to be taken, because IHE defines some of the transactions in the wrong direction. For example, the DICOM service used in the *modality worklist provided* transaction is shown as been executed from by department system scheduler on the acquisition modality. However, it is the client (acquisition modality) that queries a server during this operation, therefore the invoke activity is performed by the acquisition modality. Furthermore, the conversion results in two independent processes, therefore an IHE flow not necessarily corresponds 1:1 to a BPEL process. As another fact, an application might implement several roles in the IHE flow, therefore converting external transactions to internal which are not modeled in a BPEL process. To join two actors, the *invoke* and *receive* activities between them are converted to internal operations and omitted. The two sets of other activities are joined. The diagram outlines requirements of the process to implement. However, a BPEL process can not be directly derived because details of the underlying domain layer are omitted. These details are provided in the next step.

4.3 Selection, Definition and Normalization of transactions

In a third step we focus on the activities performed in an IHE transaction. The HL7 and DICOM messages exchanged between two systems in a *patient registration* transaction are outlined in Figure 4.

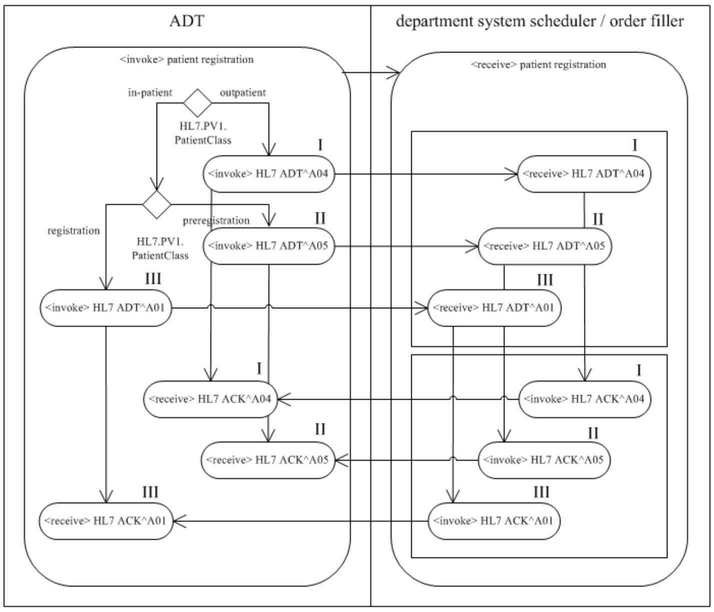

Fig. 4. *Patient registration* transaction - public process of Department System Scheduler

The activity diagram corresponds to the sequence diagram of the *patient registration* transaction defined in [4]. The diagram is a more detailed view of the IHE flow above. The simplified *invoke* and *receive* activities of Figure 3 might now be split into one or more BPEL activities. The *invoke* operation is annotated in the flow at the initial sender of the transaction (the ADT actor in our example).

Several implications for an implementation have to be depicted from the standard documents to normalize the activity diagram. For example, the patient registration distinguishes in-patient, outpatient and pre-registration. These cases depend on the PatientClass attribute of the PV1 segment of HL7 ADT messages. In the BPEL process this results in a *switch* structured activity. This implies several initiating *receive* activities for the process of the department system scheduler. BPEL supports multiple start activities by setting the *createinstance* attribute of these activities to “yes”. Furthermore, HL7 requires acknowledge messages to be sent back to the initiator. These are modeled using an additional pair of *invoke* and *receive* activities.

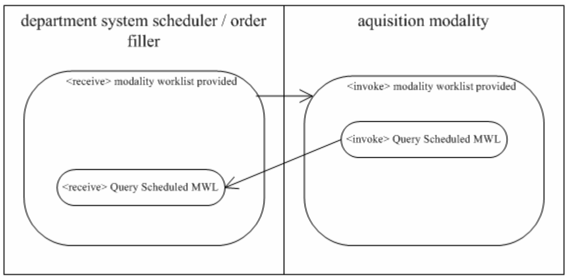

Fig. 5. *Modality worklist prov*. transaction - public process of Department System Scheduler

As another example, Figure 5 shows the activities performed in a modality worklist provided transaction. The operation is simply converted into a pair of *invoke* and *receive* activities. No additional steps are necessary.

4.4 Definition of BPEL process

In the next step we are able to derive a BPEL process specification from the provided activity diagrams for the *patient registration*. In short the following tasks are necessary. The BPEL specification contains definitions of *types*, *variables*, *messages* and *correlationSets* that can be derived from DICOM and HL7. Furthermore, business *partners* and a *process* using basic and structured activities are defined. The WSDL file contains a *portType* and a *serviceLinkType* section to define the Web services. Finally, compensation activities are provided using scopes and security issues are outlined. A complete description of this step is provided in the Appendix.

5 Conclusions and Future work

In this paper we have introduced the medical services domain, defined requirements for designing medical e-services and outlined a Web service modeling process for IHE framework transactions.

However, several points remain unsolved in this context. First, the transfer and encryption of large binary data is an open issue. For Web service security it is not clear, whether a Web service or an IHE based infrastructure should be preferred. Next, some of the standard specifications of the Web service stack are not yet widely implemented or, especially for coordination services, competing standards exist. Therefore, this paper focused on the composition of medical e-services. Design implications for the areas of coordination, security, transaction and service binding have to be defined in more detail. Those standards are still subject to change and the implications on an infrastructure for e-services have to be revised subsequently. Finally, there are problems normalizing activity diagrams resulting of ambiguities in the medical industry standards.

From here, there are several directions to proceed in future work. On the one hand, one or more existing Web service infrastructures can be used to evaluate implementation specific issues of the BPEL process example. Further evaluations should add security, service binding and other features to show the usability in more complex scenarios. On the other hand, the modeling process, especially the mapping between IHE, HL7 and DICOM standard definitions on one hand and UML diagrams and BPEL constructs on the other, has to be defined formally. Finally, evaluation results and requirements for a Web service based infrastructure should result in an architecture for the execution of medical e-services.

References

1. Workflow Management Coalition: WFMC Reference Model, www.wfmc.org (1995)
2. HL7 Organization: Health Level 7, http://www.hl7.org (2000)
3. NEMA and Global Engineering Group: DICOM 3 Standard, http://www.nema.org (1998)
4. Radiological Society of North America: IHE Technical Framework 1.1, http://www.rsna.org/IHE/index.shtml (2003)
5. CEN/TC251 Health informatics - Medical data interchange: HIS/RIS-PACS and HIS/RIS Modality Interface - ENV 13939, http://www.centc251.org/ (2001)
6. W3C: SOAP Version 1.2, http://www.w3.org/TR/soap12-part1/ (2003)
7. W3C: Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl.html (2001)
8. BEA, IBM, Microsoft: Web Services Coordination (WS-Coordination), http://www.ibm.com/developerworks/library/ws-coor/ (2002)
9. BEA, IBM, Microsoft: Web Services Transactions (WS-Transactions), http://www.ibm.com/developerworks/library/ws-transpec/ (2002)
10. BEA, IBM, Microsoft: Web Services Security (WS-Security), www-106.ibm.com/developerworks/ webservices/library/ws-secure/ (2002)
11. BEA Systems, IBM, Microsoft, SAP AG and Siebel Systems: Business Process Execution Language for Web Services version 1.1, http://www-106.ibm.com/developerworks/library/ws-bpel/ (2003)
12. Siemens Medical e-services: http://www.medical.siemens.com
13. Philips: http://www.medical.philips.com
14. GE Medical Systems: http://www.gemedicalsystems.com
15. Agfa Healthcare: http://www.agfa.com/healthcare/
16. Kodak Medical: http://www.kodak.com/global/en/health/
17. Anzböck, R.: XR OPEN RIS Architektur, D.A.T.A. Corporation, http://www.data.at (2001)
18. Anzböck, R.: XR PACS Architektur, D.A.T.A. Corporation, http://www.data.at (2001)
19. Kreider, N.A., Haselton, B.J.: The Systems Challenge: Getting the Clinical Information Support You Need to Improve Patient Care, Wiley, John & Sons, Incorporated (1997)
20. Siegel, Eliot, Kolodner, Robert M.: Filmless Radiology, Springer (1998)
21. Huang, H. K.: PACS: Basic Principles and Applications, Wiley-Liss (1998)
22. Revet, Bas: DICOM Cookbook, Philips Medical Systems (1997)
23. Oosterwijk, H.: DICOM Basics, OTech Inc/Cap Gemini Ernst and Young
24. Aalst: Interorganizational Workflows: An approach based on Message Sequence Charts and Petri Nets**,** citeseer.nj.nec.com/vanderaalst99interorganizational.html **(**1999**)**
25. ebpml.org: The Web service stack, http://www.ebpml.org/webservices.htm (2003)
26. BEA, IBM, Microsoft, TIBCO: WS-ReliableMessaging, http://www-106.ibm.com/developerworks/webservices/library/ws-rm/ (2003)

27. Anzböck, R., Dustdar, S.: Interorganizational Workflow in the Medical Imaging Domain. Proceedings of the 5th International Conference on Enterprise Information Systems (ICEIS), Angers, France, Kluwer Academic Publishers (2003)
28. Anzböck, R., Dustdar, S.: Medical e-services workflows with BPEL4WS, http:// http://www.infosys.tuwien.ac.at/Staff/sd/papers/MedicalServicesWorkflowsWithBPEL4WS.pdf (2003)
29. WS-Attachments: http://msdn.microsoft.com/library/en-us/dnglobspec/html/draft-nielsen-dime-soap-01.txt (2002)
30. Microsoft: Direct Internet Message Encapsulation (DIME), http://msdn.microsoft.com/library/en-us/dnglobspec/html/draft-nielsen-dime-02.txt (2002)
31. W3C, SOAP Messages with Attachments: http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211 (2000)
32. "Base64 Content-Transfer-Encoding," RFC 2045, Section 6.8, IETF Draft Standard (1996)
33. Alonso, Casati, Kuno, Machiraju: Web Services, Springer (2004)
34. Bunting, Chapman, Hurley, Little, Mischkinsky, Newcomer, Webber, Swenson: Web Services Composite Application Framework Version 1.0 (WS-CAF), http://www.iona.com/devcenter/standards/WS-CAF/ (2003)
35. Arjuna, Fujitsu, IONA, Oracle, Sun, WS-CTX: Web Services Context, developers.sun.com/techtopics/webservices/wscaf/wsctx.pdf (2003)
36. Arjuna, Fujitsu, IONA, Oracle, Sun, WS-CF: WS-Coordination Framework, developers.sun.com/techtopics/webservices/wscaf/wscf.pdf (2003)
37. Collaxa Inc.: Collaxa BPEL Server 2.0: Reviewer's Guide, http://www.collaxa.com/pdf/cx-bpel-review-20.pdf (2003)
38. BEA, IBM, Microsoft: Web Services Addressing (WS-Addressing), http://www-106.ibm.com/developerworks/webservices/library/ws-add/ (2003)
39. Arjuna, Fujitsu, IONA, Oracle, Sun: Web Services Transaction Management (WS-TXM), http://developers.sun.com/techtopics/webservices/wscaf/wstxm.pdf (2003)
40. Tai, Mikalsen, Wohlstadter, Desai, Rouvellou: Transaction Policies for Service-Oriented Computing (2003)
41. Frolund, Govindarajan: Transactional conversations. In Proceedings of the W3C workshop on Web services, San Jose, CA, USA (2001)
42. Mikalsen, Tai, Rouvellou: Transactional attitudes: Reliable composition of autonomous Web services. In Workshop on Dependable Middleware-based Systems, WDMS 2002, Washington D.C., USA (2002)
43. STRING Kommission, Magda-Lena 2 Richtlinie: http://www.akh-wien.ac.at/STRING/ (2000)
44. WS-Trust, IBM, Microsoft, Verisign, RSA Security: www-106.ibm.com/developerworks/library/ws-trust/ (2003)
45. WS-SecureConversation, IBM, Microsoft, Verisign, RSA Security: www-106.ibm.com/developerworks/library/ws-secon/ (2003)
46. IETF, The TLS Protocol Version 1.1: http://www.ietf.org/internet-drafts/draft-ietf-tls-rfc2246-bis-05.txt (2003)
47. IETF, The IP Security Protocol: http://www.ietf.org/html.charters/ipsec-charter.html (1995)
48. IBM/Microsoft/SAP, et.al.: UDDI 3.0.1, http://uddi.org/pubs/uddi-v3.0.1-20031014.pdf (2003)
49. Mantell: From UML to BPEL, Model Driven Architecture in a Web services world, www-106.ibm.com/developerworks/webservices/library/ws-uml2bpel/ (2003)
50. W3C, X-Path: http://www.w3.org/TR/1999/REC-xpath-19991116 (1999)
51. Wohed, P., van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M.: Pattern Based Analysis of BPEL4WS, Department of Computer and Systems Sciences, Stockholm University/The Royal Institute of Technology, Sweden (2003)
52. Object Management Group: UML 2.0 Standard specification, http://www.omg.org (2003)
53. Fowler, Martin, Scorr, Kendall: UML destilled, Addison Wesley Professional (2000)

54. Dumas, ter Hofstede: UML Activity Diagrams as a Workflow Specification Language, Proceedings of the International Conference on the Unified Modeling Language (UML'2001), Toronto, Canada (2001)

Appendix

First, we provide an overview of the BPEL business process specification. Then we provide a WSDL definition for the port types, service link types and messages. Furthermore, we apply transaction, security and error handling capabilities. The complete BPEL and WSDL code for the *register patient* transaction is listed at the end of the section.

Definition of a BPEL specification

The workflow process is specified as a BPEL file with a predefined structure. We provide the main sections of the specification. First, the *partners* section covers the implementing application, the supported servicelinktype and the role that is performed in the process.

```
<partners>
  <partner name="app X"
    serviceLinkType="IHETransPatientRegistration"
  myRole="IHEActorADT"/>
</partners>
```

Next, two sections contain message property type definitions used throughout the communication. For example the A01 message is outlined.

```
<definitions name="properties"
  <bpws:property name="patientId" type="xsd:string"/>
  <bpws:property name="registrationProcessId" type="xsd:string"/>
</definitions>

<types>
  <xsd:schema>
    <xsd:complexType name="HL7_A01_TYPE">
      <xsd:element name="patientId" type="xsd:string"/>
      <xsd:element name="registrationProcessId" type="xsd:string"/>
      <xsd:element name="payload" type="xsd:string"/>
...
```

Furthermore, two sections contain the variable and message definitions used throughout the process. For example the A01 messages and acknowledgement are outlined.

```
<variables>
  <variable name="HL7_A01_VAR" messageType="HL7_A01_MSG">
  <variable name="HL7_A01_ACK_VAR" messageType="HL7_A01_ACK_MSG">
...

<message name="HL7_A01_MSG">
  <part name="HL7_A01_PART" type="HL7_A01_TYPE"/>
...
```

An additional definition for the correlation sets is required. The correlation is defined by the unique patient identifier and a registration process id.

```
<correlationSets>
  <correlationSet name="HL7_A01_CS" properties="patientId registrationProcessId"/>
...
```

Finally, the flow section contains the process definition itself. The example shows the switch statement corresponding to the sequence diagram, the invoke and receive activities for the A01 messages and the used correlation set

```
     <flow>                                              <!–- patient registration flow -->
       <switch>                                          <!-- switching PatientClass -->
       <case HL7_A01.PV1.PatientClass=in-patient>         <!–- in-patient -->
         <switch>                                        <!-- switching PatientClass -->
         <case HL7_A01.PV1.PatientClass=registration>     <!–- registration -->
           <invoke partner="app Y" portType="IHETransPatientRegistrationCallbackPort"
            operation="HL7_A01" inputvariable="HL7_A01_MSG">
             <correlations>
               <correlation set="HL7_A01_CS">
             </correlations>
...
```

Definition of a WSDL specification

Beside the process it is also necessary to define the Web service communication ports using a WSDL description. The main parts, the *portType* and the *serviceLink-Type* sections are outlined here. A patient registration port and callback port are specified together with the required messages exchanged.

```
<portType name="IHETransPatientRegistrationPort">
   <operation name="HL7_A01">
     <input message="HL7_A01_MSG">
   </operation>


<serviceLinkType name="IHETransPatientRegistration">
   <role name="IHEActorADT">
     <portType name="IHETransPatientRegistrationPort">
   </role>
  <role name="IHEActorDepartmentSystemScheduler">
     <portType name="IHETransPatientRegistrationCallbackPort">
  </role>
</serviceLinkType>
```

The complete BPEL definition for the register patient transaction can be found in [58].

Applied transactions and security

To complete the modeling process requirements on transaction and security have to be applied. As stated in section 4, a transaction and security context has to be generated for each IHE transaction. Compensation-based transactions are supported in BPEL using the Scope section of the process. In our example a A01 message is compensated by a A11 message as outlined below.

```
<scope>
  <compensationHandlers>
    <invoke partner="app Y" portType="IHETransPatientRegistrationCallbackPort"
     operation="HL7_A11" inputvariable="HL7_A11_MSG">
      <correlations>
        <correlation set="HL7_A11_CS">
      </correlations>
    </invoke>
    <receive partner="app Y" portType="IHETransPatientRegistrationPort"
     operation="HL7_A11_ACK" inputvariable="HL7_A11_ACK">
      <correlations>
        <correlation set="HL7_A11_CS">
      </correlations>
    </receive>
  </compensationHandlers>

  <invoke partner="app Y" portType="IHETransPatientRegistrationCallbackPort"
```

```
   operation="HL7 A01 admit visit notification" inputvariable="HL7_A01">
    <correlations>
      <correlation set="HL7_A01_CS">
    </correlations>
  </invoke>
  <receive partner="app Y" portType="IHETransPatientRegistrationPort"
   operation=" HL7 A01 acknoledge" inputvariable="HL7_A01_ACK">
    <correlations>
      <correlation set="HL7_A01_CS">
    </correlations>
  </receive>
</scope>
```

Security semantics are not part of the BPEL specification and has to be implemented in a separate middleware component. The component has to intercept the SOAP engine and attach Security header into the messages by parsing the messages generated by the BPEL engine. A security module requires information about the target business partner, the preformed process. On the backend the module has to communicate with the security infrastructure like the Security Token Service.

Code listing

BPEL code

```
<!-- definitions -->
<definitions name="properties"
  <bpws:property name="patientId" type="xsd:string"/>
  <bpws:property name="registrationProcessId" type="xsd:string"/>
</definitions>

<!-- type definitions -->
<types>
  <xsd:schema>
    <xsd:complexType name="HL7_A01_TYPE">
      <xsd:element name="patientId" type="xsd:string"/>
      <xsd:element name="registrationProcessId" type="xsd:string"/>
      <xsd:element name="payload" type="xsd:string"/>
...

<!-- message definitions -->
<message name="HL7_A01_MSG">
  <part name="HL7_A01_PART" type="HL7_A01_TYPE"/>
...

<!-- partner definitions -->
<partners>
  <partner name="app X"
    serviceLinkType="IHETransPatientRegistration"
  myRole="IHEActorADT"/>
</partners>

<!-- variable definitions -->
<variables>
  <variable name="HL7_A01_VAR" messageType="HL7_A01_MSG">
  <variable name="HL7_A01_ACK_VAR" messageType="HL7_A01_ACK_MSG">
  <variable name="HL7_A04_VAR" messageType="HL7_A04_MSG">
  <variable name="HL7_A04_ACK_VAR" messageType="HL7_A04_ACK_MSG">
  <variable name="HL7_A05_VAR" messageType="HL7_A05_MSG">
  <variable name="HL7_A05_ACK_VAR" messageType="HL7_A05_ACK_MSG">
  <variable name="HL7_A11_VAR" messageType="HL7_A11_MSG">
  <variable name="HL7_A11_ACK_VAR" messageType="HL7_A11_ACK_MSG">
</variables>

<!-- workflow definition -->
<flow>                                               <!—- patient registration flow -->
  <switch>                                          <!-- switching PatientClass -->
  <case HL7_A01_VAR.PV1.PatientClass=in-patient>     <!—- in-patient -->
    <switch>                                        <!-- switching PatientClass -->
    <case HL7_A01_VAR.PV1.PatientClass=registration> <!—- registration -->
      <scope>
        <compensationHandlers>
          <invoke partner="app Y" portType="IHETransPatientRegistrationCallbackPort"
           operation="HL7_A11" inputvariable="HL7_A11_VAR">
            <correlations>
              <correlation set="HL7_A11_CS">
            </correlations>
```

```
              </invoke>
              <receive partner="app Y" portType="IHETransPatientRegistrationPort"
               operation="HL7_A11_ACK" inputvariable="HL7_A11_ACK_VAR">
                <correlations>
                  <correlation set="HL7_A11_CS">
                </correlations>
              </receive>
            </compensationHandlers>

            <invoke partner="app Y" portType="IHETransPatientRegistrationCallbackPort"
             operation="HL7_A01" inputvariable="HL7_A01_VAR">
              <correlations>
                <correlation set="HL7_A01_CS">
              </correlations>
            </invoke>
            <receive partner="app Y" portType="IHETransPatientRegistrationPort"
             operation="HL7_A01_ACK" inputvariable="HL7_A01_ACK_VAR">
              <correlations>
                <correlation set="HL7_A01_CS">
              </correlations>
            </receive>
          </scope>
        </case>                                           <!-- registration -->
        <case HL7_A01.PV1.PatientClass=preregistration>     <!—- preregistration -->
          <invoke partner="app Y" portType="IHETransPatientRegistrationCallbackPort"
           operation="HL7_A05" inputvariable="HL7_A05_VAR">
            <correlations>
              <correlation set="HL7_A05_CS">
            </correlations>
          </invoke>
          <receive partner="app Y" portType="IHETransPatientRegistrationPort"
           operation="HL7_A05_ACK" inputvariable="HL7_A05_ACK_VAR">
            <correlations>
              <correlation set="HL7_A05_CS">
            </correlations>
          </receive>
        </case>                                           <!-- preregistration -->
        </switch>                                         <!-- switching PatientClass -->
      </case>                                             <!-- in-patient -->
      <case HL7_A04.PV1.PatientClass=outpatient>            <!—- outpatient -->
        <invoke partner="app Y" portType="IHETransPatientRegistrationCallbackPort"
         operation="HL7_A04" inputvariable="HL7_A04_VAR">
          <correlations>
            <correlation set="HL7_A04_CS">
          </correlations>
        </invoke>
        <receive partner="app Y" portType="IHETransPatientRegistrationPort"
         operation=" HL7_A04_ACK" inputvariable="HL7_A04_ACK_VAR">
          <correlations>
            <correlation set="HL7_A04_CS">
          </correlations>
        </receive>
      </case>                                             <!-- outpatient -->
      </switch>                                           <!-- switching PatientClass -->
    </flow>                                               <!-- patient registration flow -->
```

WSDL code

```
<!-- port type definitions (SO abbreviates second opinion) -->

<portType name="IHETransPatientRegistrationPort">
   <operation name="HL7_A01">
     <input message="HL7_A01_MSG">
   </operation>
   <operation name="HL7_A04">
     <input message="HL7_A04_MSG">
   </operation>
   <operation name="HL7_A05">
     <input message="HL7_A05_MSG">
   </operation>
   <operation name="HL7_A11">
     <input message="HL7_A11_MSG">
   </operation>
</portType>

<portType name="IHETransPatientRegistrationCallbackPort">
   <operation name="HL7_A01_ACK">
     <input message="HL7_A01_ACK_MSG">
   </operation>
   <operation name="HL7_A04_ACK">
     <input message="HL7_A04_ACK_MSG">
   </operation>
   <operation name="HL7_A05_ACK">
     <input message="HL7_A05_ACK_MSG">
   </operation>
   <operation name="HL7_A11_ACK">
     <input message="HL7_A11_ACK_MSG">
```

```
   </operation>
</portType>

<!-- service link type definitions -->

<serviceLinkType name="IHETransPatientRegistration">
   <role name="IHEActorADT">
     <portType name="IHETransPatientRegistrationPort">
   </role>
  <role name="IHEActorDepartmentSystemScheduler">
     <portType name="IHETransPatientRegistrationCallbackPort">
  </role>
</serviceLinkType>
```