Medical services workflows with BPEL4WS

Rainer Anzböck[1], Schahram Dustdar[2]

[1, 3] 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. With recent work in the field of workflows it is possible to define more flexible business models. With the standardization of the Business Process Execution Language (BPEL4WS) a new implementation method is available. The medical services domain, complex and highly structured, provides the background for our research. This paper analyzes a highly relevant medical services workflow: the *Create Diagnosis* workflow. Additionally it points out the requirements for modeling such workflows. It abstracts from the example to the structure and dynamic behavior of a workflow model useable for medical services. Furthermore, it points out a method of mapping the workflow model's transactions and activities to BPEL language patterns. Finally, it investigates a method to define medical services in a top-down approach from scenario-based UML Use-Cases to a workflow definition and BPEL specification.

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 the Business Process Execution Language (BPEL4WS, short BPEL) [2] a new implementation method for Web service based scenarios is available. Especially the medical services domain is currently in a dynamic evolution. Its workflows are complex and highly structured and a standardization of communication protocols has been covered by HL7 [3], DICOM [4] and the IHE framework [34]. Further standardization processes for health informatics are enforced by the European Union with the CEN/TC 251 work program [5].

One goal of this paper is to analyze a highly relevant workflow [6] in the medical services domain: the *Create Diagnosis* workflow. It is important for medical reasons like on-demand collaboration of specialists and increasing requirements for quality assurance procedures and for technical reasons, because it requires domain communication with other information systems, has a service-like behavior and can serve as a test case for a BPEL specification. To gain understanding in the problem domain this paper also introduces medical services together with typical scenarios and current research issues of workflow models in general.

Another goal is the definition of a workflow model for medical services. A major drawback of the current infrastructure is its missing workflow definition. The application integration is restricted to communication on a domain layer. Proprietary workflow implementations are put on top of it. Therefore, our workflow model abstracts the communication protocols by separating the application logic (domain layer) from the control logic (workflow layer). For these layers we introduce four levels of granularity (scenarios, services, transactions and activities) and show their relation to medical services. These levels are a major aspect of the workflow model's structure and dynamic view.

A third goal is showing the applicability of the WSDL [7] and BPEL standards by providing a complete specification of the *Create Diagnosis* workflow. BPEL allows a dynamic assignment of workflow participants, which is especially important for medical services scenarios like the creation of a diagnosis. As a method we map transactions and activities in the workflow and domain model to workflow and BPEL language patterns. Finally the steps shown in this paper should lead to an overall method for defining medical services from scenario-based UML Use-Cases down to a workflow definition and BPEL specification.

To summarize, our paper (i) analyzes the *Create Diagnosis* workflow in the medical services domain and points out the requirements for modeling such workflows, (ii) abstracts from the example to the structure and dynamic behavior of a model useable for medical services workflows, (iii) points out a method of mapping workflow transactions and domain activities to workflow and BPEL language patterns and (iv) investigates a method to define medical services in a top-down approach.

The paper is structured as follows. Section 2 introduces the medical services domain: scenarios, applications and protocols and outlines structures for a workflow model. Section 3 provides concepts for modeling medical services workflows and covers static and dynamic behaviors. Section 4 specifies the *Create Diagnosis* workflow in detail and applies the abstract concepts to the specific case. Finally, Section 5 introduces the BPEL language and provides a mapping between workflow model and BPEL language patterns which results in a WSDL and BPEL definition of the *Create Diagnosis* workflow. Section 6 concludes the results and suggests topics for further research.

2 State-of-the-Art

In this section, a brief introduction to the State-of-the-Art in the medical services domain is provided. The major Use-cases are introduced and special requirements regarding protocol standards and conventions of responsibility of certain software products will be considered.

2.1 Medical services Use-cases

First, we take a look at the most relevant procedures that medical services applications have to support. Figure 1 provides an overview in UML [16, 17] Use-Case notation.

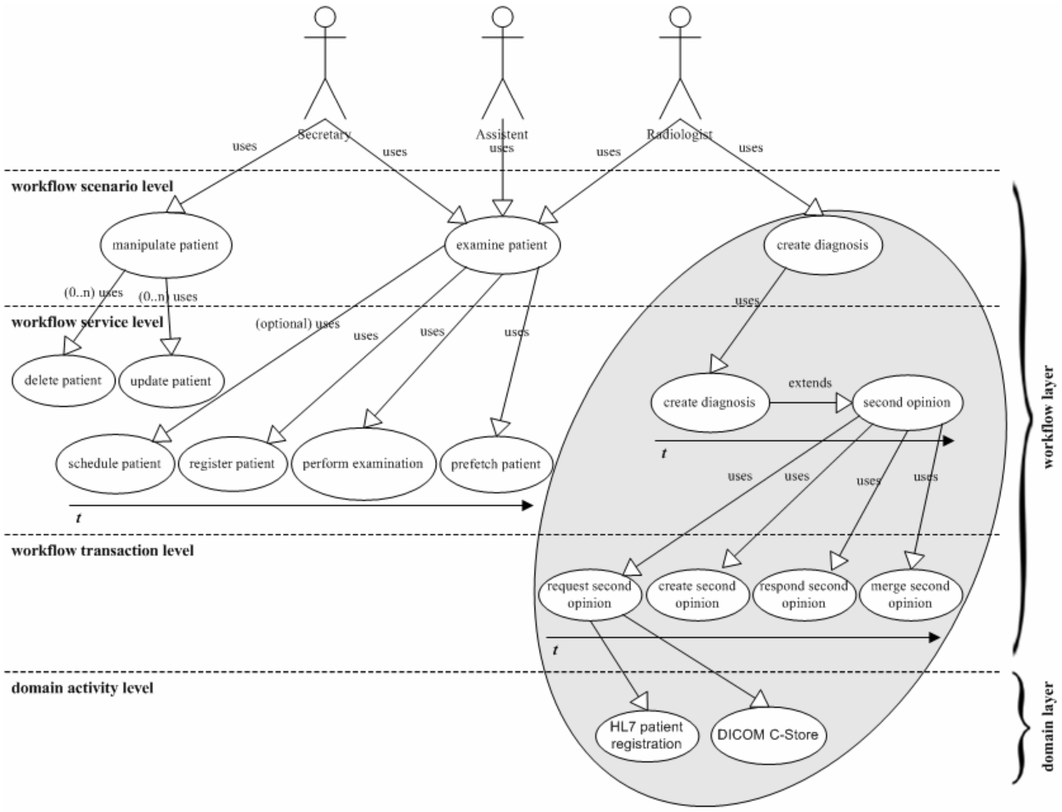


Fig. 1. Medical services use-cases follow medical and administrative scenarios. The scenarios are further divided into workflow services and transactions and domain activities. The four levels correspond to the workflow and domain layers of a medical services infrastructure.

The *Manipulate Patient*, *Examine Patient* and *Create Diagnosis* Use-cases are most common in medical services. They benefit from workflow support because of standardized working procedures in medical services facilities and the implementation of standard protocols, as described in the next section. A more detailed introduction to the domain can be found in [8-12]. We introduce a structure for Use-cases with four levels, the **workflow scenario**, **service** and **transaction** level and the **domain activity** level. Additionally the levels are part of the **workflow layer** (process) or the **domain layer** (structure and content) of a Use-case.

On the second level, services, executed by the system or the end-user, are put into sequence (time line). On this level, scenarios are divided into services that can be represented as a workflow model in the application. Those models can be aggregated for a specific scenario and reused across multiple scenarios. The granularity of a service should e.g. correspond to the definition of an IHE flow [34], a standardized workflow definition in the medical services domain. The third level of workflow transactions are atomic communication patterns, consisting of several messages exchanged unidirectional inside one or between two communication partners. Atomicity is meant semantically, where a further separation doesn't result in meaningful communication for medical services. The granularity should correspond to the definition of an IHE transaction [34]. The forth level consists of domain activities. On this layer the domain specific application data is exchanged by implementing parts of the HL7 and DICOM protocol standards.

The Use-case, we focus on, is the creation of a medical diagnosis (grey shaded in Fig. 1). This Use-case scenario consists of the *Create Diagnosis* service and an optional additional service for a *Second Opinion*. In a second opinion procedure all information about a medical case is transferred between two locations. A specialist residing in a remote location evaluates, corrects or extends the diagnosis with his expertise and increases the quality of the diagnosis result. This extension might occur several times and is not restricted to a particular number of repetitions. The Use-case is important for three reasons. First, it is for medical reasons, because skills between specialists are complementary, meanings sometimes strongly differ from each other and quality assurance procedures in the medical services domain are gaining increasing attention. Second, it consists of domain communication (exchange of domain data between different sites and applications) and requires applications to provide the service using a workflow model and a BPEL interface. Additionally, most medical equipment lack workflow functionality which reduces the number of relevant Use-cases. Third, it is possible to analyze the "extends" semantic of a Use-case and its implication on the workflow model. This is also related to the recent discussion of adhoc workflows [35].

The structuring into levels and layers has direct implications on the design of applications. It supports the definition of a workflow infrastructure consisting of scenarios, services, transactions and activities and their workflow model as shown in Section 3 and 4 in this paper. A BPEL interface for the workflow and domain activities will be defined in Section 5. Furthermore, workflow model constructs are introduced at this early stage which corresponds to a scenario-based design approach [33]. Finally, it should be possible to derive the design of other scenarios from the approach shown here. However, additional patterns for branching and merging workflow services are not investigated. It would be necessary to extend the Use-case's UML semantics, which is out of the focus of this paper.

2.2 Medical information systems

Three types of medical information systems, the HIS (Hospital Information System) [13], the RIS (Radiology Information System) [12] and the PACS (Picture

Achieving and Communication system) [10] are the backbone of current information systems in the hospital and medical services environment. They are comparable to ERP (Enterprise Resource Planning) or SCM (Supply Chain Management) in business organizations as these types of systems are found in most medical services environments. The HIS is an enterprise-wide system used for administrative services like patient and visit management, bed reservations, patient referrals, operation planning and other scheduling services and billing management. The RIS is a patient-management system required for all organizational services of a medical services facility (whether in- or outside a hospital) like examination scheduling, patient registration, worklist generation, examination control, report generation (using digital dictation equipment and speech recognition) report transcription and transfer and billing management for insurance companies. 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 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.

The Use-cases introduced above are sometimes implemented for standalone medical services facilities and sometimes integrated as departmental services for a larger hospital environment. In the second case there are also workflows that span multiple departments. Because of their special storage, network and process performance requirements RIS and PACS systems are always a major part of the infrastructure, even in hospitals. Furthermore, the functionality of these systems is overlapping and a specific role in a scenario might be played by a RIS in one case and by a PACS in another case. For a workflow model to be successful, the existing application infrastructure has to be considered. The medical services have to be attached to a common workflow layer independent of the system implementing them. Therefore, this paper concentrates on workflows integrating the medical services provided by HIS, RIS and PACS.

2.3 Medical services standards

The most relevant protocol standards for these applications 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, films, printers, and examination modalities. 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 and 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 [3] and on DICOM in [4, 14, 15]. Furthermore, the IHE technical framework [34] 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 applications acting in a specific role. To claim IHE conformity for a role, a minimal set of flows and transactions has to be implemented. Applications currently agree on a common definition of domain activities based on the DICOM and HL7 standards. IHE conformance of applications, being currently in its beginnings, leads to an agreement on how domain activities should be aggregated to workflow transactions and medical services. Nevertheless, a workflow model that integrates these services and handles workflow instances is still missing.

2.4 Medical services communication

When building medical services applications, we have to consider the application structure and protocol support of current implementations. Some of the Use-cases in Figure 1 do not have to deal with other nodes in the network and therefore completely lack the need of having a DICOM or HL7 implementation. Those scenarios further require no BPEL implementation, because BPEL is only used for the workflow execution between different service partners. This paper therefore focuses on scenarios and services requiring domain communication. An example of how medical services are currently being performed is shown in Figure 2. Domain activities are executed and domain communication is performed which corresponds to the domain activity level of Figure 1.

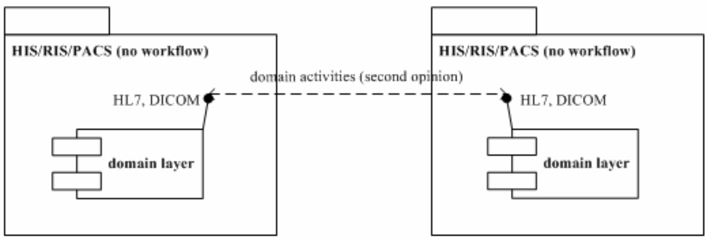


Fig. 2. The communication pattern that is used between current medical services applications is implemented in a domain layer. Applications are integrated by negotiating the protocol support on a per-project base depending on the implemented scenarios.

As we can see, the implementation of the standard protocols resides in a part of the software that we call domain layer. This layer covers the application logic, the domain specific actions that have to be performed, like exchange of patient data or transmission of images. All applications rely on it when performing their DICOM and HL7 communication through the respective interfaces. Without IHE support the applications might be called "RIS with HL7 support for patient management and a

DICOM interface for storing images". In IHE terms they play the role of an *Image Manager* and an *Evidence Creator* and implement a *second opinion* flow (workflow service) by performing *patient registration* and *modality image stored* transactions using several HL7 and DICOM messages (domain activity). While IHE is suited for defining medical services it doesn't provide a workflow model and defines flows as sequence diagrams. Integrators implement the behavior as a static process model. They provide conformance statements which represent implemented services and activities to integrate applications for a specific scenario.

Next, we provide a dynamic view on how medical services are currently executed. An example implementation of the *Second Opinion* service in the C*reate Diagnosis* scenario is shown in Figure 3.

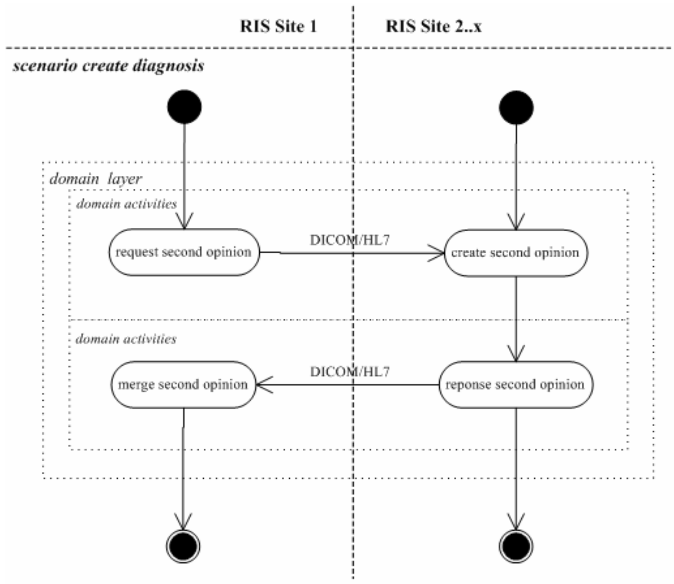


Fig. 3. Example of the *Second Opinion* service of the *Create Diagnosis* scenario as it would be modeled in current applications. Domain activities are defined to model the DICOM- and HL7-based domain communication.

The *Second Opinion* service is split up in two steps of domain activities. DICOM and HL7 communication, as defined in the domain layer, takes place in both directions. First, the primary site requests a second opinion and transfers data to a secondary site. At the second site, a second opinion is performed by creating an additional report or entering annotations to images. Afterwards the results are sent back and merged into a final diagnosis.

Primarily, this infrastructure is missing a workflow layer. The application integration is restricted to the domain layer level. No workflow information e.g. service availability, execution progress or workflow instance history is available. Even more, the aspect that a *second opinion* service is optional can not be expressed in any way. Therefore, we define a new workflow model for medical services and focus on the requirement of a workflow layer to overcome current deficiencies. A more detailed description of medical services applications can be found in [11, 12, 13].

2.5 Dynamic Workflows

In this section we provide a short introduction to related work in the domain of workflow systems. Figure 4 shows the *loosely coupled workflow model*, a model where each participant implements his own workflow. The model has been introduced and is described in more detail in [18, 19].

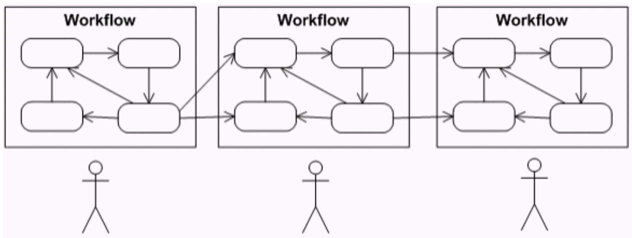


Fig. 4. In the *loosely coupled workflow model* each participant implements his own workflow and maintains an infrastructure, like a workflow engine, etc. At defined synchronization points work items can be exchanged to constitute an inter-organizational workflow.

Additionally the workflows are connected at certain points of interaction (synchronization points). The participants can now be selected dynamically leading to a more flexible way of interaction. To allow for a dynamic assignment, the process definition has to be stored as a template as introduced in [18] and evaluated in [20]. The template defines the process states, activities and roles of the workflow. When the process is instantiated a specific participant is chosen for a role in the workflow. For example the radiologist generating the report for a specific case is dynamically selected based on availability, expertise on the case and other modeled attributes of the activity. All attributes, connection parameters, etc. should be stored as metadata in the messages of a service specification. Another advantage of *loosely coupled workflow models* over traditional workflow models is the extension of models through escalations [20]. An escalation defines additional activities and is inserted into the process model at certain escalation points. It can be triggered by a participant who is defined as an expert with respect to the process definition (e.g. a radiologist for requesting a second opinion on a case). Escalations extend the running process instance and participants filling the roles that perform the activities are dynamically assigned. In [20] it is concluded that currently workflow systems don't support dynamic assignments and a new framework has been introduced, that extends the WFMC reference model. We assume that such extensions are required to support dynamic behavior. An event-based approach to extend the reference model with dynamic aspects is suggested in [21]. When defining a new workflow model for medical services, flexibility and dynamic behavior have to be considered. The *Create Diagnosis* scenario contains services that provide all aspects of a dynamic workflow and a model based on WSDL and BPEL allows the dynamic use of services. The next section introduces a model where medical services benefit from dynamic workflows' concepts.

Another approach, the Serviceflow project [36, 37], focuses on contract-based workflows. The workflow infrastructure combines *loosely coupled* and *case transfer* behavior, another model introduced in [18, 19]. It consists of service points where workflow participants exchange cases in form of XML documents called service floats, which contain workflow and domain data. Participants sign a contract by agreeing on rules and conditions defined in service point scripts. A workflow is executed by transferring service floats from one service point to the next and evaluating conditions by executing service point scripts. Dynamic changes to the route of service floats are possible at each service point. The infrastructure consists of components for service point management, service float handling and user-interaction. Components can be set-up in different combinations across the network to allow for choosing between centralized and distributed workflow architectures depending on the domain and the relation between workflow participants.

In comparison to our approach, the workflow model focuses on cases, combining workflow and domain data. As discussed in section 2.4, medical services, especially IHE transactions, are clearly defined by their protocol communication. Applications communicate on their own, which requires the flow of domain and workflow data to be separated. Nevertheless, contracts, service points and service floats, restricted to workflow information, are promising concepts for designing a workflow infrastructure. While we currently focus on workflow models and methods of service definitions, we will investigate requirements for an infrastructure in our future work.

3 Medical services workflows

In this section we provide a model for medical services workflows. As stated in the last two sections, the medical services domain requires a workflow layer as a common base of all domain activities. First Figure 5 provides an abstract definition of medical services Use-cases. The levels of granularity for scenarios, services, transactions and activities are extracted from Figure 1.

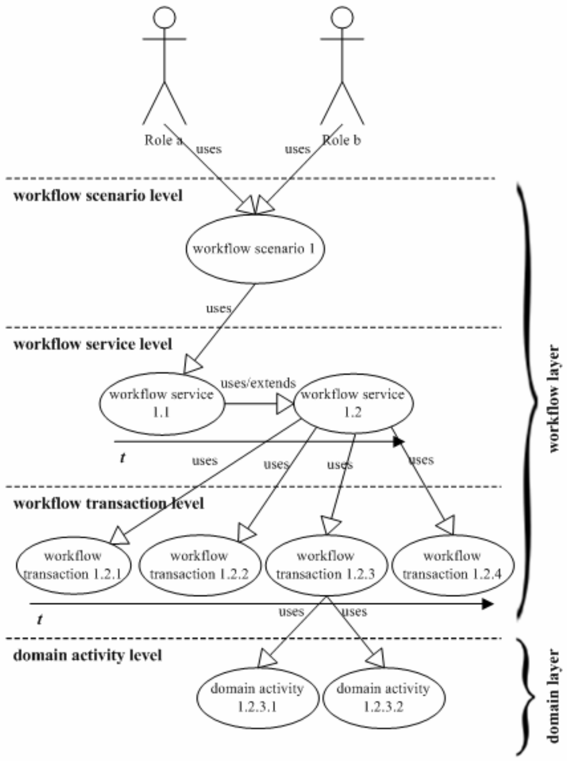


Fig. 5. An abstract definition of medical services Use-cases is introduced. The four granularity levels and the separation into workflow and domain levels and layers are part of the workflow model.

We introduce a workflow layer as a distinct part of a medical services infrastructure, to separate process and content. Furthermore, we outline that the scenario, service and transaction levels are part of a workflow layer for medical services (process) and the activity level is part of the domain layer (structure and content). Use-cases as introduced in 2.1 are first modeled on the scenario level and medical services (e.g. IHE flows) to implement them are defined on the service level. The services consist of several transactions that are performed in sequence. Transactions can also be optional and branches and other conditions can be introduced on the third level. As we point out in section 4, services and transactions correspond to BPEL processes and ports. Finally, the domain communication is again implemented as activities (e.g. HL7 and DICOM messages) on the forth level.

Compared to Figure 2 we define the extended workflow environment as shown in Figure 6.

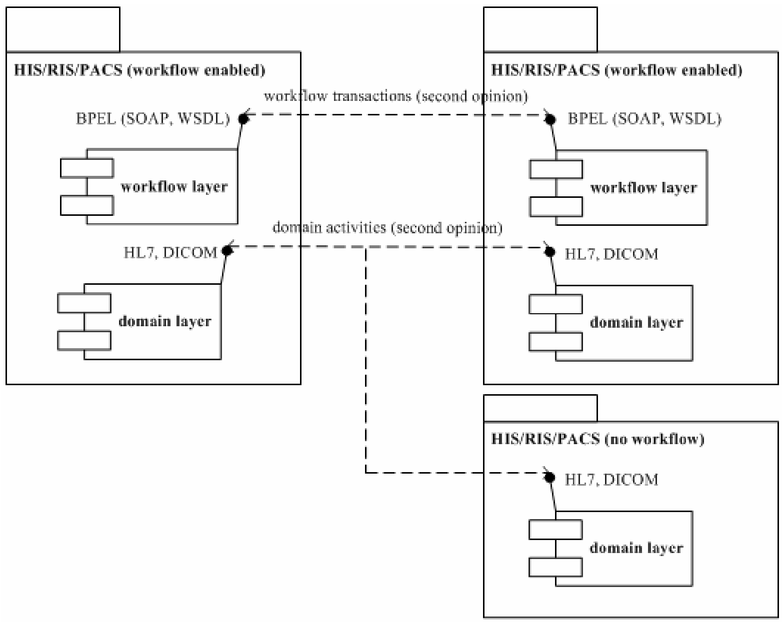


Fig. 6. For a medical services workflow environment we introduce an additional workflow layer as a distinct part of the application, to separate process and content.

The environment of a medical services infrastructure is, of course, not always workflow enabled. Most medical equipment lack workflow functionality but has to be considered part of the environment. Therefore, we have to deal with communication that is just based on the domain layer and their protocol implementation, while we extend this layer with a workflow layer wherever possible. The additional exchange of workflow information is based on BPEL processes, BPEL ports and a shared WSDL description of medical services. This model can be used in general for all workflow scenarios that were previously introduced.

Next, we provide an abstract definition for the execution of workflow services, transactions and domain activities, resulting in a generic execution environment for medical services' workflows. Starting with our example of Figure 3, the *Create Diagnosis* scenario, Figure 7 contains an abstract representation of the domain activities as they are executed in the environment with no workflow layer present.

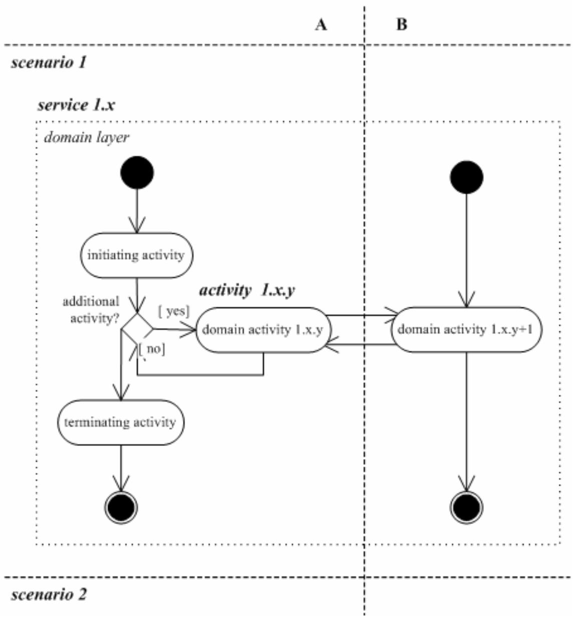


Fig. 7. The figure shows how domain activities are performed in an environment without a workflow layer.

This model refers to an instance of a domain communication, anticipating the path of execution with respect to AND and OR branches of a predefined static workflow. Each domain activity is preceded by an optional initialization activity and followed by a termination activity. Those are used to perform infrastructure operations, like security checks, establishment of network connections, etc. Figure 7 represents the dynamic view of domain activities as they are performed for an instance of a model with a single domain layer. For example A would perform the "request second opinion" and the "merge second opinion" transactions. For B the "create second opinion" and "response second opinion" are executed. All transactions are implemented as DICOM and HL7 protocol domain activities. Therefore, to implement the "create second opinion" transaction the corresponding protocol messages containing the required domain data have to exchanged. Because of the missing workflow layer, workflow information is built directly into the application. This model stays the same in a workflow-enabled environment for non-workflow enabled nodes. We don't expect that medical services can be implemented with this current model. But it is helpful to understand the application interactions, especially in comparison to our extended workflow model.

From this point we extend the model of activity execution and introduce the explicit workflow layer (see Figure 8). We have to distinguish both layers and again use the four levels of granularity for scenarios, services, transactions and activities of Figure 5.

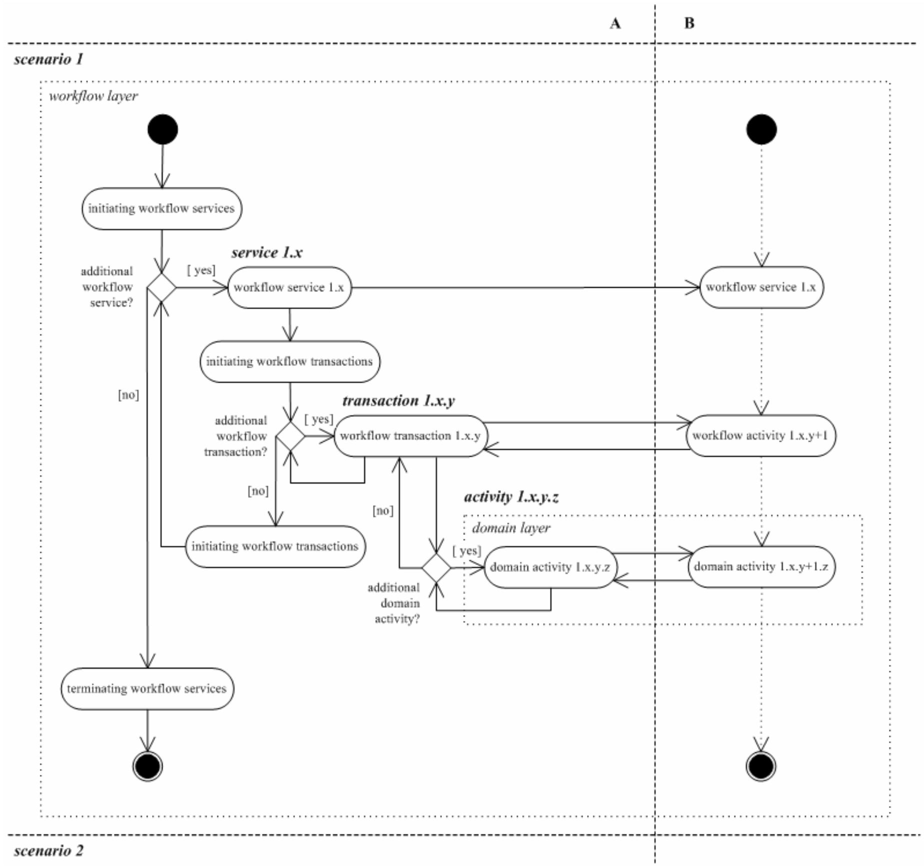


Fig. 8. With the introduction of a workflow layer the workflow and domain communication get separated and workflow information is processed explicitly.

On the workflow layer, scenarios are separated into specific services and transactions. For simplicity, we concentrate on services that are executed in sequence, branches or choices are not considered. While workflow services and transactions are meant to purely process workflow information, domain activities just contain the implementation of the communication protocols. Compared to Figure 7 application logic and control logic are separated. While in the first model the domain communication occurs without an explicitly defined workflow and between predefined partners, the workflow layer allows selecting participants dynamically based on availability and other workflow attributes. This is also consistent with Figure 6, where domain activities in the domain layer trigger the DICOM and HL7 ports, while the workflow transactions on the workflow layer perform domain activities using BPEL.

On the domain layer everything remains the same, while the transactions of the new workflow layer enclose the domain activities. If the remote site with which the workflow is executed is workflow-enabled, both layers are activated. Otherwise, the layer is omitted and the domain ports are activated directly. This approach doesn't change the way domain activities are performed, it especially enables other applications to process workflow information. Furthermore, it is possible to support a mixed environment (with and without workflow layer) with only one model. Additionally, this model is more flexible and allows independent changes to the process and the structure of medical services.

After this abstract view on medical services workflows, we take a closer look on the *Create Diagnosis* workflow to test our model and provide a definition for the required workflow communication based on BPEL. Furthermore, we take a closer look on the domain activity level, where the structure of communication becomes more complex and we have to focus on branching and merging of activities. The separation into a distinct workflow layer allows us to model workflow pattern directly using BPEL constructs.

4 Create Diagnosis workflow

In this section we investigate the *Create Diagnosis* scenario in detail and show how the introduced workflow model can be applied to the specific medical service of a *Second Opinion*. Primarily, this scenario is chosen because we require a scenario with workflow participants and domain communication between them to focus on modeling using BPEL. Furthermore, it is highly relevant as it addresses quality management in medical services. Additionally, the support for the DICOM and HL7 standards is well defined and can be interrelated by mapping protocol messages to the workflow model. Beside the possibility to interact between two sites, the *Create Diagnosis* workflow can also be implemented as an internal service on a single site. In relation to dynamic workflows (see 2.5), the workflow can be extended to form a *loosely coupled workflow system*. Related to Figure 8 the domain activities that are performed on the remote site and represent the second opinion are just used in the extended case. In this paper we describe the *Create Diagnosis* workflow including a *Second Opinion*. For an implementation of workflows based on BPEL this extension is the relevant part.

With respect to a BPEL implementation there is currently no generally used graphical notation of workflows. Several different semantics are found in the literature. All of them have some limitations and use syntactical extensions to define workflow behavior. Extensions e.g. branch conditions are also used here. Requirements for and a detailed analysis of pros and cons of modeling techniques can be found in [6] and [24]. We use UML activity diagrams as they correspond to our workflow model definition and, as can be seen, are suited for specifying Web services.

4.1 Create Diagnosis services and activities

The normal workflow and the second opinion extension are shown in Figure 9. The standalone part of the workflow is executed within a medical services facility and corresponds to a local RIS functionality. As an extension it is possible to split the *Create Diagnosis* service and perform a preliminary step that is later merged with results from the *Second Opinion* workflow.

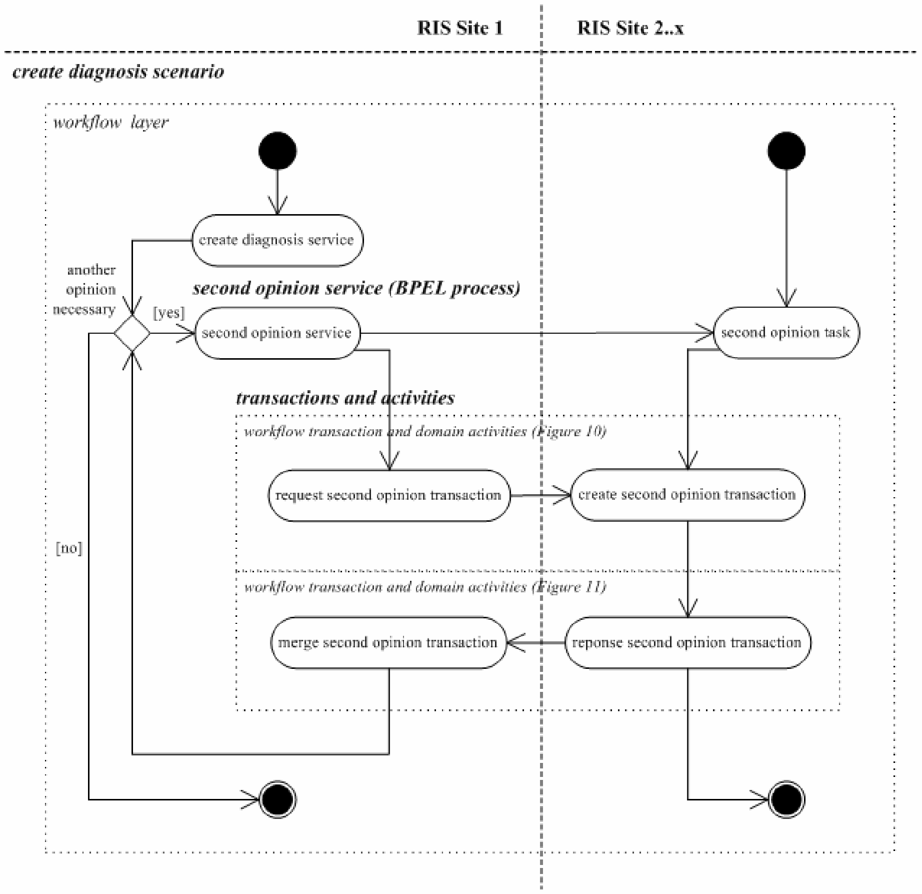


Fig. 9. The *Create Diagnosis* workflow consists of a local workflow and an optional extension to other sites where preliminary diagnosis are made. In both cases the workflow continues seamlessly.

The second opinion medical service workflow is executed between two medical services facilities. One is the initiator (RIS Site 1) or user all further (RIS Site 2..x) are provider of the second opinion. The service is related to a BPEL process and consists of four workflow transactions. Of special interest is the mapping between the *Second Opinion* service and its transactions in Figure 9 and the domain activities shown in Figures 10 and 11. This mapping defines the separation of the workflow layer and the domain layer as illustrated in Figure 6. Further, a second opinion case requires a workflow implementation for each RIS site. In real-world scenarios one has to deal with legacy applications and equipment that are not workflow-aware, but support DICOM and HL7 protocols (see Figure 6). Therefore the approach shown here has to be extended with proxy mechanisms to simulate workflow behavior. We recognize this topic as important research area, which is out of the scope of this paper.

Next, we describe the structure of the service, transaction and activity levels with patterns as suggested in [22] and [23]. The patterns also have to be separated into these levels of granularity. First, we start with the service patterns as they are not related to the BPEL implementation, the transaction and activity patterns are covered in the following sections. In Figure 9 after the *create diagnosis* service there is a workflow pattern called *Exclusive Choice* (one of several branches is chosen). This pattern is referenced as WP4 (Workflow Pattern) in [22]. The boolean-valued branch condition is based on a workflow attribute *another opinion necessary*. The user of the RIS application decides to instantiate a second opinion workflow case by case. Such service attributes have to be defined on the scenario level in a scenario-based design approach. After execution of the second opinion the workflow process continues as in a scenario without a second opinion.

4.2 Second opinion request activities

We now proceed to the next level of granularity, the domain activities. Figure 10 and 11 show the details of the activities (HL7 and DICOM message exchange) in this workflow. Two phases of interaction (BPEL service links) with two transactions on both sites occur. Each transaction consists of two BPEL ports for the DICOM and HL7 protocol. In the first phase the patient and image data is provided to RIS Site 2, while in the second phase the diagnosis results are sent back to RIS Site 1. The DICOM and HL7 messages used are defined in [3], [4]. These domain activities are mapped to BPEL messages on the workflow layer. The selection of the messages is based on real-world scenarios implemented in products like Tiani [26] and D.A.T.A. [27]. What makes this model different from just specifying a message exchange is that these messages are already grouped together meaningfully and can be reused in other BPEL processes. In the first phase (Figure 10) RIS Site 1 communicates patient information using HL7 patient registration messages. The images are transmitted using the DICOM C-Store service class implementing a Client/Server model.

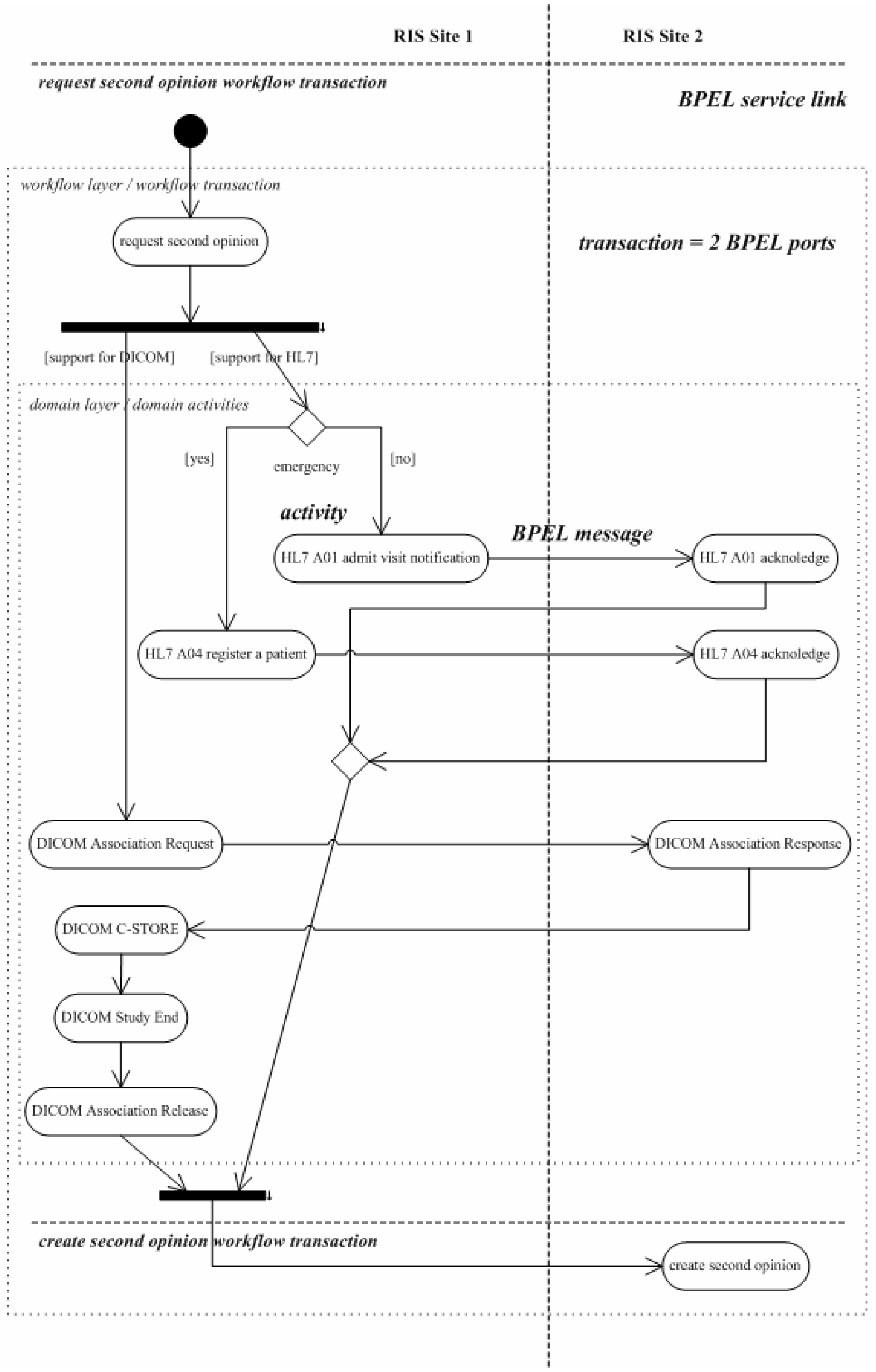

RIS Site 1
RIS Site 2
request second opinion workflow transaction
BPEL service link
workflow layer / workflow transaction
request second opinion
transaction = 2 BPEL ports
[support for DICOM]
[support for HL7]
domain layer / domain activities
[yes]
emergency
[no]
activity
HL7 A01 admit visit notification
BPEL message
HL7 A01 acknoledge
HL7 A04 register a patient
HL7 A04 acknoledge
DICOM Association Request
DICOM Association Response
DICOM C-STORE
DICOM Study End
DICOM Association Release
create second opinion workflow transaction
create second opinion

Fig. 10. The first detailed view of Figure 9 shows the message exchange between the request second opinion and create second opinion workflow activities. The HL7 and DICOM protocols are used to transfer patient and image data between both sites. Depending on the protocol support different interaction scenarios are implemented.

In Figure 10 the patterns on the transaction and activity level are now directly supported by BPEL constructs. On the transaction level there is a simple control flow of pattern type *Sequence* (WP1). On the activity level, the first point after the activities *request second opinion* is a *Parallel Split* (WP2 - single thread splits into threads executed in parallel). The HL7 and DICOM activities are allowed to be executed simultaneously and in any order. The branch conditions use boolean-valued expressions for the protocol support of the implementation. The workflow model also works without one of these protocols. Additionally, it is possible to define a proprietary protocol providing the same semantic behavior pointed out for the high-level view. At point *emergency* there is an *Exclusive Choice* (WP4) using the corresponding expression. Depending on the case more or less detailed patient information is provided for the remote site. In case of emergency only a short pre-registration services place. Between the *HL7 A01 acknowledge* and *HL7 A04 acknowledge* activities there is a *Simple Merge* (WP5) unifying these cases. Afterwards, there is a *Synchronization point* (WP3 - multiple parallel branches converge synchronizing multiple threads). The HL7 and DICOM message exchange is synchronized before proceeding creating a second opinion. All other control flows are of the simple pattern type *Sequence* (WP1).

4.3 Second opinion response activities

Figure 11 contains details of the second part of the workflow. The second opinion created by RIS Site 2 is communicated using HL7 observation messages. The DICOM protocol is not required in this scenario. The scenario could be extended to consider image annotations created during the second opinion process on the remote site. The mapping of transactions and activities to BPEL constructs remains the same as in Figure 10.

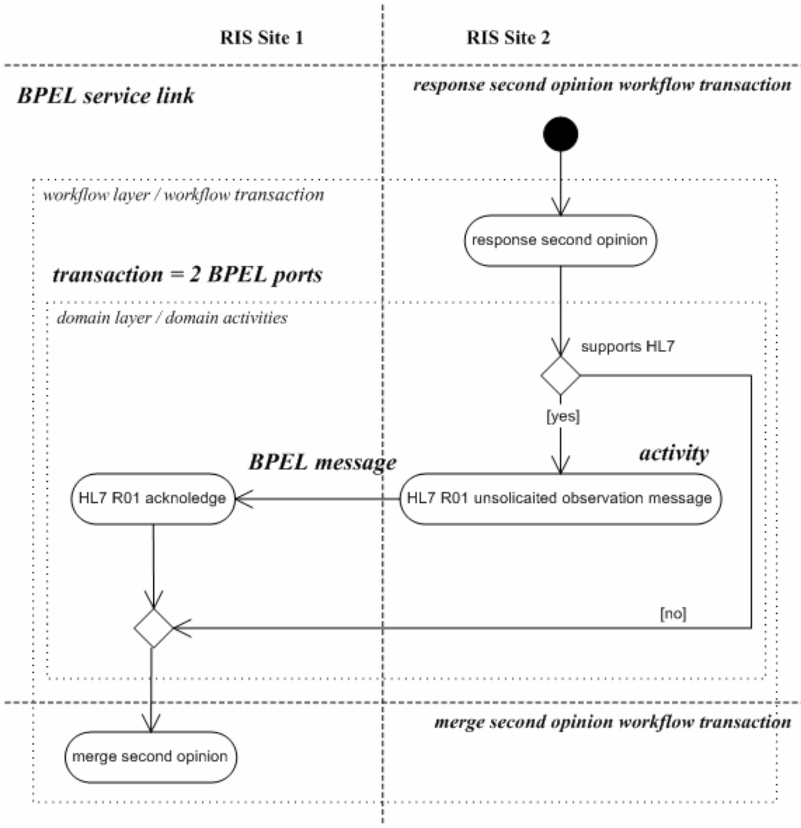


Fig. 11. The second detailed view of Figure 9 shows the message exchange between the *response second opinion* and *merge second opinion* activities of the workflow. Because results for a diagnosis are provided in text format only, the HL7 protocol is most appropriate for this situation.

Again, on the transaction level there is a simple control flow of pattern type *Sequence* (WP1). On the activity level, at the first point between activities *response second opinion* and *HL7 R01 unsolicited observation message* there is an *Exclusive Choice* (WP4) pattern. Depending on the protocol support the HL7 protocol is used. Again proprietary protocols can be implemented in accordance to the workflow model. At another point, shown before the *merge second opinion* transaction, there is a *Simple Merge* (WP5) that synchronizes the cases of protocol support. All other control flows are *Sequence* pattern (WP1). For example the *HL7 A01 admit visit notification* and the *HL7 A01 acknowledge* build such a pattern.

As a conclusion the second opinion workflow can be described using UML activity diagrams. A two level hierarchy with one overview and two detail flows has been defined. In the next section a BPEL specification is provided for the second opinion workflow.

5 BPEL definition

In this section we provide a BPEL definition of the second opinion workflow service, its transactions and activities. An introduction to BPEL and its terminology is given. In a second step, we provide example mappings between activity diagrams and BPEL code. Third, we provide a part of the BPEL and WSDL specification. Because WSDL defines the structure and BPEL the flow of data exchange both definitions are required to gain an understanding in the workflow model's formal description. Finally, some limitations of the definition are pointed out.

5.1 Introduction and Terminology

BPEL has been designed for modeling workflows. It additionally focuses on current Web service based environments and is based on XML notation and WSDL service descriptions as well understood concepts. In workflow terminology BPEL defines activities and their relationship in a flow style of XPath [28] or a directed graph style of WSFL [29]. Several language constructs for branching and merging are provided that support the pattern used in the second opinion workflow definition. BPEL defines the partners interacting in a workflow, the services and the roles each partner has. Business processes specified via BPEL prescribe the exchange of messages between Web services. These messages are WSDL messages of operations of the port types involved in the roles of the service links established between the process and its partners [17]. For a better orientation we shortly introduce major terms used in the WSDL and BPEL specification.

5.1.1 Partners, Roles, Port Types and Service Link Types

WSDL and BPEL define business partners that provide and consume services and can take several roles in a workflow model. Services partition the workflow model of partners and their activities normally into two or more partners and one or more activities. The port types (service interfaces) are defined by operations and the exchange of messages. Normally services use producer/consumer or client/server semantics, leading for example to a producer and consumer port definition and a client and server role. A service link type relates such services to the workflow model. It defines which business partner performs which role and the port types it communicates through. During runtime specific ports, corresponding to a port type, are used to define additional connection parameters, security properties, etc. What's sometimes misleading is that WSDL provides interface definitions through port types and service link types. The services themselves are not explicitly defined.

5.1.2 Operations, Messages, Context

Each port type defined in a WSDL file consists of one or several operations performed between the partners. The flow of operations is defined in the BPEL file. An operation consists of one or more input and output messages. Each message consists of workflow data and context data for security, transactions, etc.

5.1.3 Variables, Expressions, Conditions, Correlations

Variables are used to store message data of stateful workflow interactions. X-Path [30] like expressions (e.g. boolean, deadline-based, general) can be defined using variable definitions. These expressions can be used for branch conditions and other dependent operations. Message parts can be correlated by defining relations between variables in two or more messages. The variable content can also be copied for message ID-like semantics.

5.1.4 Simple and complex activities

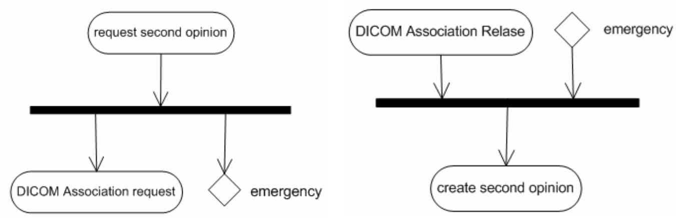
Activities are the building blocks of the BPEL workflow definition. Simple activities like *receive*, *reply*, *invoke*, *wait* define communication pattern for initiating or waiting for operations, time triggered operations, etc.; further more are defined. Complex activities like flow (for ordering activities), link (for graph-like linking of activities), scope (for grouping activities), sequence, switch and while (for flow control) provide workflow pattern for relating simple activities.

5.2 Implementation

With BPEL we have a basis for a detailed workflow specification. However BPEL doesn’t provide a graphical representation and common representations for workflows have their pros and cons [6], [24]. We have chosen activity diagrams because it is well understood and supports BPEL language constructs. Therefore, we are able to derive BPEL language pattern directly from the specified diagrams. The approach chosen here suggests that the mapping of workflow patterns can be automated. Such automation would enhance integrity and consistency of the workflow model and its implementation specification similar to techniques used for class diagrams in software development. An in depth analysis of BPEL patterns can be found in [25].

5.2.1 Dispatching HL7 and DICOM support

The first patterns we examine are the *AND Split* and the *AND Join*. As shown in Figure 10 such constructs are used to execute the DICOM and HL7 implementation in parallel. The BPEL language provides the *flow* statement to define this relationship between activities. Figure 13 compares the workflow patterns with the BPEL construct.



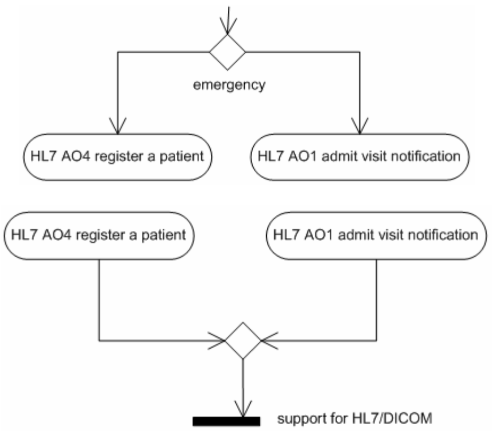

```
<!— AND Split -->
<flow>
<!-- HL7 flow -->
</flow>
<flow>
<!-- DICOM flow -->
</flow>
```

Fig. 13. An example comparison of *AND Split* and *AND Join* workflow patterns and the BPEL *flow* construct is shown. It focuses on the split for HL7 and DICOM protocol support. The corresponding BPEL file consists of two parallel flows for each implementation.

On the left side the workflow patterns enable parallel execution of services. This functionality is reached using the *flow* construct in BPEL. In our case the HL7 and DICOM support are executed in parallel. The constructs are used like in programming languages and can be nested, have conditions, etc.

5.2.2 Switching between HL7 patient registration modes

The next patterns compared are the *XOR Split* and *OR Join*. As shown in Figure 10 such constructs are used to specify DICOM and HL7 protocol support. The BPEL language provides the *switch* statement to define this relationship between activities. Figure 14 compares the workflow patterns with the BPEL construct.




```
<switch>
<!-- switching HL7 support -->
<case hl7support=false>
<!-- HL7 flow empty -->
</case>
<case hl7support=true>
<!-- HL7 flow continues -->
</case>
</switch>
<!-- HL7 support -->
```

Fig. 14. An example comparison of *XOR split* and *OR join* workflow patterns and the BPEL *switch* construct is shown. It describes the alternative threads for a registration of a patient in case or without an emergency. The BPEL file implements this behavior with the switch construct.

On the left side the workflow patterns enable exclusively one execution of two or more services. This functionality is reached using the *switch* construct in BPEL. In our case depending whether HL7 is supported by the application or not the corresponding workflow functionality is executed.

5.2.3 Example Code

The following Figure 15 provides an example of the BPEL implementation to get a better insight how the language concepts fit together.

```
<!-- partner definitions -->
<partners>
  <partner name="secondarySite"
    serviceLinkType="SOResponseSecondarySiteLT"
  myRole="SOResponseSecondarySiteUser"/>
</partners>

<!-- variable definitions -->
<variables>
  <variable name="HL7_R01_UOM" messageType="hl7:HL7 R01 unsolicited observation message">
  <variable name="HL7_R01_ACK" messageType="hl7:HL7 R01 acknoledge">
</variables>

<!-- workflow definition -->
<flow>                                  <!-- HL7 flow -->
  <switch>                              <!-- switching HL7 support -->
  <case WF_SO_REQ.hl7support=false> <!-- HL7 flow empty -->
  </case>                                <!-- HL7 flow empty -->
  <case WF_SO_REQ.hl7support =true> <!-- HL7 flow continues -->
    <receive partner="secondarySite" portType="SOPrimaryHL7CallbackPort"
     operation="HL7 R01 unsolicited observation message" inputvariable="HL7_R01_UOM">
      <correlations>
        <correlation set="HL7_R01">
      </correlations>
    </receive>

    <invoke partner="secondarySite" portType="SOSecondaryHL7Port"
     operation="HL7 R01 acknoledge" inputvariable="HL7_R01_ACK">
      <correlations>
        <correlation set="HL7_R01">
      </correlations>
    </invoke>
  </case>                         <!-- HL7 flow continues -->
  </switch>                       <!-- switching HL7 support -->
</flow>                           <!-- HL7 flow -->
</process>
```

Fig. 15. A BPEL example implementation of the Response Second opinion transaction corresponding to Figure 12.

The implementation example contains the interaction during the *response second opinion* transaction. As a simplification the scenario is reduced to two business partners. The process flow contains the switch construct separating the HL7 implementation. The HL7 messages are exchanged with invoke and receive BPEL constructs. A complete listing of the BPEL implementation of the Create Diagnosis workflow can be found in [31].

5.2.4 Limitations of the definitions

Finally, we summarize which features haven't been considered in the BPEL definition. There is no scope definition, because we think more experience is required to argue a specific design decision. One possibility is the specification of a scope per protocol, because the HL7 and DICOM flows are independent and produce different errors through external applications. A further separation can be made between the request and response of the second opinion, depending on whether this is seen as a single transaction; in real-world this is normally not the case. The definition of namespaces is performed exemplary but not used throughout the example. Further missing are *links*, because the mapping just uses XLang [28] specific constructs. Fault and compensation handler haven't been used. All those language extensions should be investigated in further workflow scenarios, especially scopes are identified as an important research area. Methods to automate the creation of BPEL specifications would further be beneficial. Additionally an implementation and simulation of a workflow model is necessary to conclude the suitability of BPEL for the medical services domain.

6 Conclusions and Future work

The contribution of this paper was to analyze medical services and to introduce a workflow model to support designing services from Use-cases down to BPEL specifications. We have chosen a highly relevant workflow in the medical services domain: the *Create Diagnosis* workflow. This workflow is a well-suited candidate for modeling an enactment using Web services technologies. The suggested workflow model covers implementation relevant aspects of the DICOM and HL7 protocols as part of a domain model while maintaining a separated workflow model. Essentially, the separation into a domain and a workflow layer (Fig. 8 – Fig. 10) and the structure of transactions and activities seem to fit perfectly well for modeling with BPEL. A correct mapping of the workflow specification to the service link type and port definitions of BPEL is possible.

Next, we will analyze workflow models for medical services based on Web services in more general. The next step should be the definition of a complete method for specifying medical services. Furthermore, an automation of workflow definitions based on UML specifications seems feasible. Here different workflow patterns on the service, transaction and activity level are of further interest. Also important is the definition of an infrastructure for our workflow model based on the WFMC model and the suggested workflow and domain layer. The infrastructure has to consider workflow attributes for routing, service availability and further aspects of workflow execution. As stated in the paper, proxy mechanisms for applications that are not workflow-aware have to be integrated.

Nevertheless, the implementation and simulation of a workflow model is necessary to conclude the suitability of BPEL for the medical services domain. For now, we conclude that BPEL is suitable, but several additional research topics have to be covered. Another step should be the evaluation of the workflow using a BPEL server like Collaxa [32]. A future work should be to implement a workflow-based application supporting a BPEL interface for medical organizations.

Acknowledgements

This research was supported in part by a research award for *Development Methods for Dynamic Web services Workflows* of the Chamber of Commerce Vienna (Wirtschaftskammer Wien).

7 References

1. Workflow Management Coalition: WMFC Reference Model, www.wfmc.org (1995)
2. 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)
3. HL7 Organization: Health Level 7, http://www.hl7.org (2000)
4. NEMA and Global Engineering Group: DICOM 3 Standard, http://www.nema.org (1998)
5. CEN/TC251, Health informatics - Medical data interchange: HIS/RIS-PACS and HIS/RIS - Modality Interface - ENV 13939, http://www.centc251.org/ (2001)
6. van der Aalst, W.M.P., Weske, M., Wirtz G.: Advanced topics in workflow management: issues, requirements and solutions
7. W3C: Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl.html (2001)
8. Siegel, Eliot, Kolodner, Robert M.: Filmless Radiology, Springer (1998)
9. Huang, H. K.: PACS: Basic Principles and Applications, Wiley-Liss (1998)
10. Anzböck, R.: XR PACS Architektur, D.A.T.A. Corporation, http://www.data.at (2001)
11. 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)
12. Anzböck, R.: XR OPEN RIS Architektur, D.A.T.A. Corporation, http://www.data.at (2001)
13. 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)
14. Revet, Bas: DICOM Cookbook, Philips Medical Systems (1997)
15. Oosterwijk, H.: DICOM Basics, OTech Inc/Cap Gemini Ernst and Young
16. Object Management Group (OMG): UML 2.0 Standard specification, http://www.omg.org (2003)
17. Fowler, Martin, Scorr, Kendall: UML destilled, Addison Wesley Professional (2000)
18. Von der Aalst, W.M.P., Weske, M.: Process-oriented Architectures for Electronic Commerce and interorganizational workflow, Eindhoven University of Technology (2001)
19. Von der Aalst, W.M.P., van Hee, K.: Workflow Management, Models, Methods and Systems, Academic Service (1997)
20. Georgakopoulos, D., Schuster H., Baker, D., Cichocki A.: Managing Escalation of Collaboration Processes in Crisis Mitigation Situations, Proceedings of the 16th Int. Conference on Data Engineering (ICDE'2000), San Diego (2000)
21. Geppert A., Tombros D.: Event-based Distributed Workflow Execution with EVE, Department of Computer Science, University of Zurich, Technical Report 96.05 (1996, 1998)
22. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., and Barros, A.P.: Workflow Patterns (2002)
23. van der Aalst, W.M.P., Barros, A.P., ter Hofstede, A.H.M. and Kiepuszewski, B.: Advanced Workflow Patterns (2002)
24. Kiepuszewski, B., ter Hofstede, A.H.M., van der Aalst, W.M.P.: Fundamentals of Control Flow in Workflows (2002)
25. 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)
26. Tiani, http://www.tiani.com (2003)
27. D.A.T.A. Corporation, http://www.data.at (2003)
28. Microsoft Corporation: XLANG: Web Services for Business Process Design, http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm (2001)

29. IBM: Web Services Flow Language (WSFL 1.0), http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf (2001)
30. W3C: XML Path Language (XPath) Version 1.0, http://www.w3.org/TR/xpath (1999)
31. BPEL Implementation of the Create Diagnosis workflow, http://www.infosys.tuwien.ac.at/Staff/sd/MedicalBPEL.pdf
32. Collaxa: Collaxa 2.0 BPEL Server, http://www.collaxa.com (2003)
33. Bardram, J.E.: Scenario-Based Design of Cooperative Systems, Re-designing an Hospital Information System in Denmark, Group Decision and Negotiation 9:237–250, Kluwer Academic Publishers (2000)
34. Radiological Society of North America: IHE Technical Framework 1.0, http://www.rsna.org/IHE/index.shtml (2003)
35. Schahram Dustdar (2004). Caramba – A Process-Aware Collaboration System Supporting Ad hoc and Collaborative Processes in Virtual Teams, Distributed and Parallel Databases, 15, 45-66, 2004. Kluwer Academic Publishers
36. Ingrid Wetzel, Ralf Klischewski: Serviceflow beyond workflow? IT support for managing inter-organizational service processes, Proceedings 14th CAiSE. Springer Lecture Notes in Computer Science, P. 500-515, Berlin (2002)
37. Ralf Klischewski, Ingrid Wetzel: Vertragsbasiertes Prozessmanagement als Leitbild für die organisationsübergreifende Workflowunterstützung, Universität Hamburg (2003)